

8360288



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

March 28, 2023

**THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE
RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS
OF:**

**APPLICATION NUMBER: 11/135,210
FILING DATE: May 23, 2005
PATENT NUMBER: 7436980
ISSUE DATE: October 14, 2008**



Certified by

Kathi

Performing the Functions and Duties of the
Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

Meta 1002
p. 1



052305

17589

U.S.

PTO

Express Mail Label No.: EV336791321US

Date of Deposit: 23.May.2005

Mail Stop PATENT APPLICATION
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

PATENT

Case Docket No.: 2004P09036 US01

Date: 23.May.2005

113014 U.S. PTO
11/135210

052305

Sir:

Transmitted herewith for filing is the patent application of:

Inventor(s):

Dorin Comaniciu

Leonid Sigal

Ying Zhu

For: GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

This application includes:

25 pages of specifications, including 5 claims and 1 abstract
1 title page
5 sheets of drawings, formal/ X informal containing 5 Figures
 photographs (design patent application only)

Also enclosed is:

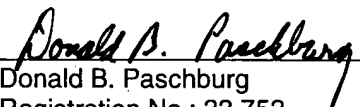
 Declaration and Power of Attorney
 An assignment of the invention to Siemens Corporate Research Inc..
 A certified copy of a application.
X Information Disclosure Statement pursuant to 37 CFR 1.56.

The filing fee has been calculated as shown below:

FOR:	(Col. 1) NO. FILED	(Col. 2) NO. EXTRA	OTHER THAN A SMALL ENTITY RATE FEE
BASIC FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 300.00
SEARCH FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 500.00
EXAMINATION FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 200.00
TOTAL CLAIMS	16 - 20 =	0 x \$ 50.00 =	\$.00
INDEP CLAIMS	2 - 3 =	0 x \$ 200.00 =	\$.00
* MULTIPLE DEPENDENT CLAIM PRESENTED +\$ 360.00 =			\$.00
TOTAL			\$ 1000.00

- X Please charge my Deposit Account No. 19-2179 in the amount of \$ 1000.00 duplicate copy of this sheet is enclosed.
- X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 19-2179. A duplicate copy of this sheet is enclosed.
- X Any additional filing fees required under 37 CFR 1.16.
- X Any patent application processing fees under 37 CFR 1.17.
- X The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 19-2179. A duplicate copy of this sheet is enclosed.
- X Any patent application processing fees under 37 CFR 1.17.
- X Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Dated: 23.May.2005
Siemens Corporation
Customer No. 28524
Tel. +1 (732) 321 3191


Donald B. Paschburg
Registration No.: 33,753

VR

Express Mail Label No. EV336791321US
2004P09036US01
Date of Deposit: 23.May.2005

**APPLICATION FOR LETTERS PATENT
OF THE UNITED STATES**

NAME OF INVENTOR(S):

Leonid Sigal
115 Waterman Street, 4th Floor
Providence, RI 02912
UNITED STATES OF AMERICA
Citizen: USA

Ying Zhu
88 Kingsland Circle
Monmouth Junction, NJ 08852
UNITED STATES OF AMERICA
Citizen: PR of China

Dorin Comaniciu
2 Stuart In. West
Princeton Jct., NJ 08550
UNITED STATES OF AMERICA
Citizen: Romania

TITLE OF INVENTION:

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

TO WHOM IT MAY CONCERN, THE FOLLOWING IS
A SPECIFICATION OF THE AFORESAID INVENTION

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

5 This application claims priority to U.S. Provisional
Application Serial No. 60/574,799, filed on May 27, 2004,
which is herein incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

10 1. Technical Field:

 The present invention relates to image processing, and
more particularly to automatic detection and tracking of
objects in images.

15 2. Discussion of Related Art:

 The problem of describing and recognizing categories
of objects (e.g., faces, people, cars) is important to
computer vision applications. It is common to represent
objects as collections of features with distinctive
20 appearance, spatial extent, and position. There is however
a large variation in how many features are needed and how
these features are detected and represented.

 Therefore, a need exists for a system and method of
detecting and tracking an object, implementing component
25 detection and performing inference over space and time.

SUMMARY OF THE INVENTION

According to an embodiment of the present disclosure,
a computer implemented method for object detection
comprises providing a spatio-temporal model for an object
5 to be detected, providing a video comprising a plurality of
images including the object, and measuring the object as a
collection of components in each image. The method further
comprises determining a probability that the object is in
each image, and detecting the object in any image upon
10 comparing the probabilities for each image to a threshold
for detecting the object.

Providing the spatio-temporal model for the object to
be detected comprises providing detectors for each of the
collection of components.

15 The spatio-temporal model is a graphical model
comprising nodes corresponding to each of the collection of
components and to the object.

Determining the probability that the object is in each
image comprises detecting the object in a current image
20 according to measurements of the object as a collection of
components determined from a prior image and a later image
relative to the current image.

Providing the spatio-temporal model for the object to
be detected further comprises providing a temporal window

defining a plurality of images in which measurements of components detected therein are passed to components detected in the current image.

5 Determining the probability that the object is in each image comprises determining the probability for a position and a size of the object in each image.

The threshold is provided for the object to be detected, wherein the threshold is determined empirically.

10 A joint probability distribution for the spatio-temporal model with N components is:

$$P(\mathbf{X}_0^O, \mathbf{X}_0^{C_0}, \mathbf{X}_0^{C_1}, \dots, \mathbf{X}_0^{C_N}, \dots, \mathbf{X}_T^O, \mathbf{X}_T^{C_0}, \mathbf{X}_T^{C_1}, \dots, \mathbf{X}_T^{C_N}) = \frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{X}_i^O, \mathbf{X}_j^O) \prod_{ik} \psi_{ik}(\mathbf{X}_i^O, \mathbf{X}_i^{C_k}) \prod_{ikl} \psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l}) \prod_i \phi_i(\mathbf{X}_i^O) \prod_{ik} \phi_i(\mathbf{X}_i^{C_k})$$

According to an embodiment of the present disclosure, a program storage device is provided readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for object detection. 15 The method steps comprising providing a spatio-temporal model for an object to be detected, providing a video comprising a plurality of images including the object, and measuring the object as a collection of components in each 20 image. The method further comprises determining a probability that the object is in each image, and detecting

the object in any image upon comparing the probabilities for each image to a threshold for detecting the object.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Preferred embodiments of the present invention will be described below in more detail, with reference to the accompanying drawings:

Figure 1 is a system according to an embodiment of the present disclosure;

10 Figures 2A and 2B are graphic models for a pedestrian and vehicle, respectively, according to an embodiment of the present disclosure;

Figure 3 is an illustration of a pedestrian and identified components of the pedestrian according to an
15 embodiment of the present disclosure;

Figure 4 is a flow chart of a method for message passing in graphic models according to an embodiment of the present disclosure; and

Figure 5 is a flow chart of a method according to an
20 embodiment of the present disclosure.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

According to an embodiment of the present disclosure, a probabilistic framework for automatic component-based
5 detection and tracking of objects in images and/or video combines object detection with tracking in a unified framework. Tracking makes use of object detection for initialization and re-initialization during transient failures for occlusions. Object detection considers the
10 consistency of the detection over time. Modeling objects by an arrangement of image-base, and possibly overlapping, components facilitates detection of complex articulated objects as well as helps in handling partial object occlusions or local illumination changes.

15 Referring to Figures 2A and 2B, object detection and tracking is formulated as an inference in a two-layer graphical model in which a coarse layer node represents the whole object and fine layer nodes represent multiple components of the object. Directed edges between nodes
20 represent learned spatial and temporal probabilistic constraints. Each node in the graphical model corresponds to a position and scale of the component or the object as a whole in an image at a given time instant. Each node also has an associated Adaptive Boosting (Adaboost) detector

that is used to define the local image likelihood and a proposal process. Typically the likelihoods and dependencies are not Gaussian. To infer the 2D position and scale at each node a non-parametric belief propagation (BP) is implemented that uses a variation of particle filtering and can be applied over a loopy graph.

It is to be understood that the present invention may be implemented in various forms of hardware, software, firmware, special purpose processors, or a combination thereof. In one embodiment, the present invention may be implemented in software as an application program tangibly embodied on a program storage device. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture.

Referring to Figure 1, according to an embodiment of the present disclosure, a computer system 101 for implementing an object detection and tracking method can comprise, *inter alia*, a central processing unit (CPU) 102, a memory 103 and an input/output (I/O) interface 104. The computer system 101 is generally coupled through the I/O interface 104 to a display 105 and various input devices 106 such as a mouse and keyboard. The support circuits can include circuits such as cache, power supplies, clock circuits, and a communications bus. The memory 103 can

include random access memory (RAM), read only memory (ROM), disk drive, tape drive, etc., or a combination thereof. The present invention can be implemented as a routine 107 that is stored in memory 103 and executed by the CPU 102 to
5 process the signal from the signal source 108. As such, the computer system 101 is a general purpose computer system that becomes a specific purpose computer system when executing the routine 107 of the present invention.

The computer platform 101 also includes an operating
10 system and micro instruction code. The various processes and functions described herein may either be part of the micro instruction code or part of the application program (or a combination thereof), which is executed via the operating system. In addition, various other peripheral
15 devices may be connected to the computer platform such as an additional data storage device and a printing device.

It is to be further understood that, because some of the constituent system components and method steps depicted in the accompanying figures may be implemented in software,
20 the actual connections between the system components (or the process steps) may differ depending upon the manner in which the present invention is programmed. Given the teachings of the present invention provided herein, one of ordinary skill in the related art will be able to

contemplate these and similar implementations or configurations of the present invention.

Graphical Object Models; An object, such as an automobile (see Figure 2B), is modeled as a spatio-temporal directed graphical model. Each node in the graph represents either the object or a component of the object at time t . Nodes have an associated state vector $X^T = (x, y, s)$ defining the component's real-valued position and scale within an image (where $x, y, s \in \mathbb{R}$). The joint probability distribution for this spatio-temporal graphical object model with N components can be written as:

$$P(\mathbf{X}_0^O, \mathbf{X}_0^{C_0}, \mathbf{X}_0^{C_1}, \dots, \mathbf{X}_0^{C_N}, \dots, \mathbf{X}_T^O, \mathbf{X}_T^{C_0}, \mathbf{X}_T^{C_1}, \dots, \mathbf{X}_T^{C_N}) = \frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{X}_i^O, \mathbf{X}_j^O) \prod_{ik} \psi_{ik}(\mathbf{X}_i^O, \mathbf{X}_i^{C_k}) \prod_{ikl} \psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l}) \prod_i \phi_i(\mathbf{X}_i^O) \prod_{ik} \phi_i(\mathbf{X}_i^{C_k})$$

where X_i^O and $X_i^{C_n}$ is the state of the object, O , and object's n -th component, C_n , at time t respectively ($n \in (1, N)$ and $t \in (1, T)$); $\psi_{ik}(\mathbf{X}_i^O, \mathbf{X}_i^{C_k})$ is the temporal compatibility of object state between frames i and j ; $\psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l})$ is the spatial compatibility of the object and it's components at frame i ; $\psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l})$ is the spatial compatibility between object components at frame i and $\phi_i(\mathbf{X}_i^O)$ and $\phi_i(\mathbf{X}_i^{C_k})$ denote the

local evidence for the object and component states respectively.

According to an embodiment of the present disclosure, a system comprises the following modules: a graphical model
5 modules, an inference module providing the ability to infer a state of each node in the graph, a local evidence distribution (or image likelihood) module, a proposal process module for some or all nodes in a graphical model, and a module comprising a set of spatial and/or temporal
10 constraints corresponding to the edges in a graph.

Building the Graphical Model; For a single frame, objects are represented using a two-layer spatial graphical model, for example as shown in Figures 2A and 2B. The fine, component, layer 201 includes a set of loosely connected
15 parts, e.g., 202. The course, object, layer 203 corresponds to an entire appearance model of the object and is connected to all constituent components, e.g., 202.

Examples of such models for pedestrian and vehicle detection are shown in the shaded regions of Figure 2A and
20 2B, respectively. In both cases objects are modeled using four overlapping image components. For the vehicle the components are: top-left (TL) 204, top-right (TR) 205, bottom-right (BR) 206 and bottom-left (BL) 207 corners; while for an image 301 of the pedestrian, they are: head

(HD) 208, left arm (LA) 209, right arm (RA) 210 and legs
(LG) 202 (see Figure 3).

To integrate temporal constraints the spatial graphical
models are extended over time to an arbitrary length
5 temporal window. The resulting spatio-temporal graphical
models are shown in Figures 2A and 2B. Having a two-layer
graphical model allows the inference process to reason
explicitly about the object as a whole, e.g., 203, and
reduce the complexity of the graphical model by allowing
10 the assumption of the conditional independence of
components, e.g., 202 and 208-210, over time given the
overall object appearance. Alternatively, a single object
layer model can be built with bi-directional temporal
constraints.

15 As can be seen in Figures 2A and 2B, an object at a
current time is denoted as O_t , with an earlier model denoted
as $O_{t-(w-1)/2}$ and a later model denoted as $O_{t+(w-1)/2}$.

Learning Spatial and Temporal Constraints; Each directed
edge between components i and j has an associated potential
20 function $\psi_{ij}(X_i, X_j)$ that encodes the compatibility between
pairs of node states. The potential $\psi_{ij}(X_i, X_j)$ is modeled using
a mixture of M_{ij} Gaussians:

$$\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) = \lambda^0 \mathcal{N}(\mathbf{X}_j; \mu_{ij}, \Lambda_{ij}) + (1 - \lambda^0) \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_j; F_{ijm}(\mathbf{X}_i, \mathbf{X}_j), G_{ijm}(\mathbf{X}_i, \mathbf{X}_j))$$

where λ^0 is a fixed outlier probability, μ_{ij} and Λ_{ij} are the mean and covariance of the Gaussian outlier process, and $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ are functions that return the mean and covariance matrix respectively of the m -th Gaussian mixture component. δ_{ijm} is the relative weight of an individual component and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$. For experiments described herein $M_{ij} = 2$ mixture components were used.

Given a set of labeled images, where each component is associated with a single reference point, an iterative Expectation-Maximization (EM) method with K-means initialization is implemented to learn $F_{ijm}(\cdot)$ of the form:

$$F_{ijm}(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{X}_i + \left[\frac{\mu_{ijm}^x}{\mu_{ijm}^s}, \frac{\mu_{ijm}^y}{\mu_{ijm}^s}, \mu_{ijm}^s \right]^T \quad (1)$$

where $\mu_{ijm}^x, \mu_{ijm}^y, \mu_{ijm}^s$ is the mean position and scale of component or object j relative to i . $G_{ijm}(\cdot)$ is assumed to be diagonal matrix, representing the variance in relative position and scale.

AdaBoost Image Likelihoods; A likelihood $\phi_i(X_i)$ models the probability of observing the image conditioned on the state X_i of the node i . A likelihood model according to an embodiment of the present disclosure uses a boosted classifier and is robust to partial occlusions and the

variability of image statistics across many different inputs.

Boosted detectors are trained for each component. For simplicity AdaBoost may be used without a cascade. Training
5 with a cascade would likely improve the computational efficiency of the system. To reduce the number of false positives produced by the detectors, a bootstrapping method iteratively adds false positives that are collected by running the trained strong classifier over the set of
10 background images (e.g., those not containing the desired object) and the detectors are retrained using the old positive and the new extended negative sets.

Given a set of labeled patterns the AdaBoost procedure learns a weighted combination of base weak classifiers,
15 $h(I) = \sum_{k=1}^K \alpha_k h_k(I)$, where I is an image pattern, and $h_k(I)$ is the weak classifier chosen for the round k of boosting, and α_k is the corresponding weight. A weak classifier method is implemented: $h_k(I) = p_k([(f_k(I))^{\beta_k}]^{1/\beta_k} < \theta_k)$ where $f_k(I)$ is a feature of the pattern I computed by convolving I with the
20 delta function over the extent of a spatial template; θ_k is a threshold, p_k is the polarity indicating the direction of inequality, and $\beta_k \in \{1, 2\}$ allowing for a symmetric two sided pulse classification.

The output of the AdaBoost classifier is a confidence $h_k(I)$ that the given pattern I is of the desired class. An object is considered present if $h(I) \geq \frac{1}{2} \sum_{k=1}^K \alpha_k$. This confidence is converted into a likelihood function by normalizing the α_k 's, so that $h(I) \in [0,1]$, and exponentiating:

$$\phi_i(I|\mathbf{X}_i) \propto \exp(h(I)/T) \quad (2)$$

where T is a temperature parameter that controls the smoothness of the likelihood function, with smaller values of T leading to a peakier distribution. Consequently the likelihood can be annealed by deriving a schedule with which T changes. An exponential annealing schedule $T = T_0 v^\kappa$ has been determined where T_0 is the initial temperature, v is a fraction $\in (0,1)$, and κ is the annealing iteration, to work well in practice.

Depending on an object one may or may not have a likelihood or a proposal process for the object layer nodes. For example, if the whole appearance of an object is too complicated to model as a whole (e.g., arbitrary size vehicles) and can only be modeled in terms of components, a uniform likelihood can be assumed over the object state space. In such cases the object layer nodes simply fuse the component information to produce estimates for the object state that are consistent over time.

Non-parametric BP; Inferring the state of the object and its components in a framework is defined as estimating belief in a graphical model. A form of non-parametric belief propagation PAMPAS (proposed by M. Isard in "PAMPAS: Real-valued graphical models for computer vision", in the Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. I, pp. 421-428, 2004.) is implemented to deal with this task. The approach is a generalization of particle filtering, which allows inference over arbitrary graphs rather than a simple chain. In this generalization the message used in belief propagation is approximated with a kernel density, formed by propagating a particle set through a mixture of Gaussians density, and the conditional distribution used in standard particle filtering is replaced by product of incoming messages. Most of the computational complexity lies in sampling from a product of kernel densities required for message passing and belief estimation; we use efficient sequential multiscale Gibbs sampling and epsilon-exact sampling to address this problem.

Individual messages may not constrain a node well, however the product over all incoming messages into the node tends to produce a very tight distribution in the state space. For example, any given component of a vehicle

is incapable of estimating the height of the vehicle reliably, however once information from all components is intergraded in the object layer node, an estimate for the overall object size can be determined.

5 More formally a message m_{ij} is written as

$$m_{ij}(X_j) = \int \psi_{ij}(X_i, X_j) \phi_i(X_i) \prod_{k \in \{A_i \setminus j\}} m_{ki}(X_i) dX_i \quad (3)$$

where A_i is the set of neighbors of node i , $\{A_i \setminus j\}$ is the set of neighbors of node i except node j , and $\phi_i(X_i)$ is the local evidence (or likelihood) associated with the node i , and

10 $\psi_{ij}(X_i, X_j)$ is the potential designating the compatibility between the states of node i and j . The details of how the message updates can be carried out by stratified sampling from belief and proposal function is illustrated in Figure 4.

15 Referring to Figure 4, a method for passing messages between nodes of the graphical model includes determining a likelihood associated with node i 401, observing an image conditioned on the state of node i 402, determining a potential designating the compatibility between states of nodes i and j , nodes i and k and nodes i and l 403. Messages
20 405 are passed between nodes to propagate information about what states neighboring nodes would be 404.

While it is possible to perform inference over the spatio-temporal model defined for the entire image sequence, there are many applications for which this is not an option due to the lengthy off-line processing needed. A
5 w -frame windowed smoothing algorithm is used where w is an odd integer ≥ 1 . There are two ways one can do windowed smoothing: in an object-detection centric way or a tracking centric way. In the former all nodes are re-initialized every time a window is shifted, hence the temporal
10 integration is only applied in the window of size w . In the tracking centric way only the nodes associated with a new frame are initialized, which tends to enforce temporal consistency from before $t-(w-1)/2$. While the tracking-centric way tends to converge faster and produce more
15 consistent results over time, it is also less sensitive to objects entering and leaving the scene. Note that with $w=1$, the algorithm resembles single frame component-based fusion.

Proposal Process; To reliably detect and track the
20 object non-parametric BP makes use of the bottom-up proposal process, that constantly looks for and suggests alternative hypothesis for the state of the object and components. Proposal distribution is modeled using a weighted particle set. To form a proposal particle set for

a component, a corresponding AdaBoost detector is run over an image at a number of scales to produce a set of detection results that score above the $\frac{1}{2}\sum_{k=1}^K\alpha_k$ threshold.

While this set tends to be manageable for the entire appearance model, it is large for non-specific component detectors (e.g., a few thousand locations can easily be found). To reduce the dimensionality only the top P scoring detections are kept, where P is on the order of a 100 to 200. To achieve breadth of search sample particles from the proposal were imported using uniform distribution.

Referring to Figure 5, a computer implemented method for object detection includes providing a spatio-temporal model, e.g., see Figures 2A and 2B, for an object 501, providing a video including a plurality of images including the object 502, and measuring the object as a collection of components in each image of the video 503. The method further includes determining a probability that the object is in each image 504 by using message passing between components represented as nodes of the spatio-temporal model, and detecting the object in any image upon comparing the probabilities for each image to a threshold for detecting the object 505.

Experiments; Tests were performed using a set of images collected with a single car-mounted grayscale camera. The

result of vehicle detection and tracking over a sequence of 55 consecutive frames was evaluated. A 3-frame spatio-temporal object model was used and was shifted in a tracking centric way over time. BP was run with 30 particles for 10 iterations at every frame. For comparison a simple fusion scheme was implemented that averages the best detection result from each of the four components to produce an estimate for the vehicle position and scale independently at every frame. The performance of the simple fusion detection is poor suggesting that the noisy component detectors often do not have the global maximum at the correct position and scale. In contrast, the spatio-temporal object model consistently combines the evidence for accurate estimates throughout the sequence.

A 3-frame spatio-temporal object model was run at a single instance in time for two pedestrians in two different scenes. Similar to the vehicle detection BP was run with 30 particles for 10 iterations. For both experiments the temperature of the likelihood is set to $T_0=0.2$.

According to an embodiment of the present disclosure, an object detection and tracking framework exploits boosted classifiers and non-parametric belief propagation. The approach provides component-based detection and integrates

temporal information over an arbitrary size temporal window. The performance of the framework is illustrated with two classes of objects: vehicles and pedestrians. In both cases position and scale of the objects and their
5 components can be reliably inferred.

Having described embodiments for a system and method for a object detection and tracking, it is noted that modifications and variations can be made by persons skilled in the art in light of the above teachings. It is therefore
10 to be understood that changes may be made in the particular embodiments of the invention disclosed which are within the scope and spirit of the invention as defined by the appended claims. Having thus described the invention with the details and particularity required by the patent laws,
15 what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A computer implemented method for object detection comprising:

5 providing a spatio-temporal model for an object to be detected;

providing a video comprising a plurality of images including the object;

10 measuring the object as a collection of components in each image;

determining a probability that the object is in each image; and

15 detecting the object in any image upon comparing the probabilities for each image to a threshold for detecting the object.

2. The computer implemented method of claim 1, wherein providing the spatio-temporal model for the object to be detected comprises providing detectors for each of the
20 collection of components.

3. The computer implemented method of claim 1, wherein the spatio-temporal model is a graphical model comprising nodes corresponding to each of the collection of components and to the object.

5

4. The computer implemented method of claim 1, wherein determining the probability that the object is in each image comprises detecting the object in a current image according to measurements of the object as a collection of components determined from a prior image and a later image relative to the current image.

10

5. The computer implemented method of claim 1, wherein providing the spatio-temporal model for the object to be detected further comprises providing a temporal window defining a plurality of images in which measurements of components detected therein are passed to components detected in the current image.

15

6. The computer implemented method of claim 1, wherein determining the probability that the object is in each image comprises determining the probability for a position and a size of the object in each image.

20

7. The computer implemented method of claim 1, wherein the threshold is provided for the object to be detected, wherein the threshold is determined empirically.

5 8. The computer implemented method of claim 1, wherein a joint probability distribution for the spatio-temporal model with N components is:

$$P(\mathbf{X}_0^O, \mathbf{X}_0^{C_0}, \mathbf{X}_0^{C_1}, \dots, \mathbf{X}_0^{C_N}, \dots, \mathbf{X}_T^O, \mathbf{X}_T^{C_0}, \mathbf{X}_T^{C_1}, \dots, \mathbf{X}_T^{C_N}) = \frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{X}_i^O, \mathbf{X}_j^O) \prod_{ik} \psi_{ik}(\mathbf{X}_i^O, \mathbf{X}_i^{C_k}) \prod_{ikl} \psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l}) \prod_i \phi_i(\mathbf{X}_i^O) \prod_{ik} \phi_i(\mathbf{X}_i^{C_k})$$

10

9. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for object detection, the method steps comprising:

15 providing a spatio-temporal model for an object to be detected;

providing a video comprising a plurality of images including the object;

measuring the object as a collection of components in each image;

20 determining a probability that the object is in each image; and

detecting the object in any image upon comparing the probabilities for each image to a threshold for detecting the object.

5 10. The method of claim 9, wherein providing the spatio-temporal model for the object to be detected comprises providing detectors for each of the collection of components.

10 11. The method of claim 9, wherein the spatio-temporal model is a graphical model comprising nodes corresponding to each of the collection of components and to the object.

15 12. The method of claim 9, wherein determining the probability that the object is in each image comprises detecting the object in a current image according to measurements of the object as a collection of components determined from a prior image and a later image relative to the current image.

20 13. The method of claim 9, wherein providing the spatio-temporal model for the object to be detected further comprises providing a temporal window defining a plurality of images in which measurements of components detected

therein are passed to components detected in the current image.

14. The method of claim 9, wherein determining the
5 probability that the object is in each image comprises
determining the probability for a position and a size of
the object in each image.

15. The method of claim 9, wherein the threshold is
10 provided for the object to be detected, wherein the
threshold is determined empirically.

16. The method of claim 9, wherein a joint probability
distribution for the spatio-temporal model with N
15 components is:

$$P(\mathbf{X}_0^O, \mathbf{X}_0^{C_0}, \mathbf{X}_0^{C_1}, \dots, \mathbf{X}_0^{C_N}, \dots, \mathbf{X}_T^O, \mathbf{X}_T^{C_0}, \mathbf{X}_T^{C_1}, \dots, \mathbf{X}_T^{C_N}) =$$

$$\frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{X}_i^O, \mathbf{X}_j^O) \prod_{ik} \psi_{ik}(\mathbf{X}_i^O, \mathbf{X}_i^{C_k}) \prod_{ikl} \psi_{kl}(\mathbf{X}_i^{C_k}, \mathbf{X}_i^{C_l}) \prod_i \phi_i(\mathbf{X}_i^O) \prod_{ik} \phi_i(\mathbf{X}_i^{C_k})$$

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

Abstract of the Disclosure

A computer implemented method for object detection
5 includes providing a spatio-temporal model for an object to
be detected, providing a video including a plurality of
images including the object, and measuring the object as a
collection of components in each image. The method further
includes determining a probability that the object is in
10 each image, and detecting the object in any image upon
comparing the probabilities for each image to a threshold
for detecting the object.

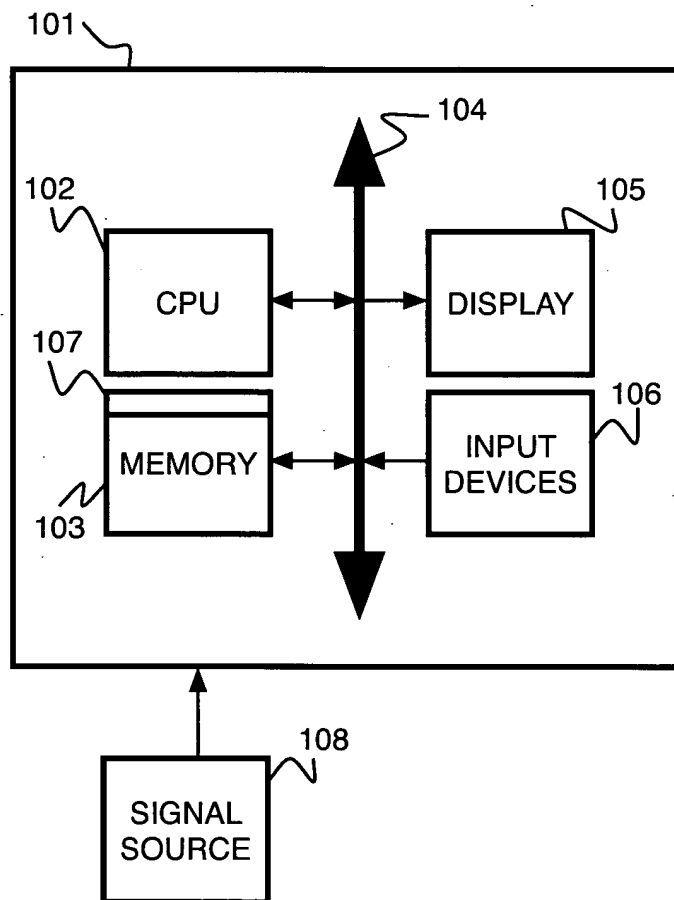


FIGURE 1

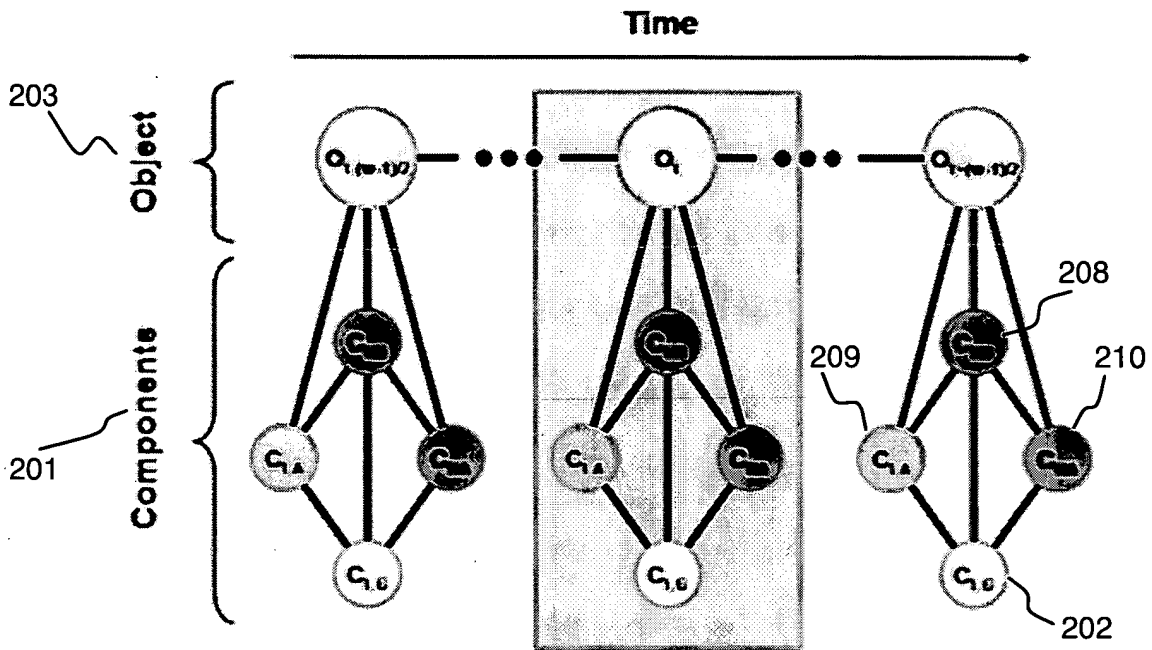


FIGURE 2A

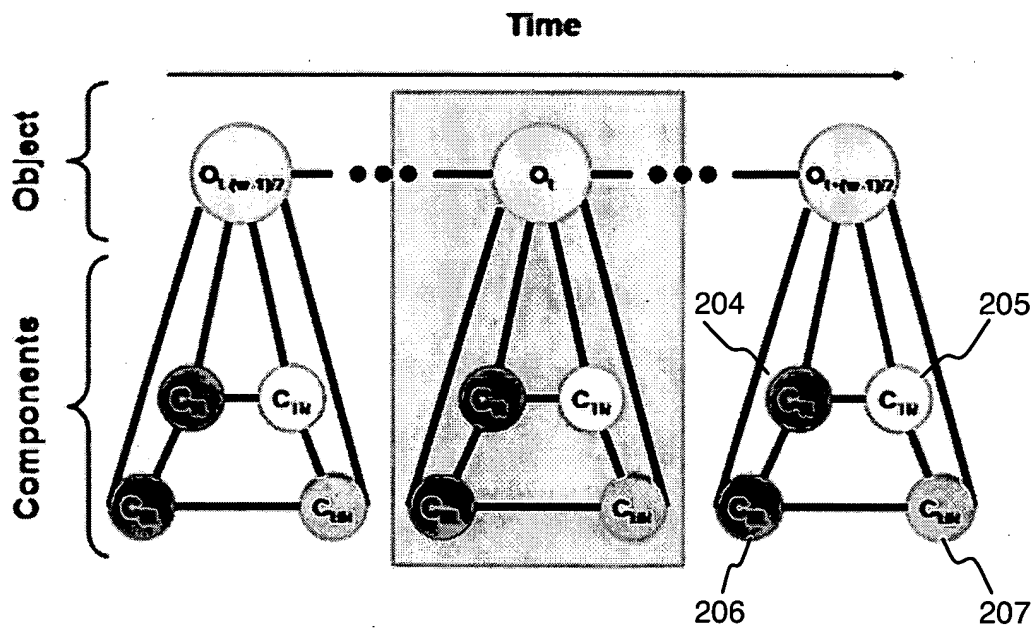
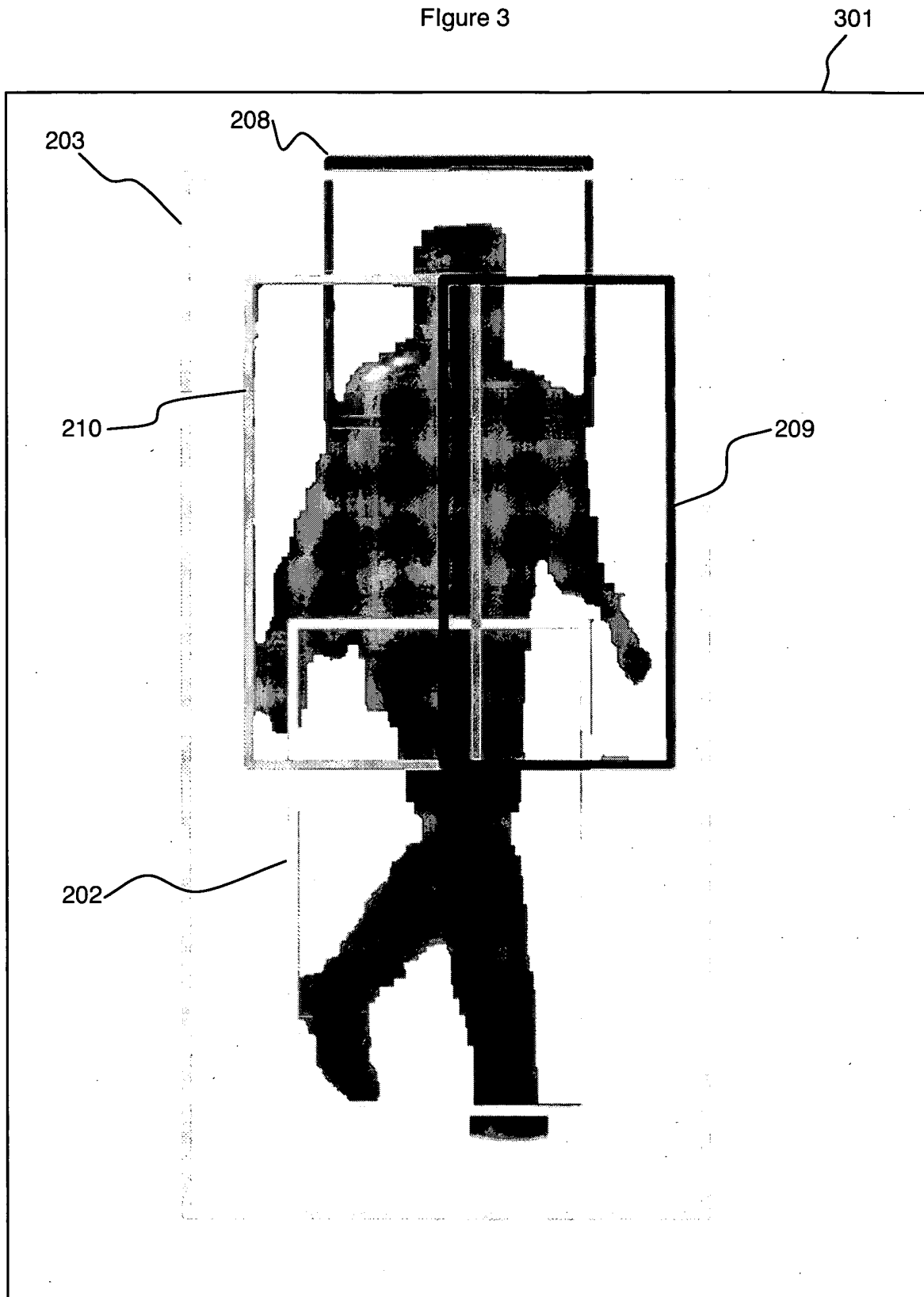


FIGURE 2B

3/5

Figure 3



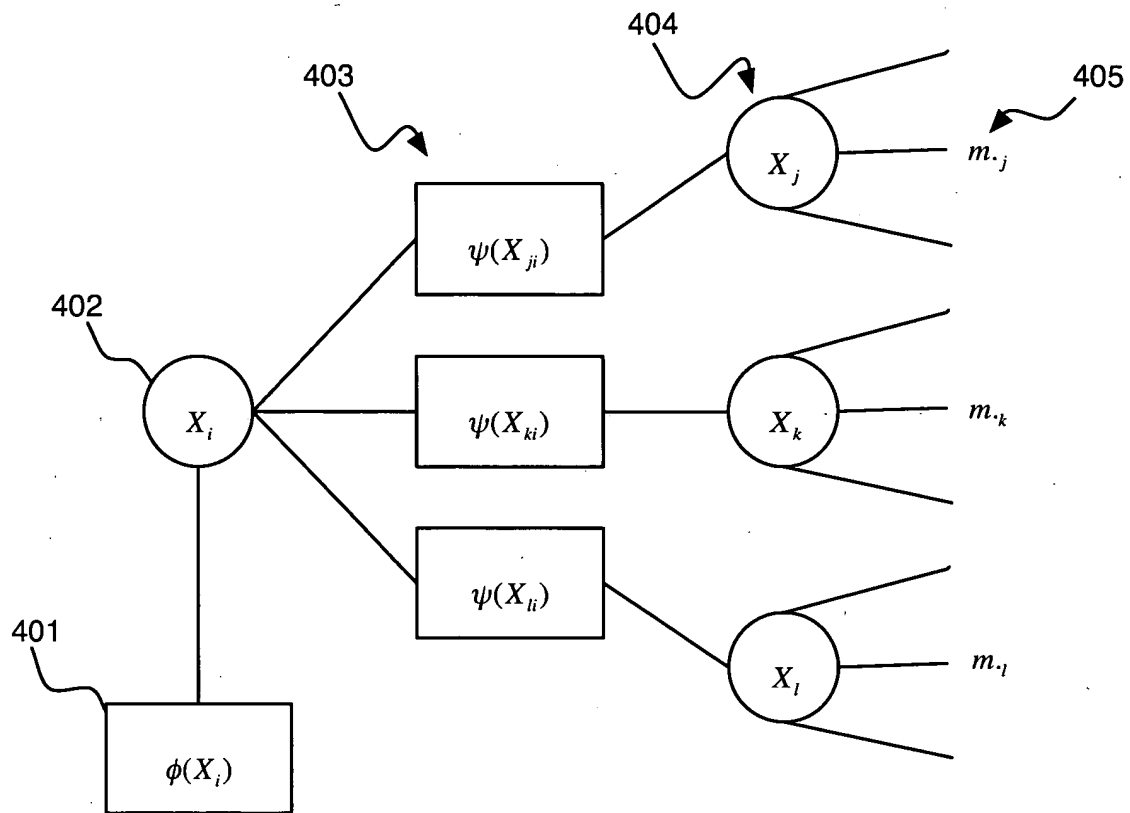


FIGURE 4

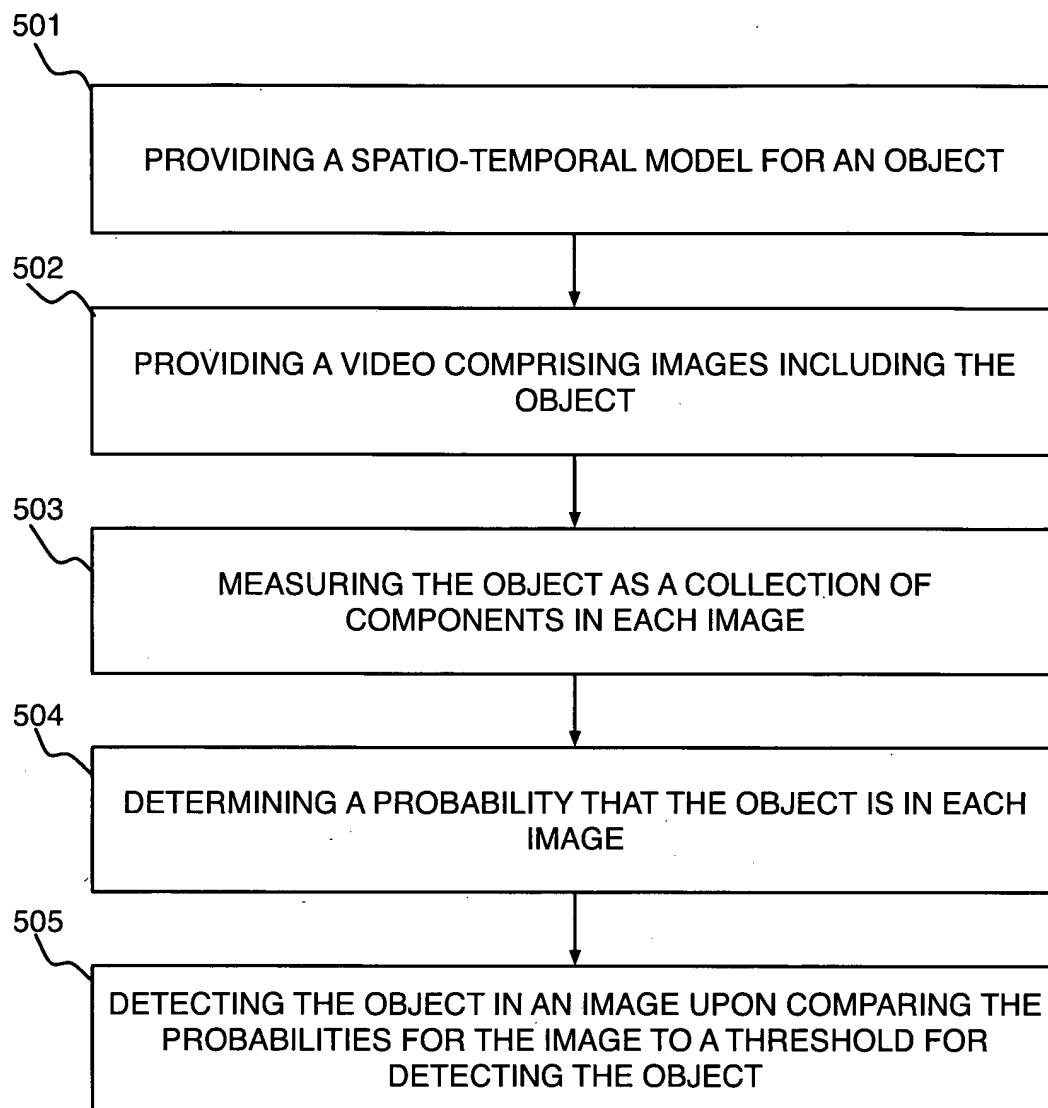


FIGURE 5

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 8, 2004

11135210

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	16	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	16 minus 20 = *	
INDEPENDENT CLAIMS	2 minus 3 = *	
MULTIPLE DEPENDENT CLAIM PRESENT ☐		

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

- * If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
 - ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
 - *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
- The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

SMALL ENTITY TYPE ☐

OR OTHER THAN SMALL ENTITY

RATE	FEE
BASIC FEE	150.00
X\$ 25=	
X100=	
+180=	
TOTAL	

RATE	FEE
BASIC FEE	300.00
X\$50=	
X200=	
+360=	
TOTAL	300

SMALL ENTITY

OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

05/25/2005 HTECKLU1 00000030 192179 11135210

01 FC:1011	300.00 DA
02 FC:1111	500.00 DA
03 FC:1311	200.00 DA

PTO-1556
(5/87)

Express Mail Label No.: EV336791321US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Sigal et al.

Docket: 2004P09036US01 (8706-817)

Serial No.: unassigned

Date: May 23, 2005

Filed: concurrently herewith

For: **BOOSTED GRAPHICAL MODELS FOR OBJECT DETECTION AND TRACKING**

Commissioner for Patents
PO Box 1450
Alexandria VA 20231-1450

INFORMATION DISCLOSURE STATEMENT

Sir:

Pursuant to Applicant(s) duty of disclosure, the information listed in the attached form PTO-1449 is brought to the attention of the Examiner. Copies of the listed items are enclosed.

The citation of the listed items is not a representation that they constitute a complete or exhaustive listing of the relevant art or that the references are prior art. The items listed are submitted in good faith, but are not intended to substitute for the Examiner's search. It is hoped, however, that in addition to apprising the Examiner of these particular items, they will assist in identifying fields of search and in making as full and complete a search as possible.

The filing of this information disclosure statement is not an admission that the information cited herein is, or is considered to be, material to patentability as defined in 37 C.F.R. § 1.56(b).

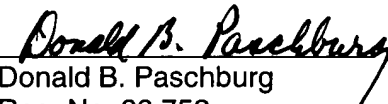
- ☒ This information disclosure statement is being filed within three (3) months of the filing date of this application.
- ☐ This information disclosure statement is being filed within three (3) months of the date of entry of the national stage as set forth in 37 C.F.R. § 1.491 in an international application.
- ☐ To the best of Applicant(s) knowledge, this information disclosure statement is being filed before the date of mailing of a first Office Action on the merits in connection with this case.
- ☐ Statement under 37 C.F.R. § 1.97(e):
I hereby state that each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than 3 months prior to the filing of this statement.
- ☐ Enclosed herewith is a petition under 37 C.F.R. § 1.97(d)(ii).
 - ☐ Enclosed by check is the petition fee of \$130.00. (37 C.F.R. § 1.17(i)(1))
 - ☐ Please charge the fee of \$130.00 for petition fee to deposit account 19-2179.
- ☐ Enclosed by check is the \$180.00 fee required by 37 C.F.R. § 1.17(p).
- ☐ Please charge the fee of \$180.00 fee required by 37 C.F.R. § 1.17(p) by enclosed herewith.

[X] Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17 at any time during the pendency of this application, or credit any overpayment of such fee(s) to Deposit Account 19-2179. Also, in the event any extensions of time for responding are required for the pending application(s), please treat this paper as a petition to extend the time as required and charge Deposit Account No. 19-2179 therefor. A COPY OF THIS SHEET IS ENCLOSED.

The claims of the application as now presented are believed to patentably distinguish over the prior art and to be in condition for allowance. Early and favorable consideration of the case is respectfully requested.

Respectfully submitted,

By:


Donald B. Paschburg
Reg. No. 33,753
Attorney for Applicant(s)

Mailing Address:

Siemens Corporation
Intellectual Property Dept, 5TH Floor
170 Wood Avenue South
Iselin, New Jersey 08830
(732) 321-3191
(732) 321-3030 (FAX)

Form PTO-1449		U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE		ATTY. DOCKET NO. 2004P09036US01 (8706-817)		SERIAL NO. unassigned	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary)				APPLICANT Sigal et al.			
				FILING DATE concurrently here with		GROUP ART UNIT	
U.S. PATENT DOCUMENTS							
EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
FOREIGN PATENT DOCUMENTS							
		PUBLICATION NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	ABSTRACT TRANSLATION
							YES NO
OTHER PRIOR ART (Including Author, Title, Date, Pertinent Pages, Etc.)							
			M. Isard. PAMPAS: Real-valued graphical models for computer vision, CVPR, Vol. 1, pp. 613-620, 2003.				
			K. Okuma, A. Teleghani, N. de Freitas, J. Little and D. Lowe. A boosted particle filter: Multi-target detection and tracking, ECCV, 2004.				
			L. Sigal, S. Bhatia, S. Roth, M. Black and M. Isard. Tracking loose- limbed people, CVPR, 2004.				
			E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, MIT Laboratory for Information and Decision Systems Technical Report P-2551 (October 15, 2002).				
			P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, CVPR, 2001.				
			P. Viola, M. Jones and D. Snow. Detecting pedestrians using patterns of motion and appearance, ICCV, pp. 734-741, 2003.				
EXAMINER				DATE CONSIDERED			
* EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.							

(Form PTO-1449 [6-4])

Tracking Loose-limbed People

Leonid Sigal* Sidharth Bhatia* Stefan Roth* Michael J. Black * Michael Isard†

*Department of Computer Science, Brown University Providence, RI 02912

†Microsoft Research Silicon Valley, Mountain View, CA 94043

{ls,sibhatia,roth,black}@cs.brown.edu, misard@microsoft.com

Abstract

We pose the problem of 3D human tracking as one of inference in a graphical model. Unlike traditional kinematic tree representations, our model of the body is a collection of loosely-connected limbs. Conditional probabilities relating the 3D pose of connected limbs are learned from motion-captured training data. Similarly, we learn probabilistic models for the temporal evolution of each limb (forward and backward in time). Human pose and motion estimation is then solved with non-parametric belief propagation using a variation of particle filtering that can be applied over a general loopy graph. The loose-limbed model and decentralized graph structure facilitate the use of low-level visual cues. We adopt simple limb and head detectors to provide “bottom-up” information that is incorporated into the inference process at every time-step; these detectors permit automatic initialization and aid recovery from transient tracking failures. We illustrate the method by automatically tracking a walking person in video imagery using four calibrated cameras. Our experimental apparatus includes a marker-based motion capture system aligned with the coordinate frame of the calibrated cameras with which we quantitatively evaluate the accuracy of our 3D person tracker.

1 Introduction

We present a fully automatic method for tracking human bodies in 3D. Initialization and failure recovery are facilitated by the use of a loose-limbed body model [22] in which limbs are connected via learned probabilistic constraints. The tracking problem is formulated as one of inference in a graphical model and belief propagation is used to estimate the pose of the body at each time-step. Each node in the graphical model represents the 3D position and orientation of a limb (Figure 1). Directed edges between nodes represent statistical dependencies and these constraints between limbs are used to form messages that are sent to neighboring nodes in space and time. Additionally, each node has an associated likelihood defined over a rich set of image cues using a learned Gibbs model [19, 28]. The combination of

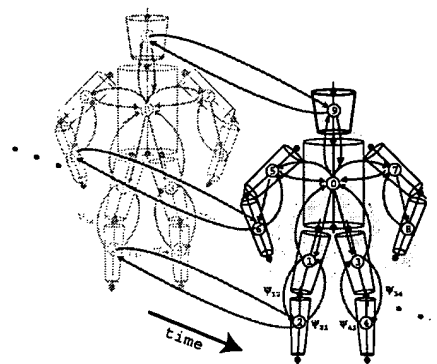


Figure 1: Graphical model for a person. Nodes represent limbs and arrows represent conditional dependencies between limbs. For clarity only a few temporal dependencies have been shown: in our model each part i at time t is connected by an edge to part i at times $t - 1$ and $t + 1$.

highly non-Gaussian likelihoods and a six-dimensional continuous parameter space (3D position and orientation) for each limb makes standard belief propagation algorithms infeasible. Consequently we exploit a form of non-parametric belief propagation that uses a variation of particle filtering and can be applied over a loopy graph [13, 25].

There are a number of significant advantages to this approach as compared to traditional methods for tracking human motion. Most current techniques model the body as a kinematic tree in 2D [14], 2.5D [17], or 3D [2, 5, 21, 23] leading to a high-dimensional parameter space (25–50 dimensions is not uncommon). Searching such a space directly is impractical and so current methods typically rely on manual initialization of the body model. Additionally, they often exploit strong priors characterizing the types of motions present. When such algorithms lose track (as they always do), the dimensionality of the state space makes it difficult to recover.

While the full body pose is hard to recover directly, the location and pose of individual limbs is much easier to compute. Many good head detectors exist and limb detectors have been used for some time (e.g. [18]). The approach we take here can use bottom up information from feature

detectors of any kind and consequently should be able to cope with a rich variety of input images. In our implementation we exploit background/foreground separation for computational simplicity but part detectors which perform well against arbitrary backgrounds are becoming standard [18, 26].

With a kinematic tree model, exploiting this partial, “bottom-up” information is challenging. If one could definitively detect the body parts then inverse kinematics could be used to solve for the body pose, but in practice low-level part detectors are noisy and unreliable. The use of a loose-limbed model and belief propagation provides an elegant framework for incorporating information from part detectors. Because the inference algorithm operates over a general graph rather than a forward chain as in traditional particle filter trackers, it is also straightforward to perform forward-backward smoothing of the limb trajectories without modifying the basic approach.

A loose-limbed body model requires a specification of the probabilistic relationships between joints at a given time instant and over time. We represent these non-Gaussian relationships using mixture models that are learned from a database of motion capture sequences. It is worth noting that these models encode information about joint limits and represent a relatively weak prior over human poses, which is appropriate for tracking varied human motions.

The model also requires an image likelihood measure for each limb. Using training data of known limb poses in images, we learn a novel likelihood model that captures the joint statistics of first and second derivative filter responses at multiple scales. We formulate and learn the likelihoods using a Gibbs model based on the maximum entropy principle [28].

We test the method by tracking a subject viewed from four calibrated cameras in an indoor environment with no special clothing. Quantitative evaluation is performed using a novel motion capture testbed that provides “ground truth” human motion from a commercial motion capture system that is synchronized with the video streams¹. In particular we compare the accuracy of our method with that of a more standard kinematic tree body tracker using annealed particle filtering [5]. We find that the traditional approach loses track rapidly compared with the loose-limbed model when the image quality is poor.

Previous work

Disaggregated models are not new for finding or tracking articulated objects and date back at least to Fischler and Elschlager’s pictorial structures [8]. Variations on this type of model have been recently applied by Burl *et al.* [1], Felzenszwalb and Huttenlocher [7], Coughlan and Ferreira

[3], Ioffe and Forsyth [10, 11, 12] and Ramanan and Forsyth [18]. Ioffe and Forsyth [10, 11] first find body parts and then group them into figures in a bottom-up fashion. The approach exploits the fact that they have a discrete set of features that need to be assembled, but it prevents them from using rich likelihood information to “co-operate” with the body model when estimating the pose. Ramanan and Forsyth [18] propose an elegant estimation of appearance models jointly with the body’s trajectory, but their inference algorithm relies on the fact that the 2D model has a relatively low-dimensional state-space for each body part. A similar approach to ours has been adopted in [27] for tracking a 2D human silhouette using a dynamic Markov network. A much simplified observation model was adopted in [27] and their system does not perform automatic initialization. They adopt a somewhat different inference algorithm and a comparison between the two methods merits future research.

In previous work [22] we presented the general loose-limbed body model and belief propagation algorithm but only addressed the problem of human pose estimation at a single time instant; here we extend the inference method over time to perform visual tracking. In [22], the potential functions linking limbs were constructed manually whereas here they are learned from training data and are also extended in time. Here we also propose a multi-view eigenmethod to implement bottom-up body part detectors and we exploit a learned Gibbs likelihood model [19].

2 Loose-limbed body model

Following the framework in [22] the body is represented by a graphical model in which each graph node corresponds to a body part (upper leg, torso, etc.). Each part has an associated configuration vector defining the part’s position and orientation in 3-space. Placing each part in a global coordinate frame enables the part detectors to operate independently while the full body is assembled by inference over the graphical model. Edges in the graphical model correspond to position and angle relationships between adjacent body parts in space and time, as illustrated in Figure 1.

In order to describe the body parts in a graphical model, we assume the variables in a node are conditionally independent of those in non-neighboring nodes given the values of the node’s neighbors. Each part/limb is modeled by a tapered cylinder having 5 fixed and 6 estimated parameters. The fixed parameters $\Phi_i = (l_i, w_i^p, w_i^d, o_i^p, o_i^d)$ correspond respectively to the part length, width at the proximal and distal ends and the offset of the proximal and distal joints along the axis of the limb as shown in Figure 2. The estimated parameters $\mathbf{X}_i^T = (\mathbf{x}_i^T, \Theta_i^T)$ represent the configuration of the part i in a global coordinate frame where $\mathbf{x}_i \in \mathbb{R}^3$ and $\Theta_i \in \text{SO}(3)$ are the 3D position of the proximal joint

¹Available at <http://www.cs.brown.edu/research/vision/motioncapture/>.

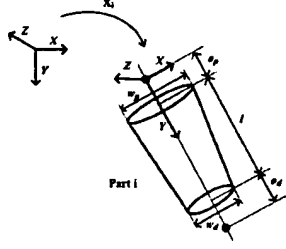


Figure 2: Parameterization of part i .

and the angular orientation of the part respectively. The rotations are represented by unit quaternions.

Each directed edge between parts i and j has an associated potential function $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ that encodes the compatibility between pairs of part configurations and intuitively can be thought of as the probability of configuration $\mathbf{X}_j$ of part j conditioned on the $\mathbf{X}_i$ of part i . The potential $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ is in general non-Gaussian and is approximated by a mixture of M_{ij} Gaussians:

$$\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) = \lambda^0 \mathcal{N}(\mathbf{X}_j; \mu_{ij}, \Lambda_{ij}) + (1 - \lambda^0) \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_j; F_{ijm}(\mathbf{X}_i), G_{ijm}(\mathbf{X}_i)) \quad (1)$$

where λ^0 is a fixed outlier probability, μ_{ij} and Λ_{ij} are the mean and covariance of the Gaussian outlier process, and $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ are functions that return the mean and covariance matrix respectively of the m -th Gaussian mixture component. $\delta_{ijm} \geq 0$ is the relative weight of an individual component and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$.

2.1 Learning limb conditionals

Given a ground truth parameter vector $\mathbf{X}_j$ for part j , we can construct the 3D object-to-world transform $M(\mathbf{X}_j)$ defining the pose of the limb j as

$$M(\mathbf{X}_j) = M(\mathbf{X}_i)M(\mathbf{X}_{ij}) \quad (2)$$

where $M(\mathbf{X}_i)$ is the pose of the neighboring spatial or temporal part, and $\mathbf{X}_{ij}$ encodes the position and orientation of part j in i 's coordinate frame. We can approximate the potential compatibility functions:

$$\begin{aligned} \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) &\approx p(\mathbf{X}_j | \mathbf{X}_i) \\ &= p(M(\mathbf{X}_j) | M(\mathbf{X}_i)) \\ &= p(M(\mathbf{X}_i)M(\mathbf{X}_{ij}) | M(\mathbf{X}_i)) \\ &\approx p(M(\mathbf{X}_{ij})) = p(\mathbf{X}_{ij}) \end{aligned} \quad (3)$$

by learning the distribution over $\mathbf{X}_{ij}$ to fix the $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ functions from (1). We model all spatial and temporal potentials using mixtures of Gaussians with

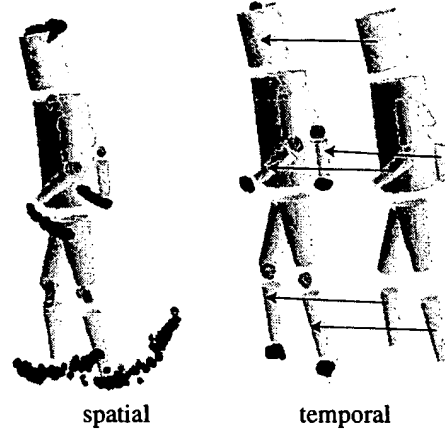


Figure 3: *Learned Spatial and Temporal Potentials.* Spatial and temporal potentials are illustrated by sampling from them. The potentials for the lower limbs (arms and legs) and head are shown. The spatial potentials show the distribution of limb positions and orientations conditioned on the neighboring limb. Green spheres indicate the joint position of a sample while the red spheres indicate the distal end of the limb for each sample. The spread of these samples illustrates the learned distribution encoded by the potentials. The temporal potentials shown are for the forward time direction.

$M_{ij} = 3$ components, and for example the first component $F_{ij1}(\mathbf{X}_j) = M^{-1}(M(\mathbf{X}_i)M(\hat{\mathbf{X}}_{ij1}))$ where $\hat{\mathbf{X}}_{ij1}$ is the mean of the first learned component of $p(\mathbf{X}_{ij})$. The distributions are learned using $S = 4928$ ground truth motion capture frames of walking data collected at 60 Hz.

We use a standard iterative Expectation-Maximization (EM) algorithm with K-means initialization for learning the Gaussian mixture model (GMM), however care must be taken to estimate the mean and covariance of the quaternion orientations. Given a set of S unit quaternions $Q = \{q_0, q_1, \dots, q_S\}$ where $q_i = [q_{i,x}, q_{i,y}, q_{i,z}, q_{i,w}]$, we extend the approximation presented in [4] to approximate the mean of Q by

$$E_q[Q] \approx \frac{1}{S} \left[\sum_{i=0}^S \frac{q_{i,x}}{q_{i,w}}, \sum_{i=0}^S \frac{q_{i,y}}{q_{i,w}}, \sum_{i=0}^S \frac{q_{i,z}}{q_{i,w}}, S \right]. \quad (4)$$

This approximation suffers from a singularity when any $q_{i,w} \rightarrow 0$. To mitigate the effect of this singularity we compute $E_q[Q]$ in a normalized quaternion space constructed to minimize $\max_{q_i \in Q} (q_i \cdot [0, 0, 0, 1]^T)$. Similarly we approximate the covariance of Q by computing the deviations $\tilde{q}_i = q_i^{-1} * E_q[Q]$ and set $Cov_q[Q] \approx$

$$\frac{1}{S} \sum_{i=0}^S \left(\begin{bmatrix} \tilde{q}_{i,x} & \tilde{q}_{i,y} & \tilde{q}_{i,z} \\ \tilde{q}_{i,w} & \tilde{q}_{i,w} & \tilde{q}_{i,w} \end{bmatrix}^T \begin{bmatrix} \tilde{q}_{i,x} & \tilde{q}_{i,y} & \tilde{q}_{i,z} \\ \tilde{q}_{i,w} & \tilde{q}_{i,w} & \tilde{q}_{i,w} \end{bmatrix} \right). \quad (5)$$

While our learning algorithm is general enough to learn

distributions that have couplings between positional and rotational components of the state space, resulting in block-diagonal covariance matrices, for computational purposes we restrict ourselves to the diagonal-covariance distributions. For sampling and evaluating the probability of the GMM we refer the reader to [4].

Figure 3 shows a few of the learned potential distributions. Samples are shown from several limb-to-limb potentials. For example, the lower leg distribution is shown conditioned on the pose of the upper leg. The proximal end of the shin (green circle) is predicted with high confidence given the thigh location, but there is a wide distribution over possible ankle locations, as expected.

3 Image Likelihoods

The inference algorithm outlined in the next section combines the body model described above with a probabilistic image likelihood model. We define $\phi_i(\mathbf{X}_i)$ to be the likelihood of observing the image measurements conditioned on the pose of limb i . Ideally this model would be robust to partial occlusions, the variability of image statistics across different input sequences, and variability among subjects.

To that end, we combine a variety of cues including multi-scale edge and ridge filters following [20]. However, we explicitly model the conditional dependencies between the various filter responses by learning the joint density using a Gibbs model [19, 28] of the form

$$p(\mathbf{f} | \mathbf{X}_i) \propto \exp \left(- \sum_i \langle \lambda^{(i)}, \xi^{(i)}(\mathbf{f}, \mathbf{X}_i) \rangle \right),$$

where $\mathbf{f}$ represents a vector of filter responses, the $\xi^{(i)}$ are functions selecting various marginals and the $\lambda^{(i)}$ are their learned weights. Since this likelihood is trained also in situations where the limb is partially or fully occluded, it is fairly robust to these conditions.

Separate foreground models are learned for the appearance of each limb. In addition, a pooled background model is learned from non-limb patches sampled from the training images. These are combined into a limb likelihood by taking the likelihood ratio [19]. These likelihood ratios along with background subtraction information are then combined across views, assuming independence of the views conditioned on the limb position.

4 Non-parametric BP

Inferring the body pose in our framework is defined as estimating belief in the graphical model. To cope with the continuous 6D parameter space of each limb, the non-Gaussian

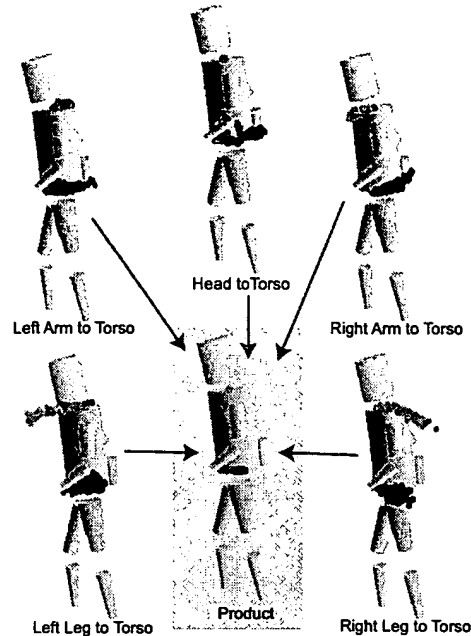


Figure 4: *Message Product*. The head, upper arms, and upper legs send messages to the torso. Samples from these messages are illustrated by showing the predicted torso location with green balls. The distribution over the orientation of the torso is illustrated by showing a red ball at the distal end of the torso for each sample. While any single message represents uncertain information about the torso pose, the product of these messages tightly constrains the torso position and orientation.

conditionals between nodes, and the non-Gaussian likelihood, we use a form of non-parametric belief propagation [13, 25]. The approach is a generalization of particle filtering [6] which allows inference over arbitrary graphs rather than a simple chain. In this generalization the “message” used in standard belief propagation is approximated with a particle set, and the conditional distribution used in standard particle filtering is replaced by a product of incoming message sets. The two formulations [13, 25] have different strengths; we adopt the PAMPAS algorithm [13] because it maps better to our models where the potentials are small mixtures of Gaussians and the likelihoods are simple to evaluate up to an unknown normalization. NBP [25] is more suitable for applications with complex potential functions. We use the Gibbs sampler from [25] to evaluate products of $D > 2$ messages.

The message passing framework is illustrated in Figure 4 where the head, upper arms and upper legs all send messages to the torso. These messages are distributions that are represented by a set of weighted samples as in particle filtering. Belief propagation requires forming the product of these incoming messages. As Figure 4 shows, the individual limbs may not constrain the torso very precisely. The

product over all the incoming messages however produces a very tight distribution over the torso pose.

A message m_{ij} from node $i \rightarrow j$ is written

$$m_{ij}(\mathbf{X}_j) = \int \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) \phi_i(\mathbf{X}_i) \prod_{k \in A_i \setminus j} m_{ki}(\mathbf{X}_i) d\mathbf{X}_i,$$

where A_i is the set of neighbors of node i and $\phi_i(\mathbf{X}_i)$ is the local likelihood associated with node i . The message $m_{ij}(\mathbf{X}_j)$ can be approximated by importance sampling $N' = (N - 1)/M_{ij}$ times from a proposal function $f(\mathbf{X}_i)$, and then doing importance correction. As discussed in [13] the N' samples may be stratified into groups with different proposal functions $f(\cdot)$, so some proportion $\lambda_B N'$ of samples come from the product of all incoming messages A_i into the node, $\lambda_I N'$ come from $A_i \setminus j$ (i.e. A_i excluding j) and $\lambda_S N'$ from an importance function $Q_t(\mathbf{X}_i)$ which is in general a function of the time-step t — we use a limb proposal distribution based on local image measurements described in Section 5. For algorithmic details see [13].

The basic algorithm leaves open the question of what proportions to use in the stratified sampler and what order to update the messages in. We use a 3-frame windowed smoothing algorithm for our tracking results where the estimates at time t are based on observations at times $(t - 1, t, t + 1)$. There are 30 nodes in the graph (10 body parts at each time-step) and 94 edges (18 between adjacent body parts within each time-step and two between each part at each consecutive time-step). All the messages are updated in batch, and this batch update takes place 4 times in 4 belief-propagation iterations. For the first iteration, the proportion of limb proposal samples is $\lambda_S = 0.50$ and this proportion halves for each subsequent iteration. In each iteration the proportion of samples taken from the belief estimate is $\lambda_B = 1/2(1 - \lambda_S)$ and the remainder λ_I are taken from the incoming message product.

The algorithm must sample, evaluate, and take products over Gaussian distributions defined over $SO(3)$ and represented in terms of unit quaternions. We adopt the approximation given in [4] for dealing with rotational distributions by treating the quaternions locally linearly in $\mathbb{R}^4$ — this approximation is only valid for kernels with small rotational covariance and can in principle suffer from singularities if product distributions are widely distributed about the sphere, but we have not encountered problems in practice.

5 Bottom-up Part Detectors

Occlusion of body parts, changes in illumination, and a myriad of other situations may cause a person tracker to lose track of some, or all, parts of a body. We argue that reliable tracking requires bottom-up processes that constantly

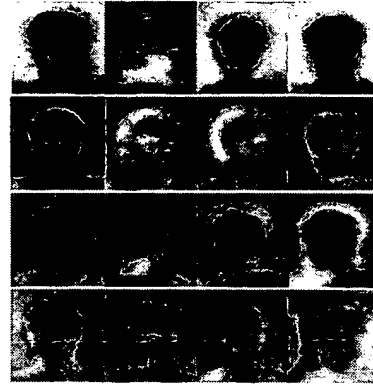


Figure 5: *Multi-view Eigenfeatures*. We learn the correlations between the projections of an object in our fixed cameras by concatenating the four views at each time-step into a single ‘multi-view’ image vector. Top row: mean multi-view head. Next three rows: first three principal components.

search for body parts and suggest their location and pose to the tracker; we call these ‘shouters’².

One expects shouters to be noisy in that they will sometimes fail to detect parts or will find spurious parts. Furthermore they will probably not be able to differentiate between left and right arms. Both of these behaviours can be seen in Figure 6a. Even these noisy ‘guesses’ provide valuable low-level cues, however, and our belief propagation framework is designed to incorporate this bottom-up information in a principled way. As described in Section 4 we use a stratified sampler for the messages to graph node i at time t that draws some samples from an importance function $Q_t(\mathbf{X}_i)$. This importance function is constructed by the node’s shouter process, and draws samples from locations in pose space (3D location and orientation) near the detected body parts.

Multi-View Eigenspaces

There are many approaches to body-part detection in single or multiple images. We have implemented simple eigen-template detectors for the head, upper arms, and lower legs; other shouters could be added as desired. Given calibrated training images with known body-part locations, we build a set of *multi-view* training images. Specifically, we construct a single training sample from the four camera views by concatenating the image regions of the part in each view. We perform PCA on these multi-view images as described in [16], keeping 9–40 principal components, depending on the detector, which describe approximately 80% of the variation in the training data. Figure 5 shows the first few principal components of our head detector model; each part detector is orientation independent.

²This term came from discussions with A. Jepson and D. Fleet.

Using the training data we construct a bounding box in 3-space where we expect each part to appear. We exploit the fact that the background is static, model it using a mixture model at each pixel, and perform standard foreground detection [24]. Selecting any camera view, we examine all foreground pixels within the projected bounding box region for the given body part. Each pixel defines a ray in 3D and we search along this ray for matches to our orientation-independent eigen-model, rejecting any location on the ray which does not project to a foreground pixel in every view. For each of the 10 most probable 3D locations we find the 5 closest multi-view matches from the training set and use their orientations to construct 50 candidate poses for the proposal mixture distribution $Q_t(\mathbf{X}_t)$.

6 Experiments and Evaluation

Figure 6a shows the automatic initialization of the 3D body model using bottom-up part detectors. Note that we use only detectors for the head, upper arms, and calfs and that these detectors are very inaccurate. While they give a rough 3D location and orientation for the limbs, they cannot differentiate left from right limbs reliably. Note also that the right calf was not detected. For body parts with no bottom-up detector, the initialization is random (Figure 6b). After several iterations of belief propagation, the algorithm “finds” the limbs and has a reasonable distribution over the limbs poses (Figure 6c and d). Figure 7 shows the results of tracking over 25 frames after automatic initialization.

There are no standard performance metrics for video-based 3D human tracking. Ramanan and Forsyth [18] report tracking success whenever there is *any* overlap between a limb and the ground truth; this seems overly generous. We propose a quantitative evaluation of accuracy based on the absolute distance of true and estimated marker locations on the limbs. We chose 15 markers corresponding roughly to the locations of the joints and “ends” of the limbs. For each limb we sample from the belief and compute the normalized likelihood of each sample to obtain a weight. These weights are then used to compute an expected absolute distance in *mm* for the markers associated with the limb. The expected deviations are then summed over all defined virtual markers to produce the final distance-based error measure.

Figure 8 shows accuracy of the initialization (left) and tracking results (right). After a few iterations of belief propagation, the initialization error decreases and stays stable. We also observe that the error in the estimated pose increases only slightly over the tracking sequence.

To compare our method against the state of the art, we independently implemented a kinematic tree based tracker that uses annealed particle filtering [5]. This allows a quantitative performance comparison between the methods. Our implementation of [5] uses the same image likelihood as

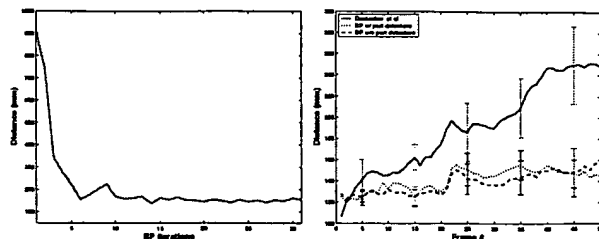


Figure 8: *Quantitative tracking evaluation.* (left) Automatic initialization error as a function of iterations of the belief propagation algorithm. (right) Tracking error over a 50-frame sequence. The dotted black and dashed blue lines show the error for the loose-limbed method with and without shoulders respectively. The red solid line shows the error for the kinematic tree method [5].

used with the loose-limbed model and both trackers are initialized with the same ground-truth pose obtained by the motion capture system. As suggested in [5] the annealing process uses 10 layers and 200 particles. The distance error was computed using the same markers but here we sampled from the particle set, computed the likelihood, and normalized to obtain the weights used in computing the expected absolute marker distance. We found that the performance of the tracker was much poorer than that obtained in [5] and posit that this was due to the more complex image data in our experiments and the lack of contrast between the foreground and background; see Figure 8.

In Figure 8 the performance of the loose-limbed tracker is shown with and without body part detectors. Due to the stochastic nature of the algorithms, the mean and the standard deviation of the error was computed over 10 runs. Initially the kinematic tree model appears more accurate. This is due to the fact that the “ground truth” has the same kinematic structure while the loose-limbed model is able to deviate from that. The annealed particle filter however steadily moves away from the ground truth and becomes lost. Both versions of the loose-limbed tracker outperform the competing algorithm after the first three frames. Loose-limbed tracking with part detection slightly outperforms the tracker that does not rely on the part detection. In most cases the variance of the latter is slightly higher. We predict that this difference would be more significant if we had better part detectors.

7 Conclusion

We present a probabilistic method for fully automatic 3D human detection and tracking. We show that a “loose-limbed” model with continuous-valued parameters can effectively represent a person’s location and pose, and that inference over such a model can be tractably performed using belief propagation over particle sets. The belief prop-

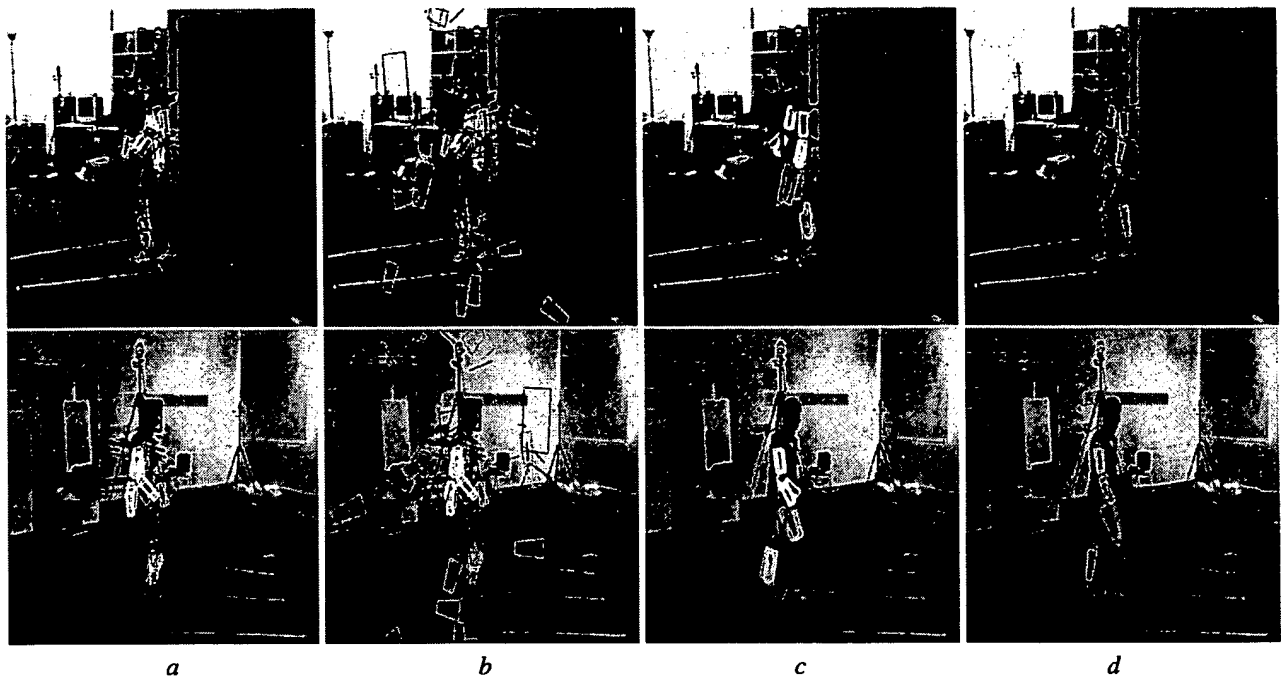


Figure 6: **Automatic Initialization from bottom-up detectors.** Each row corresponds to a different camera view (2 of the 4 views shown). The initialization used two frames but only the first frame is shown for brevity. (a) Samples from the shoulder proposal distribution (head, upper arms, calves only). Notice that they completely fail to detect one calf while the other is detected by both left and right calf detectors. The upper arms are found but the edge is not well determined. The head is well localized. (b) Samples from the full proposal distribution. The body parts not constrained by shoulders are sampled from a uniform distribution in position and orientation. (c) Samples from the belief distribution after 30 iterations of belief propagation. (d) Most likely limb poses after 30 iterations.

agation framework allows us to avoid distinguishing between initialization and tracking, but instead to use bottom-up part detectors to stabilize the motion estimation and provide “initialization” cues at every time-step.

The main advantages of our approach are: the complexity of the search task is linear rather than exponential in the number of body parts; bottom-up processes are integrated at every frame allowing automatic initialization and recovery from transient tracking failures; the conditional probabilities between limbs in space and time are learned from training data; a novel Gibbs likelihood model is learned from training data and models conditional dependencies between image measurements (filter responses); and forward-backward smoothing, either over a time-window or an entire sequence, is straightforward. Additionally, we exploit a novel data set with synchronized 3D “ground truth” and video data for the quantitative evaluation of performance. We also compared our method with the state of the art as proposed in [5].

While our preliminary results are promising, details of our implementation could be improved. We have used simple detectors for the head and limbs, but we expect that hands and feet would provide valuable additional cues. Other machine learning methods such as AdaBoost [26],

might prove faster and more robust than the eigenfeatures we adopt. For greater applicability the method must be extended to use monocular image data and to allow a moving camera.

There are also limitations imposed by our use of a loose-limbed model. Since we assume independence of, for example, the left and right arms conditional on the torso location, it is cumbersome to fully avoid poses where one limb penetrates another. These problems are much easier to address with a full kinematic tree body model, and therefore one might think of the loose-limbed model as an intermediate stage between the bottom-up part detectors and a full kinematic model. The details and implementation of such a scheme are postponed to future research.

References

- [1] M. Burl, M. Weber and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry, *ECCV*, pp. 628–641, 1998.
- [2] C. Bregler and J. Malik. Tracking people with twists and exponential maps, *CVPR*, pp. 8–15, 1998.
- [3] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation, *ECCV* 3:453–468, 2002.

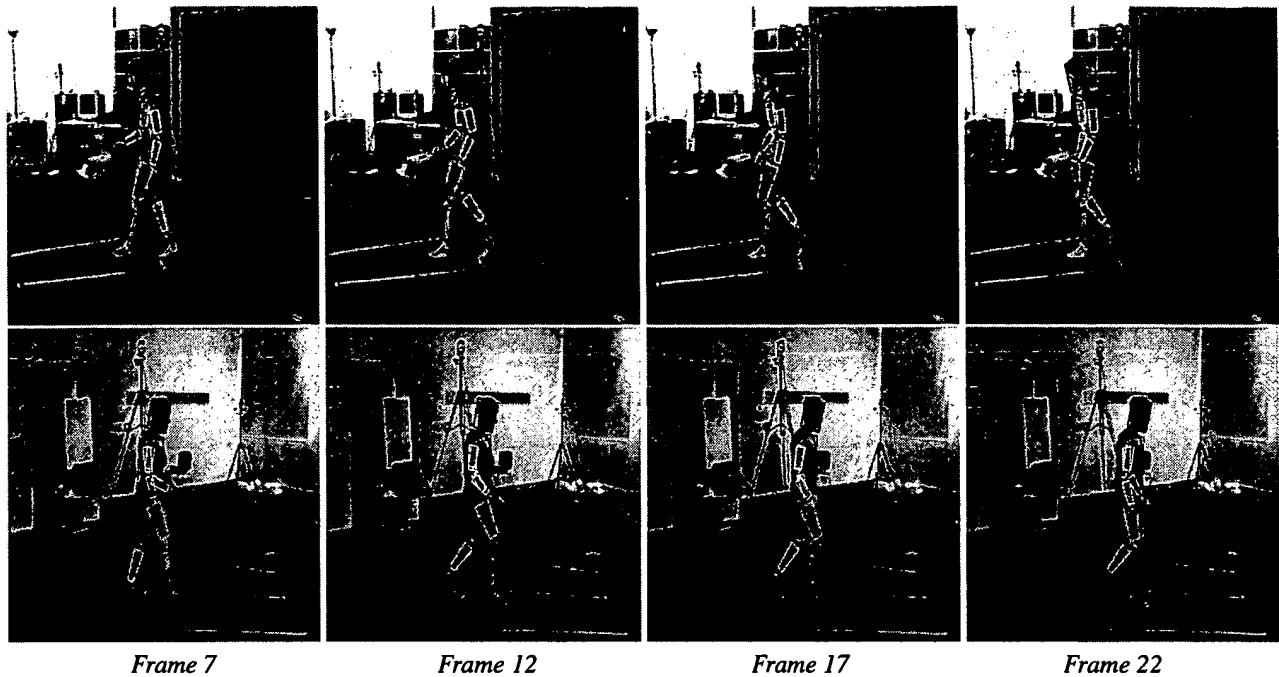


Figure 7: *Loose-limbed tracking*. The body model whose automatic initialization is shown in Figure 6 is tracked for a further 25 frames. Four sample frames are shown from 2 of the 4 camera views used.

- [4] J. Deutscher, M. Isard and J. MacCormick. Automatic camera calibration from a single manhattan image, *ECCV*, pp. 175–188, 2002.
- [5] J. Deutscher, A. Blake and I. Reid. Articulated body motion capture by annealed particle filtering, *CVPR*, II:126–133, 2000.
- [6] A. Doucet, N. de Freitas and N. Gordon. Sequential Monte Carlo methods in practice, *Stats. for Eng. and Info. Sciences*, Springer Verlag, 2001.
- [7] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures, *CVPR*, 2:66–73, 2000.
- [8] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22(1), 1973.
- [9] A.T. Ihler, E.B. Sudderth, W.T. Freeman and A.S. Willsky. Efficient Multiscale Sampling from Products of Gaussian Mixtures, *NIPS*, 2003.
- [10] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees, *ICCV*, I:690–695, 2001.
- [11] S. Ioffe and D. Forsyth. Probabilistic methods for finding people, *IJCV* 43(1):45–68, 2001.
- [12] S. Ioffe and D. Forsyth. Finding people by sampling, *ICCV*, pp. 1092–1097, 1999.
- [13] M. Isard. PAMPAS: Real-valued graphical models for computer vision, *CVPR*, I:613–620, 2003.
- [14] S. Ju, M. Black and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. *Int. Conf. on Automatic Face and Gesture Recognition*, pp. 38–44, 1996.
- [15] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. *ECCV* 2:3–19, 2000,
- [16] Moghaddam B. and Pentland A., Probabilistic Visual Learning for Object Representation, *PAMI* 19(7):696–710, 1997.
- [17] V. Pavolvić, J. Rehg, T-J. Cham and K. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models, *ICCV*, pp. 94–101, 1999.
- [18] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up, *CVPR*, II:467–474, 2003.
- [19] S. Roth, L. Sigal, and M. Black. Gibbs likelihoods for Bayesian tracking, *CVPR*, 2004.
- [20] H. Sidenbladh and M. Black. Learning the statistics of people in images and video, *IJCV* 54(1–3):183–209, 2003.
- [21] H. Sidenbladh, M. Black and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion, *ECCV*, 2:702–718, 2000.
- [22] L. Sigal, M. Isard, B.H. Sigelman, M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation, *NIPS*, 2003.
- [23] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking, *CVPR*, 1:447–454, 2001.
- [24] C. Stauffer and W. Grimson. “Adaptive background mixture models for real-time tracking.” *CVPR*, pp. 246–252, 1999.
- [25] E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, *CVPR*, I:605–612, 2003.
- [26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, *CVPR*, I:511–518, 2001.
- [27] Y. Wu, G. Hua, T. Yu. Tracking articulated body by dynamic Markov network, *ICCV*, pp. 1094–1101, 2003.
- [28] S. Zhu, Y. Wu, and D. Mumford. FRAME: Filters, random field and maximum entropy: Towards a unified theory for texture modeling. *PAMI*, 27(2):1–20, 1998.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

A Boosted Particle Filter: Multitarget Detection and Tracking

Kenji Okuma, Ali Taleghani, Nando De Freitas,
James J. Little, and David G. Lowe

University of British Columbia, Vancouver B.C V6T 1Z4, CANADA,
okumak@cs.ubc.ca,
WWW home page: <http://www.cs.ubc.ca/~okumak>

Abstract. The problem of tracking a varying number of non-rigid objects has two major difficulties. First, the observation models and target distributions can be highly non-linear and non-Gaussian. Second, the presence of a large, varying number of objects creates complex interactions with overlap and ambiguities. To surmount these difficulties, we introduce a vision system that is capable of learning, detecting and tracking the objects of interest. The system is demonstrated in the context of tracking hockey players using video sequences. Our approach combines the strengths of two successful algorithms: mixture particle filters and Adaboost. The mixture particle filter [17] is ideally suited to multi-target tracking as it assigns a mixture component to each player. The crucial design issues in mixture particle filters are the choice of the proposal distribution and the treatment of objects leaving and entering the scene. Here, we construct the proposal distribution using a mixture model that incorporates information from the dynamic models of each player and the detection hypotheses generated by Adaboost. The learned Adaboost proposal distribution allows us to quickly detect players entering the scene, while the filtering process enables us to keep track of the individual players. The result of interleaving Adaboost with mixture particle filters is a simple, yet powerful and fully automatic multiple object tracking system.

1 Introduction

Automated tracking of multiple objects is still an open problem in many settings, including car surveillance [10], sports [12, 13] and smart rooms [6] among many others [5, 7, 11]. In general, the problem of tracking visual features in complex environments is fraught with uncertainty [6]. It is therefore essential to adopt principled probabilistic models with the capability of learning and detecting the objects of interest. In this work, we introduce such models to attack the problem of tracking a varying number of hockey players on a sequence of digitized video from TV.

Over the last few years, particle filters, also known as condensation or sequential Monte Carlo, have proved to be powerful tools for image tracking [3, 8, 14, 15]. The strength of these methods lies in their simplicity, flexibility, and systematic treatment of nonlinearity and non-Gaussianity.

Various researchers have attempted to extend particle filters to multi-target tracking. Among others, Hue *et. al* [5] developed a system for multitarget tracking by expanding the state dimension to include component information, assigned by a Gibbs sampler. They assumed a fixed number of objects. To manage a varying number of objects efficiently, it is important to have an automatic detection process. The Bayesian Multiple-Blob tracker (BraMBLe) [7] is an important step in this direction. BraMBLe has an automatic object detection system that relies on modeling a fixed background. It uses this model to identify foreground objects (targets). In this paper, we will relax this assumption of a fixed background in order to deal with realistic TV video sequences, where the background changes.

As pointed out in [17], particle filters may perform poorly when the posterior is multi-modal as the result of ambiguities or multiple targets. To circumvent this problem, Vermaak *et al* introduce a mixture particle filter (MPF), where each component (mode or, in our case, hockey player) is modelled with an individual particle filter that forms part of the mixture. The filters in the mixture interact only through the computation of the importance weights. By distributing the resampling step to individual filters, one avoids the well known problem of sample depletion, which is largely responsible for loss of track [17].

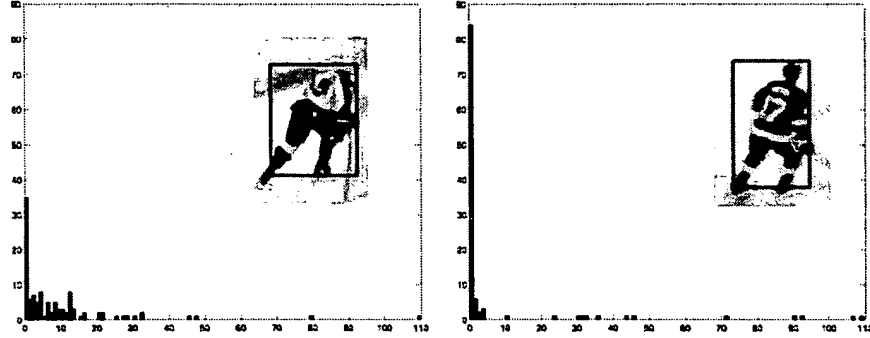
In this paper, we extend the approach of Vermaak *et al*. In particular, we use a cascaded Adaboost algorithm [18] to learn models of the hockey players. These detection models are used to guide the particle filter. The proposal distribution consists of a probabilistic mixture model that incorporates information from Adaboost and the dynamic models of the individual players. This enables us to quickly detect and track players in a dynamically changing background, despite the fact that the players enter and leave the scene frequently. We call the resulting algorithm the Boosted Particle Filter (BPF).

2 Statistical Model

In non-Gaussian state-space models, the state sequence $\{\mathbf{x}_t; t \in \mathbb{N}\}$, $\mathbf{x}_t \in \mathbb{R}^{n_x}$, is assumed to be an unobserved (hidden) Markov process with initial distribution $p(\mathbf{x}_0)$ and transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, where n_x is the dimension of the state vector. In our tracking system, this transition model corresponds to a standard autoregressive dynamic model. The observations $\{\mathbf{y}_t; t \in \mathbb{N}^*\}$, $\mathbf{y}_t \in \mathbb{R}^{n_y}$, are conditionally independent given the process $\{\mathbf{x}_t; t \in \mathbb{N}\}$ with marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$, where n_y is the dimension of the observation vector.

2.1 Observation model

Following [14], we adopt a multi-color observation model based on Hue-Saturation-Value (HSV) color histograms. Since HSV decouples the intensity (i.e., value) from color (i.e., hue and saturation), it is reasonably insensitive to illumination effects. An HSV histogram is composed of $N = N_h N_s + N_v$ bins and we denote



Color histogram of a player (LEFT: white uniform RIGHT: red uniform)

Fig. 1. Color histograms: This figure shows two color histograms of selected rectangular regions, each of which is from a different region of the image. The player on left has uniform whose color is the combination of dark blue and white and the player on right has a red uniform. One can clearly see concentrations of color bins due to limited number of colors. In (a) and (b), we set the number of bins, $N = 110$, where N_h , N_s , and N_v are set to 10.

$b_t(\mathbf{d}) \in \{1, \dots, N\}$ as the bin index associated with the color vector $\mathbf{y}_t(\mathbf{k})$ at a pixel location $\mathbf{d}$ at time t . Figure 1 shows two instances of the color histogram. If we define the candidate region in which we formulate the HSV histogram as $R(\mathbf{x}_t) \triangleq \mathbf{l}_t + s_t W$, then a kernel density estimate $\mathbf{K}(\mathbf{x}_t) \triangleq \{k(n; \mathbf{x}_t)\}_{n=1, \dots, N}$ of the color distribution at time t is given by [1, 14]:

$$k(n; \mathbf{x}_t) = \eta \sum_{\mathbf{d} \in R(\mathbf{x}_t)} \delta[b_t(\mathbf{d}) - n] \quad (1)$$

where δ is the delta function, η is a normalizing constant which ensures k to be a probability distribution, $\sum_{n=1}^N k(n; \mathbf{x}_t) = 1$, and a location $\mathbf{d}$ could be any pixel location within $R(\mathbf{x}_t)$. Eq. (1) defines $k(n; \mathbf{x}_t)$ as the probability of a color bin n at time t .

If we denote $\mathbf{K}^* = \{k^*(n; \mathbf{x}_0)\}_{n=1, \dots, N}$ as the reference color model and $\mathbf{K}(\mathbf{x}_t)$ as a candidate color model, then we need to measure the data likelihood (i.e., similarity) between $\mathbf{K}^*$ and $\mathbf{K}(\mathbf{x}_t)$. As in [1, 14], we apply the Bhattacharyya similarity coefficient to define a distance ξ on HSV histograms. The mathematical formulation of this measure is given by [1]:

$$\xi[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)] = \left[1 - \sum_{n=1}^N \sqrt{k^*(n; \mathbf{x}_0) k(n; \mathbf{x}_t)} \right]^{\frac{1}{2}} \quad (2)$$

Statistical properties of near optimality and scale invariance presented in [1] ensure that the Bhattacharyya coefficient is an appropriate choice of measuring similarity of color histograms. Once we obtain a distance ξ on the HSV color

histograms, we use the following likelihood distribution given by [14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{-\lambda \xi^2[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)]} \quad (3)$$

where $\lambda = 20$. λ is suggested in [14, 17], and confirmed also on our experiments. Also, we set the size of bins N_h , N_s , and N_v as 10.

The HSV color histogram is a reliable approximation of the color density on the tracked region. However, a better approximation is obtained when we consider the spatial layout of the color distribution. If we define the tracked region as the sum of r sub-regions $R(\mathbf{x}_t) = \sum_{j=1}^r R_j(\mathbf{x}_t)$, then we apply the likelihood as the sum of the reference histograms $\{k_j^*\}_{j=1, \dots, r}$ associated with each sub-region by [14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{\sum_{j=1}^r -\lambda \xi^2[\mathbf{K}_j^*, \mathbf{K}_j(\mathbf{x}_t)]} \quad (4)$$

Eq. (4) shows how the spatial layout of the color is incorporated into the data likelihood. In Figure 2, we divide up the tracked regions into two sub-regions in order to use spatial information of the color in the appearance of a hockey player.

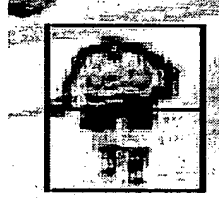


Fig. 2. Multi-part color likelihood model: This figure shows our multi-part color likelihood model. We divide our model into two sub-regions and take a color histogram from each sub-region so that we take into account the spatial layout of colors of two sub-regions.

For hockey players, their uniforms usually have a different color on their jacket and their pants and the spatial relationship of different colors becomes important.

2.2 The Filtering Distribution

We denote the state vectors and observation vectors up to time t by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0 \dots \mathbf{x}_t\}$ and $\mathbf{y}_{0:t}$. Given the observation and transition models, the solution to the filtering problem is given by the following Bayesian recursion [3]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t} \end{aligned} \quad (5)$$

To deal with multiple targets, we adopt the mixture approach of [17]. The posterior distribution, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, is modelled as an M -component non-parametric mixture model:

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \sum_{j=1}^M \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \quad (6)$$

where the mixture weights satisfy $\sum_{m=1}^M \Pi_{m,t} = 1$. Using the the filtering distribution, $p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$, computed in the previous step, the predictive distribution becomes

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \sum_{j=1}^M \Pi_{j,t-1} p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \quad (7)$$

where $p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1}) d\mathbf{x}_{t-1}$. Hence, the updated posterior mixture takes the form

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{\sum_{j=1}^M \Pi_{j,t-1} p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \\ &= \sum_{j=1}^M \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right] \\ &\quad \times \left[\frac{p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right] \\ &= \sum_{j=1}^M \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \end{aligned} \quad (8)$$

where the new weights (independent of $\mathbf{x}_t$) are given by:

$$\Pi_{j,t} = \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t}{\sum_{k=1}^M \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) d\mathbf{x}_t} \right]$$

Unlike a mixture particle filter by [17], we have M different likelihood distributions, $\{p_j(\mathbf{y}_t|\mathbf{x}_t)\}_{j=1 \dots M}$. When one or more new objects appear in the scene, they are detected by Adaboost and automatically initialized with an observation model. Using a different color-based observation model allows us to track different colored objects.

3 Particle Filtering

In standard particle filtering, we approximate the posterior $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ with a Dirac measure using a finite set of N particles $\{\mathbf{x}_t^i\}_{i=1 \dots N}$. To accomplish this, we sample candidate particles from an appropriate proposal distribution $\tilde{\mathbf{x}}_t^i \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ (In the simplest scenario, it is set as $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$,

yielding the bootstrap filter [3]) and weight these particles according to the following importance ratio:

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t | \tilde{\mathbf{x}}_t^i) p(\tilde{\mathbf{x}}_t^i | \mathbf{x}_{t-1}^i)}{q(\tilde{\mathbf{x}}_t^i | \mathbf{x}_{0:t-1}^i, \mathbf{y}_{0:t})} \quad (9)$$

We resample the particles using their importance weights to generate an un-weighted approximation of $p(\mathbf{x}_t | \mathbf{y}_{0:t})$. In the mixture approach of [17], the particles are used to obtain the following approximation of the posterior distribution:

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \approx \sum_{j=1}^M \Pi_{j,t} \sum_{i \in \mathcal{I}_j} w_t^i \delta_{\mathbf{x}_t^i}(\mathbf{x}_t)$$

where $\mathcal{I}_j$ is the set of indices of the particles belonging to the j -th mixture component. As with many particle filters, the algorithm simply proceeds by sampling from the transition priors (note that this proposal does not use the information in the data) and updating the particles using importance weights derived from equation (8); see Section 3 of [17] for details.

In [17], the mixture representation is obtained and maintained using a simple K-means spatial reclustering algorithm. In the following section, we argue that boosting provides a more satisfactory solution to this problem.

4 Boosted Particle Filter

The boosted particle filter introduces two important extensions of the MPF. First, it uses Adaboost in the construction of the proposal distribution. This improves the robustness of the algorithm substantially. It is widely accepted that proposal distributions that incorporate the recent observations (in our case, through the Adaboost detections) outperform naive transition prior proposals considerably [15, 16]. Second, Adaboost provides a mechanism for obtaining and maintaining the mixture representation. This approach is again more powerful than the naive K-means clustering scheme used for this purpose in [17]. In particular, it allows us to detect objects leaving and entering the scene efficiently.

4.1 Adaboost Detection

We adopt the cascaded Adaboost algorithm of Viola and Jones [18], originally developed for detecting faces. In our experiments, a 23 layer cascaded classifier is trained to detect hockey players. In order to train the detector, a total of 6000 figures of hockey players are used. These figures are scaled to have a resolution of 10×24 pixels. In order to generate such data in a limited amount of time, we speed the selection process by using a program to extract small regions of the image that most likely contain hockey players. Simply, the program finds regions that are centered on low intensities (i.e., hockey players) and surrounded by high intensities (i.e., rink surface). However, it is important to note that the

data generated by such a simple script is not ideal for training Adaboost, as shown in Figure 3. As a result, our trained Adaboost produces false positives alongside the edge of the rink shown in (c) of Figure 4. More human intervention with a larger training set would lead to better Adaboost results, although failures would still be expected in regions of clutter and overlap. The non-hockey-player subwindows used to train the detector are generated from 100 images manually chosen to contain nothing but the hockey rink and audience. Since our tracker is implemented for tracking hockey scenes, there is no need to include training images from outside the hockey domain.



Fig. 3. Training set of images for hockey players: This figure shows a part of the training data. A total of 6000 different figures of hockey players are used for the training.

The results of using Adaboost in our dataset are shown in Figure 4. Adaboost performs well at detecting the players but often gets confused and leads to many false positives.

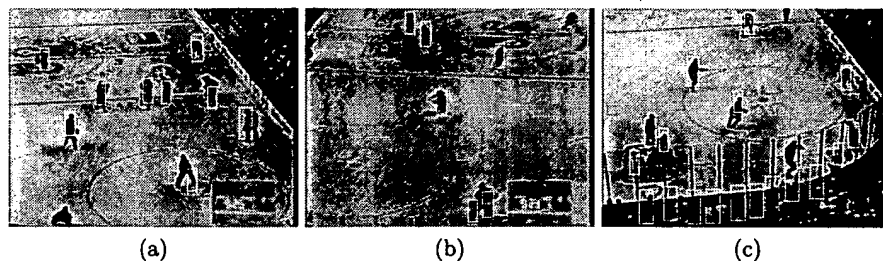


Fig. 4. Hockey player detection result: This figure shows results of the Adaboost hockey detector. (a) and (b) shows mostly accurate detections. In (c), there are a set of false positives detected on audience by the rink.

4.2 Incorporating Adaboost in the Proposal Distribution

It is clear from the Adaboost results that they could be improved if we considered the motion models of the players. In particular, by considering plausible motions, the number of false positives could be reduced. For this reason, we incorporate Adaboost in the proposal mechanism of our MPF. The expression for the proposal distribution is given by the following mixture.

$$q_B^*(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = \alpha q_{ada}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) + (1 - \alpha)p(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (10)$$

where q_{ada} is a Gaussian distribution that we discuss in the subsequent paragraph (See Figure 5). The parameter α can be set dynamically without affecting the convergence of the particle filter (it is only a parameter of the proposal distribution and therefore its influence is corrected in the calculation of the importance weights). When $\alpha = 0$, our algorithm reduces to the MPF of [17]. By increasing α we place more importance on the Adaboost detections. We can adapt the value of α depending on tracking situations, including cross overs, collisions and occlusions.

Note that the Adaboost proposal mechanism depends on the current observation $\mathbf{y}_t$. It is, therefore, robust to peaked likelihoods. Still, there is another critical issue to be discussed: determining how close two different proposal distributions need to be for creating their mixture proposal. We can always apply the mixture proposal when q_{ada} is overlaid on a transition distribution modeled by autoregressive state dynamics. However, if these two different distributions are not overlapped, there is a distance between the mean of these distributions.

If a Monte Carlo estimation of a mixture component by a mixture particle filter overlaps with the nearest cluster given by the Adaboost detection algorithm, we sample from the mixture proposal distribution. If there is no overlap between the Monte Carlo estimation of a mixture particle filter for each mixture component and clusters given by the Adaboost detection, then we set $\alpha = 0$ so that our proposal distribution takes only a transition distribution of a mixture particle filter.

5 Experiments

This section shows BPF tracking results on hockey players in a digitized video sequence from broadcast television. The experiments are performed using our non-optimized implementation in C on a 2.8 GHz Pentium IV.

Figure 6 shows scenes where objects are coming in and out. In this figure, it is important to note that no matter how many objects are in the scene, the mixture representation of BPF is not affected and successfully adapts to the change. For objects coming into the scene, in (a) of the figure, Adaboost quickly detects a new object in the scene within a short time sequence of only two frames. Then BPF immediately assigns particles to an object and starts tracking it.

Figure 7 shows the Adaboost detection result in the left column, a frame number in the middle, and BPF tracking results on the right. In Frame 32

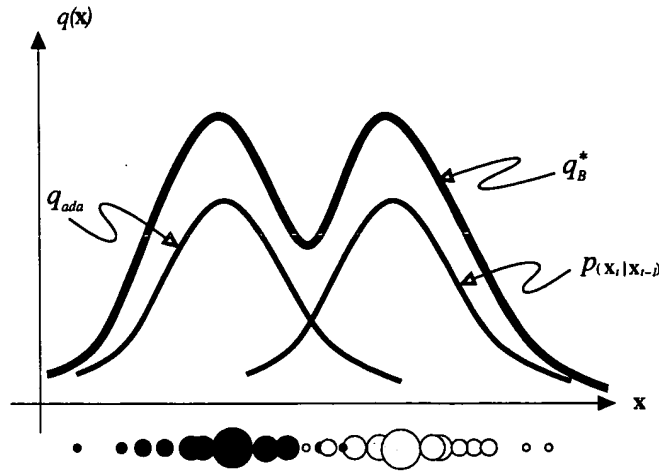


Fig. 5. Mixture of Gaussians for the proposal distribution.

and 33, a small crowd of three players in the middle of the center circle on the rink are not well detected by Adaboost. This is an example of the case in which Adaboost does not work well in clutter. There are only two windows detected by Adaboost. However, BPF successfully tracks three modes in the cluttered environment. Frames 34 and 35 show another case when Adaboost fails. Since Adaboost features are based on different configurations of intensity regions, it is highly sensitive to a drastic increase/decrease of the intensity. The average intensity of the image is clearly changed by a camera flash in Frame 34. The number of Adaboost detections is much smaller in a comparison with the other two consecutive frames. However, even in the case of an Adaboost failure, mixture components are well maintained by BPF, which is also clearly shown in Frame 205 in the figure.

6 Conclusions

We have described an approach to combining the strengths of Adaboost for object detection with those of mixture particle filters for multiple-object tracking. The combination is achieved by forming the proposal distribution for the particle filter from a mixture of the Adaboost detections in the current frame and the dynamic model predicted from the previous time step. The combination of the two approaches leads to fewer failures than either one on its own, as well as addressing both detection and consistent track formation in the same framework.

We have experimented with this boosted particle filter in the context of tracking hockey players in video from broadcast television. The results show that most players are successfully detected and tracked, even as players move in and out of the scene. We believe results can be further improved by including

more examples in our Adaboost training data for players that are seen against non-uniform backgrounds. Further improvements could be obtained by dynamic adjustment of the weighting parameter selecting between the Adaboost and dynamic model components of the proposal distribution. Adopting a probabilistic model for target exclusion [11] may also improve our BPF tracking.

7 Acknowledgment

This research is funded by the Institute for Robotics and Intelligent Systems (IRIS) and their support is gratefully acknowledged. The authors would like to specially thank Matthew Brown, Jesse Hoey, and Don Murray from the University of British Columbia for fruitful discussions and helpful comments about the formulation of BPF.

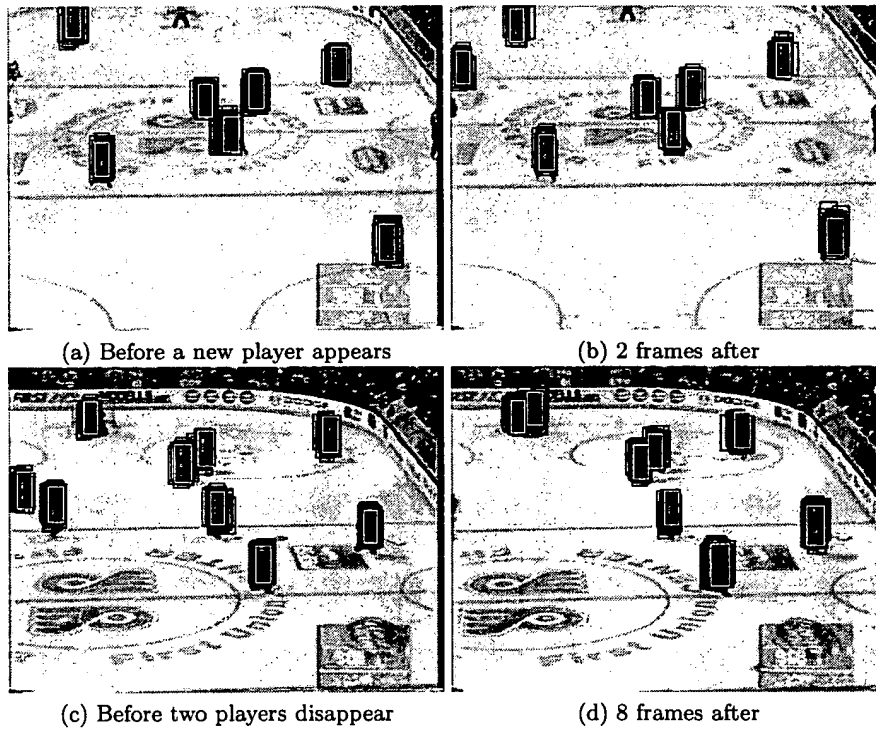
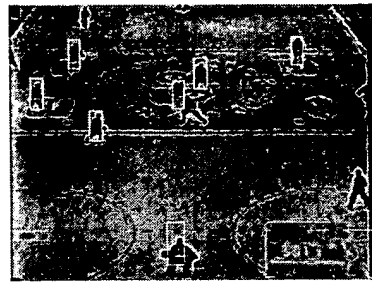
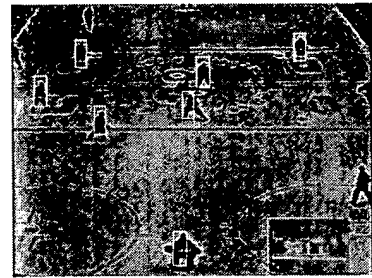


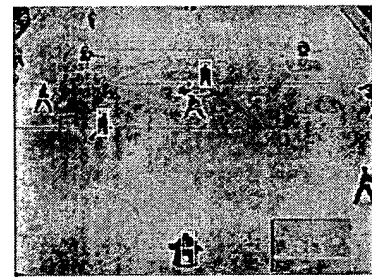
Fig. 6. Objects appearing and disappearing from the scene: This is a demonstration of how well BPF handles objects randomly coming in and out of the scene. (a) and (b) show that a new player that appears in the top left corner of the image is successfully detected and starts to get tracked. (c) and (d) show that two players are disappearing from the scene to left.



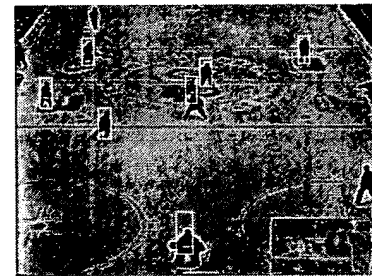
32



33



34

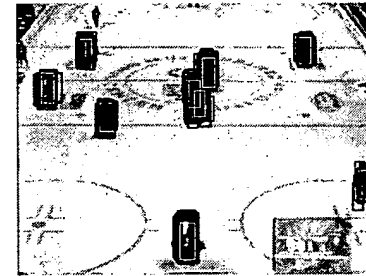
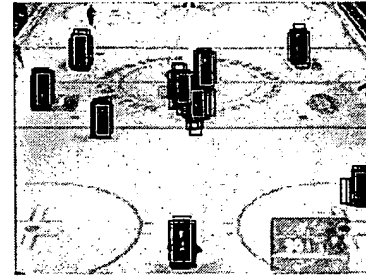
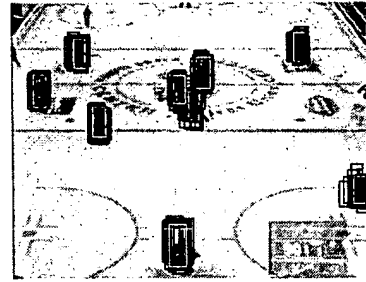


35



205

(a) Adaboost detection result



(b) BPF tracking

Fig. 7. BPF tracking result: The results of Adaboost detection are shown on the left, with the corresponding boosted particle filter results on the right.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-151 (2000)
2. Deutscher, J., Blake, A., Ried, I.: Articulated body motion capture by annealed particle filtering. IEEE Conference on Computer Vision and Pattern Recognition, (2000)
3. Doucet, A., de Freitas, J. F. G., N. J. Gordon, editors: Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
4. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory, pp. 23-37, Springer-Verlag, (1995)
5. Hue, C., Le Cadre, J.-P., Pérez, P.: Tracking Multiple Objects with Particle Filtering. IEEE Transactions on Aerospace and Electronic Systems, 38(3):791-812 (2002)
6. Intille, S. S., Davis, J. W., Bobick, A.F.: Real-Time Closed-World Tracking. IEEE Conference on Computer Vision and Pattern Recognition, pp. 697-703 (1997)
7. Isard, M., MacCormick, J.: BraMBLe: A Bayesian multiple-blob tracker. International Conference on Computer Vision, pp. 34-41(2001)
8. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal on Computer Vision, 28(1):5-28 (1998)
9. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems Transactions of the ASME–Journal of Basic Engineering, vol.82 Series D pp.35-45 (1960)
10. Koller, D., Weber, J., Malik, J.: Robust Multiple Car Tracking with Occlusion Reasoning. European Conference on Computer Vision, pp. 186-196, LNCS 800, Springer-Verlag (1994)
11. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. International Conference on Computer Vision, pp. 572-578 (1999)
12. Misu, T., Naemura, M., Wentao Zheng, Izumi, Y., Fukui, K.: Robust Tracking of Soccer Players Based on Data Fusion IEEE 16th International Conference on Pattern Recognition, pp. 556-561 vol.1 (2002)
13. Needham, C. J., Boyle, R. D.: Tracking multiple sports players through occlusion, congestion and scale. British Machine Vision Conference, vol. 1, pp. 93-102 BMVA (2001)
14. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. European Conference on Computer Vision, (2002)
15. Rui, Y., Chen, Y.: Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. IEEE Conference on Computer Vision and Pattern Recognition, pp. 786-793 (2001)
16. van der Merwe, R., Doucet, A., de Freitas, J. F. G., Wan, E: The Unscented Particle Filter. Advances in Neural Information Processing Systems, vol. 8 pp 351-357 (2000)
17. Vermaak, J., Doucet, A., Pérez, P.: Maintaining Multi-Modality through Mixture Tracking. International Conference on Computer Vision (2003)
18. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition (2001)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Nonparametric Belief Propagation

Erik B. Sudderth, Alexander T. Ihler,
William T. Freeman, and Alan S. Willsky

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

esuddert@mit.edu, ihler@mit.edu, wtf@ai.mit.edu, willsky@mit.edu

Abstract

In applications of graphical models arising in fields such as computer vision, the hidden variables of interest are most naturally specified by continuous, non-Gaussian distributions. However, due to the limitations of existing inference algorithms, it is often necessary to form coarse, discrete approximations to such models. In this paper, we develop a *nonparametric belief propagation* (NBP) algorithm, which uses stochastic methods to propagate kernel-based approximations to the true continuous messages. Each NBP message update is based on an efficient sampling procedure which can accommodate an extremely broad class of potential functions, allowing easy adaptation to new application areas. We validate our method using comparisons to continuous BP for Gaussian networks, and an application to the stereo vision problem.

1 Introduction

Graphical models provide an efficient, powerful framework for modeling probabilistic relationships. Most existing applications of graphical models employ hidden variables which take values from a discrete, finite set. In some fields, such as error correcting codes [1], these discrete models are entirely appropriate. However, in areas such as computer vision [2–5], they typically arise from quantizations of continuous variables. Computational costs invariably limit the accuracy of such quantizations, leading to biases and artifacts in the resulting estimates.

The popularity of discrete graphical models can be traced to the existence of efficient algorithms for solving learning and inference problems [6]. For discrete models, the difficulties associated with inference are purely *computational*, and involve finding ways to efficiently decompose and/or approximate sums of exponentially many terms. Although exact inference in general discrete graphs is NP hard [7], approximate inference algorithms such as loopy belief propagation (BP) [1, 8, 9] have been shown to produce good results for a wide range of interesting models [2, 5].

For graphical models containing continuous hidden variables, there is an additional *representational* challenge. The conditional distributions sought by inference algorithms are not finite vectors but continuous functions. Unfortunately, even if the

potential functions specifying the graphical model have simple parametric forms, the integral equations used to propagate information across the graph rarely have tractable analytic solutions [10]. A notable exception occurs when all variables are jointly Gaussian, so that distributions may be parameterized by their mean and covariance [9]. However, many real-world phenomena exhibit significant outliers, bimodalities, and other statistical features which are far from Gaussian.

Since the conditional densities arising in non-Gaussian graphical models lack tractable parametric forms, it is natural to consider nonparametric approximations. For temporal inference problems defined on Markov chains, a variety of density representations have been explored. Gaussian sum filters [11] use local linearizations to deterministically propagate Gaussian mixture models. In contrast, particle filters [12] use Monte Carlo methods to stochastically update a set of weighted point samples. The stability and robustness of particle filters can often be improved by regularization methods [12, Chapter 12] in which smoothing kernels [13, 14] explicitly represent the uncertainty associated with each point sample.

For more general graphs, the junction tree representation [15] has been used to develop structured approximate inference techniques. A wide variety of algorithms can be specified by combining an approximate clique variable representation with local methods for updating these approximations [16, 17]. For example, distributions over large cliques of discrete variables can be approximated by a set of weighted point samples, and then related to neighboring nodes using standard message-passing recursions [18, 19]. Koller et al. [17] propose a more sophisticated framework in which the current clique potential estimate is used to guide message computations, allowing approximations to be gradually refined over successive iterations. However, the sample algorithm they provide is limited to networks containing mixtures of discrete and Gaussian variables. In addition, for many graphs (e.g. nearest-neighbor grids) the size of the junction tree's largest cliques grows exponentially with problem size, requiring the estimation of extremely high-dimensional distributions.

The *nonparametric belief propagation* (NBP) algorithm we develop in this paper extends this earlier work in two key ways. First, for graphs with cycles we do *not* form a junction tree, but instead iterate our local message updates until convergence as in loopy BP. This has the advantage of greatly reducing the dimensionality of the spaces over which we must infer distributions. In addition, we provide a message update algorithm specifically adapted to graphs containing continuous, non-Gaussian potentials. Each message is represented by a kernel-based nonparametric density estimate, and message products are found by an efficient *local* Gibbs sampling algorithm. We validate the NBP algorithm on a small Gaussian network, and present stereo vision results demonstrating its effectiveness.

2 Undirected Graphical Models

An undirected graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}$, and a corresponding set of edges $\mathcal{E}$. The *neighborhood* of a node $s \in \mathcal{V}$ is defined as $\Gamma(s) \triangleq \{t | (s, t) \in \mathcal{E}\}$, the set of all nodes which are directly connected to s . Graphical models associate each node $s \in \mathcal{V}$ with an unobserved, or hidden, random variable x_s , as well as a noisy local observation y_s . Let $x = \{x_s\}_{s \in \mathcal{V}}$ and $y = \{y_s\}_{s \in \mathcal{V}}$ denote the sets of all hidden and observed variables, respectively. To simplify the presentation, we consider models with pairwise potential functions, for which $p(x, y)$ factorizes as

$$p(x, y) = \frac{1}{Z} \prod_{(s, t) \in \mathcal{E}} \psi_{s, t}(x_s, x_t) \prod_{s \in \mathcal{V}} \psi_s(x_s, y_s) \quad (1)$$

However, the nonparametric updates we present may be directly extended to models with higher-order potential functions.

In this paper, we focus on the calculation of the conditional marginal distributions $p(x_s | y)$ for all nodes $s \in \mathcal{V}$. These distributions may be used to calculate the best estimates of the hidden variables x_s relative to a wide range of criteria. For example, the conditional mean $E[x_s | y]$ provides the Bayes' least squares estimate. More generally, conditional distributions are useful because they provide information about the degree of uncertainty in the estimate of each hidden node.

2.1 Belief Propagation

For graphs which are acyclic or tree-structured, the desired conditional distributions $p(x_s | y)$ can be directly calculated by a local message-passing algorithm known as *belief propagation* (BP) [1, 6]. At iteration n of the BP algorithm, each node $t \in \mathcal{V}$ calculates a message $m_{ts}^n(x_s)$ to be sent to each neighboring node $s \in \Gamma(t)$:

$$m_{ts}^n(x_s) = \alpha \int_{x_t} \psi_{s,t}(x_s, x_t) \psi_t(x_t, y_t) \prod_{u \in \Gamma(t) \setminus s} m_{ut}^{n-1}(x_t) dx_t \quad (2)$$

Here, α denotes an arbitrary proportionality constant. At any iteration, each node can produce an approximation $\hat{p}^n(x_s | y)$ to the marginal distributions $p(x_s | y)$ by combining the incoming messages with the local observation potential:

$$\hat{p}^n(x_s | y) = \alpha \psi_s(x_s, y_s) \prod_{t \in \Gamma(s)} m_{ts}^n(x_s) \quad (3)$$

For tree-structured graphs, the approximate marginals, or beliefs, $\hat{p}^n(x_s | y)$ will converge to the true marginals $p(x_s | y)$ once the messages from each node have propagated to every other node in the graph. Conceptually, the steady state BP message $m_{ts}(x_s)$ is a sufficient statistic of the observations in the subgraph which node t separates from node s .

Because each iteration of the BP algorithm involves only local message updates, it can be applied even to graphs with cycles. For such graphs, the statistical dependencies between BP messages are not properly accounted for, and the sequence of beliefs $\hat{p}^n(x_s | y)$ will *not* converge to the true marginal distributions. In many applications, however, the resulting loopy BP algorithm exhibits excellent empirical performance [2, 5]. Recently, several theoretical studies have provided insight into the approximations made by loopy BP, partially justifying its application to graphs with cycles [1, 8, 9].

2.2 Nonparametric Representations

Exact evaluation of the BP update equation (2) involves an integration which, as discussed in the Introduction, is not analytically tractable for most continuous hidden variables. An interesting alternative is to represent the resulting message $m_{ts}(x_s)$ nonparametrically as a kernel-based density estimate [13, 14]. Let $\mathcal{N}(x; \mu, \Lambda)$ denote the value of a Gaussian density of mean μ and covariance Λ at the point x . We may then approximate $m_{ts}(x_s)$ by a mixture of M Gaussian kernels as

$$m_{ts}(x_s) = \sum_{i=1}^M w_s^i \mathcal{N}(x_s; x_s^i, \Lambda_s) \quad (4)$$

where w_s^i is the weight associated with the i^{th} kernel mean x_s^i , and Λ_s is a bandwidth or smoothing parameter. Other choices of kernel functions are possible [14], but in this paper we restrict our attention to mixtures of diagonal-covariance Gaussians.

In the following section, we describe stochastic methods for determining the kernel centers x_s^i and associated weights w_s^i . The resulting nonparametric representations are only meaningful when the messages $m_{ts}(x_s)$ are finitely integrable.¹ To guarantee this, it is sufficient to assume that all potentials satisfy the following constraints:

$$\int_{x_s} \psi_{s,t}(x_s, x_t = \bar{x}) dx_s < \infty \quad \int_{x_s} \psi_s(x_s, y_s = \bar{y}) dx_s < \infty \quad (5)$$

Under these assumptions, a simple induction argument will show that all messages are normalizable. Heuristically, equation (5) requires all potentials to be “informative,” so that fixing the value of one variable constrains the likely locations of the other. In most application domains, this can be trivially achieved by assuming that all hidden variables take values in a large, but bounded, range.

3 Nonparametric Message Updates

Conceptually, the BP update equation (2) naturally decomposes into two stages. First, the message product $\psi_t(x_t, y_t) \prod_u m_{ut}^{n-1}(x_t)$ combines information from neighboring nodes with the local evidence y_t to produce a likelihood function for x_t . Second, this likelihood function is combined with the compatibility potential $\psi_{s,t}(x_s, x_t)$, and then integrated to produce likelihoods for x_s . The nonparametric belief propagation (NBP) algorithm stochastically approximates these two stages, producing consistent nonparametric representations of the messages $m_{ts}(x_s)$. Approximate marginals $\hat{p}(x_s | y)$ may then be determined from these messages by applying the following section’s stochastic product algorithm to equation (3).

3.1 Message Products

For the moment, assume that the local observation potentials $\psi_t(x_t, y_t)$ are represented by weighted Gaussian mixtures (such potentials arise naturally from learning-based approaches to model identification [2]). The product of d Gaussian densities is itself Gaussian, with mean and covariance given by

$$\prod_{i=1}^d \mathcal{N}(x; \mu_i, \Lambda_i) \propto \mathcal{N}(x; \bar{\mu}, \bar{\Lambda}) \quad \bar{\Lambda}^{-1} = \sum_{i=1}^d \Lambda_i^{-1} \quad \bar{\Lambda}^{-1} \bar{\mu} = \sum_{i=1}^d \Lambda_i^{-1} \mu_i \quad (6)$$

Thus, a BP update operation which multiplies d Gaussian mixtures, each containing M components, will produce another Gaussian mixture with M^d components. The weight $\bar{w}$ associated with product mixture component $\mathcal{N}(x; \bar{\mu}, \bar{\Lambda})$ is given by

$$\bar{w} \propto \frac{\prod_{i=1}^d w_i \mathcal{N}(x; \mu_i, \Lambda_i)}{\mathcal{N}(x; \bar{\mu}, \bar{\Lambda})} \quad (7)$$

where $\{w_i\}_{i=1}^d$ are the weights associated with the input Gaussians. Since integration of Gaussian mixtures is straightforward, in principle the BP message updates could be performed exactly by repeated use of equations (6,7). In practice, however, the exponential growth of the number of mixture components forces approximations to be made. Given d input mixtures of M Gaussian, the NBP algorithm approximates their M^d -component product mixture by drawing M independent samples.

Direct sampling from this product, achieved by explicitly calculating each of the product component weights (7), would require $\mathcal{O}(M^d)$ operations. The complexity

¹Probabilistically, BP messages are likelihood functions $m_{ts}(x_s) \propto p(y = \bar{y} | x_s)$, not densities, and are *not* necessarily integrable (e.g., when x_s and y are independent).

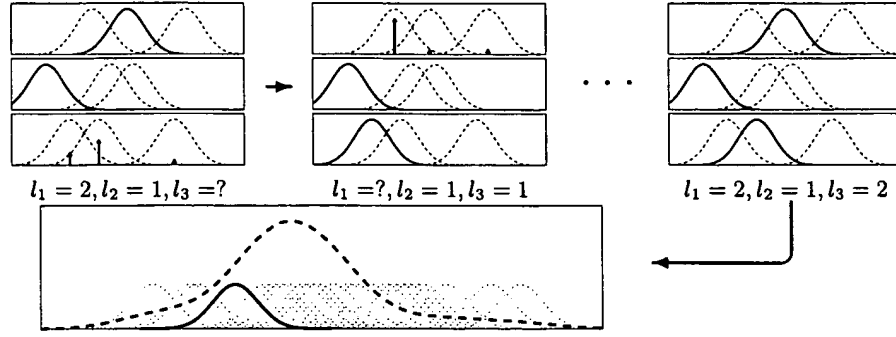


Figure 1: Gibbs sampler for a product of 3 Gaussian mixtures, with 3 kernels each. New indices are sampled according to weights (arrows) determined by the two fixed components (solid). The final labels identify one of the 3^3 components of the product density (dashed).

associated with this sampling is combinatorial: each product component is defined by d labels $\{l_i\}_{i=1}^d$, where l_i identifies a kernel in the i^{th} input mixture. Although the joint distribution of the d labels is complex, the conditional distribution of any individual label l_j is simple. In particular, assuming fixed values for $\{l_i\}_{i \neq j}$, equation (7) can be used to sample from the conditional distribution of l_j in $\mathcal{O}(M)$ operations.

Since the mixture label conditional distributions are tractable, we may use a Gibbs sampler [3] to draw asymptotically unbiased samples from the product distribution. At each iteration, the labels $\{l_i\}_{i \neq j}$ for $d - 1$ of the input mixtures are fixed, and a new value for the j^{th} label is chosen according to equation (7). At the following iteration, the newly chosen l_j is fixed, and another label is updated (see Figure 1). Following the final iteration, the mean and covariance of the selected mixture component is found using equation (6), and a sample point is drawn.

Assuming the number of iterations for the Gibbs sampler to equilibrate is independent of M , we may draw M samples from the product mixture in $\mathcal{O}(dM^2)$ operations. Although formal verification of the Gibbs sampler's convergence is difficult, in our experiments we have observed good performance using far fewer computations than required by direct sampling. Note that in the NBP algorithm, the Gibbs sampler only involves a few hidden variables, in contrast to the very large state spaces of classic simulated annealing [3].

In some applications, the observation potentials $\psi_t(x_t, y_t)$ are most naturally specified by analytic functions. The previously proposed Gibbs sampler may be easily adapted to this case using *importance sampling* [12]. At each iteration, the weights produced by equation (7) are rescaled by $\psi_t(\bar{\mu}, y_t)$, the observation likelihood at that kernel's center. Then, the final sample x_s^i is assigned weight $w_s^i = \psi_t(x_s^i, y_t) / \psi_t(\bar{\mu}, y_t)$ to account for variations of the analytic potential over the kernel's support. This procedure will be most effective when $\psi_t(x_t, y_t)$ varies slowly relative to the typical kernel bandwidth.

3.2 Message Propagation

In the second stage of the NBP algorithm, the information contained in the incoming message product is propagated by stochastically approximating the belief update integral (2). To perform this stochastic integration, the pairwise potential $\psi_{s,t}(x_s, x_t)$ must be decomposed to separate its *marginal* influence on x_t from the

conditional relationship it defines between x_s and x_t .

The marginal influence function $\zeta(x_t)$ is determined by the relative weight assigned to all x_s values for each x_t :

$$\zeta(x_t) = \int_{x_s} \psi_{s,t}(x_s, x_t) dx_s \quad (8)$$

The NBP algorithm accounts for the marginal influence of $\psi_{s,t}(x_s, x_t)$ by incorporating $\zeta(x_t)$ into the Gibbs sampler. If $\psi_{s,t}(x_s, x_t)$ is a Gaussian mixture, extraction of $\zeta(x_t)$ is trivial. Alternately, if $\zeta(x_t)$ can be evaluated (or approximated) pointwise, analytic pairwise potentials may be dealt with using importance sampling. In the common case where pairwise potentials depend only on the difference between their arguments ($\psi_{s,t}(x, \bar{x}) = \psi_{s,t}(x - \bar{x})$), $\zeta(x_t)$ is constant and can be neglected.

To complete the stochastic integration, each particle x_t^j produced by the Gibbs sampler is propagated to node s by sampling $x_s^j \sim \alpha \psi_{s,t}(x_s, x_t^j)$. Note that the assumptions of section 2.2 ensure that $\psi_{s,t}(x_s, x_t^j)$ is normalizable for any x_t^j . The method by which this sampling step is performed will depend on the specific functional form of $\psi_{s,t}(x_s, x_t)$, and may involve importance sampling or MCMC techniques.

Having produced a set of independent samples from the desired output message $m_{ts}(x_s)$, NBP must choose a kernel bandwidth to complete the nonparametric density estimate. There are many ways to make this choice; for the results in this paper, we used leave-one-out likelihood cross-validation [14].

4 Gaussian Graphical Models

Gaussian graphical models provide one of the few continuous distributions for which the BP algorithm may be implemented exactly [9]. For this reason, Gaussian models may be used to test the accuracy of the nonparametric approximations made by NBP. Note that we cannot hope for NBP to outperform algorithms (like Gaussian BP) designed to take advantage of the linear structure underlying Gaussian problems. Instead, our goal is to verify NBP's performance in a situation where exact comparisons are possible.

We have tested the NBP algorithm on Gaussian models with a range of graphical structures, including chains, trees, and grids. Similar results were observed in all cases, so here we only present data for a single typical 5×5 nearest-neighbor grid, with randomly selected inhomogeneous potential functions. For each node $s \in \mathcal{V}$, Gaussian BP converges to a steady-state estimate of the marginal mean μ_s and variance σ_s^2 after about 15 iterations. To evaluate NBP, we performed 15 iterations of the NBP message updates using several different particle set sizes $M \in [10, 400]$. We then found the marginal mean $\hat{\mu}_s$ and variance $\hat{\sigma}_s^2$ estimates implied by the final NBP density estimates. For each tested particle set size, the NBP comparison was repeated 100 times.

Using the data from each NBP trial, we computed the error in the mean and variance estimates, normalized so each node behaved like a unit-variance Gaussian:

$$\tilde{\mu}_s = \frac{\hat{\mu}_s - \mu_s}{\sigma_s} \quad \tilde{\sigma}_s^2 = \frac{\hat{\sigma}_s^2 - \sigma_s^2}{\sqrt{2}\sigma_s^2} \quad (9)$$

Figure 2 shows the mean and variance of these error statistics, across all nodes and trials, for different particle set sizes M . The NBP algorithm always provides unbiased estimates of the conditional mean, but overly large variance estimates. This

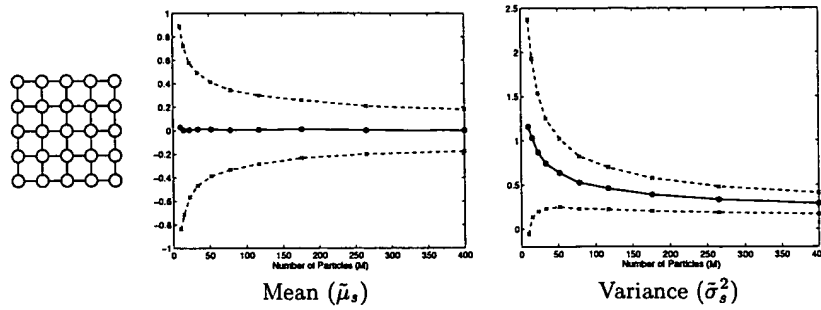


Figure 2: NBP performance for a 5×5 grid with Gaussian potentials and observations. Plots show the mean (solid line) and standard deviation (dashed line) of the normalized error measures of equation (9), as a function of particle set size M .

bias, which decreases as more particles are used, is due to the smoothing inherent in kernel-based density estimates. As expected for samples drawn from Gaussian distributions, the standard deviation of both error measures falls as $M^{-1/2}$.

5 Stereo Vision

Stereo vision algorithms infer point correspondences between horizontally aligned image pairs of the same three-dimensional scene [4, 20]. The distance, or disparity, between matching pixels is inversely proportional to the corresponding object's distance from the camera. Estimation of accurate, dense disparity maps is complicated by textureless regions, pixelization effects, and occlusions. For these reasons, prior knowledge about disparity variations plays a critical role in stereo computation.

The prior assumptions underlying stereo vision algorithms are often represented by grid-structured graphical models in which pairwise potentials correlate neighboring disparity values, while observation potentials measure similarities between local image features [5, 20]. In [4], three types of potential functions are suggested, each encoding different assumptions about the underlying disparity “worlds”:

World I Pairwise potentials assume disparity differences are Gaussian. Similarly, observation potentials assign Gaussian distributions to image feature differences. Note that the resulting disparity likelihoods are *not* Gaussian.

World II The Gaussian potentials of World I are augmented by binary outlier processes to model occlusions and depth discontinuities. We instead use Gaussians contaminated by uniform noise, as suggested by [21]. A similar model has been used with the discrete BP algorithm [5].

World III Observation potentials are as in World II, but the hidden variable at each node is augmented to model horizontal and vertical disparity gradients. Uniform-contaminated Gaussian potentials model differences in adjacent gradients.

Our goal is not to present state-of-the-art stereo results, but to explore the characteristics of the NBP algorithm using the unique features of each world.

5.1 Worlds I & II

Figure 3 compares the NBP algorithm's performance to discretized BP on two 50×50 patches containing interesting disparity features. The patches were extracted from the standard “Venus” and “Sawtooth” stereo test images [20]. Discrete BP

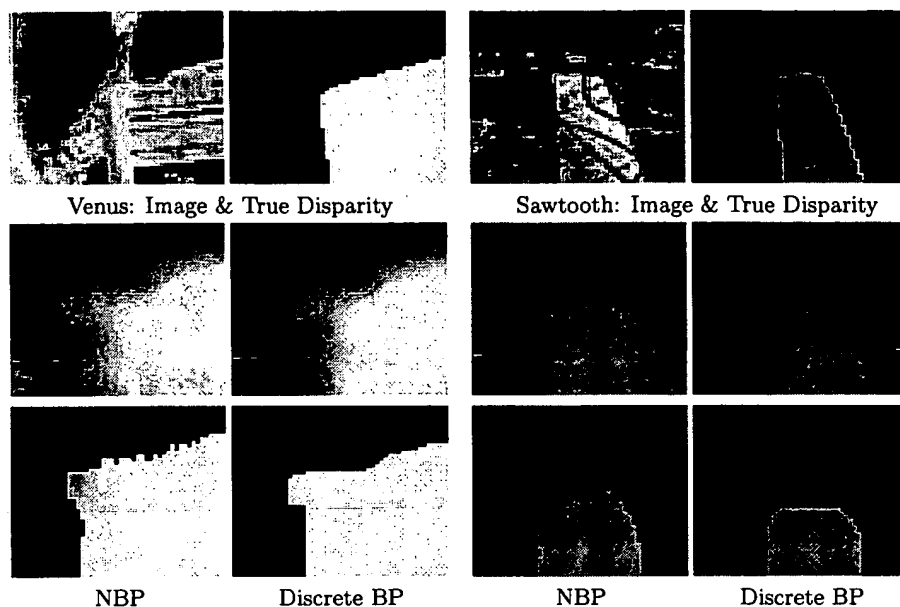


Figure 3: NBP and discrete BP stereo results for the “Venus” and “Sawtooth” patches. Top row: reference image and true disparities. Second row: World I. Third row: World II.

quantized the one-dimensional disparity space to 100 levels, while NBP used 100 particles sampled with 100 MCMC iterations. Because of the limited range of possible disparities, discrete BP’s results approximate the true continuous BP results extremely accurately. We see that NBP accurately captures the qualitative features of both worlds, matching the discrete BP results extremely well.

5.2 World III

All of the objects in the Venus and Sawtooth images have slopes which are nearly parallel to the image plane. For this reason, the gradient information used in World III’s sophisticated prior does not significantly modify the World II results. We have constructed a synthetic stereo problem, shown in Figure 4, consisting of a single constant slope scan line. Both ends of the scan line contain good observations, but the central section contains no disparity information, as would be caused by a textureless image region. Figures 4(b,c) show that World II’s bias towards horizontal surfaces is inappropriate for this data set. Figure 4(d) shows performance attained by discrete BP with slopes chosen to exactly match the true surface. However, in practical situations it is not possible to discretize the three-dimensional World III state space finely enough to achieve this. Figure 4(e) shows the heavy quantization artifacts produced by discrete BP using 1000 ($40 \times 5 \times 5$) points. In contrast, using only 100 points, NBP produces the high-resolution density estimates of Figure 4(f).

6 Discussion

We have developed a nonparametric sampling-based variant of the belief propagation algorithm for graphical models with continuous, non-Gaussian random variables. Our stereo vision results suggest that NBP achieves performance comparable

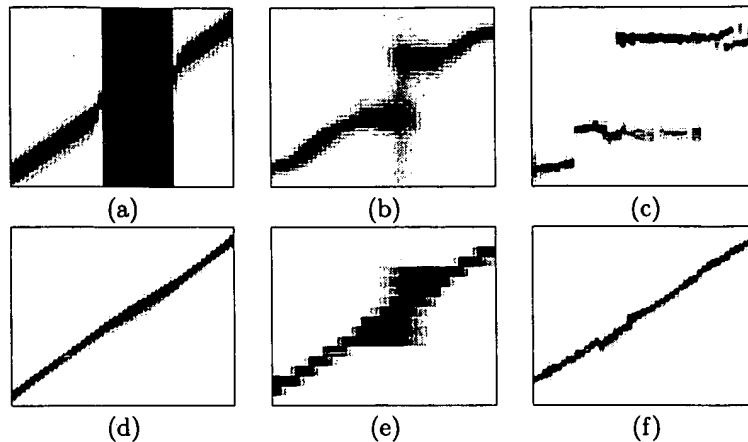


Figure 4: Stereo results for a synthetic scan line. Each plot is a columnwise representation of the nodes' marginal distributions. (a) Observation likelihoods. (b) World II discrete BP. (c) World II NBP. (d) World III ideal. (e) World III discrete BP. (f) World III NBP.

to discretization for low-dimensional hidden variables, and may offer significant advantages in higher dimensional spaces. In the future, the application of more sophisticated Monte Carlo and density estimation techniques could improve the statistical accuracy and computational efficiency of the message updates. We hope that the nonparametric approach will allow richer, more realistic models to be designed and used in applications in which continuous variables naturally arise.

Acknowledgments

This research was supported in part by ONR under Grant N00014-00-1-0089, by AFOSR under Grant F49620-00-1-0362, and by an ODDR&E MURI funded through ARO Grant DAAD19-00-1-0466. E.S. supported by an NDSEG fellowship.

References

- [1] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *IJCAI*, August 2001.
- [2] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.
- [3] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, November 1984.
- [4] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *IJCV*, 19(3):237–262, 1996.
- [5] J. Sun, H. Shum, and N. Zheng. Stereo matching using belief propagation. In *ECCV*, pages 510–524, 2002.
- [6] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.
- [7] G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
- [8] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization for approximate inference on loopy graphs. In *NIPS 14*. MIT Press, 2002.

- [9] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Comp.*, 13:2173–2200, 2001.
- [10] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
- [11] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. AC*, 17(4):439–448, August 1972.
- [12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [13] E. Parzen. On estimation of a probability density function and mode. *Ann. of Math Stats.*, 33:1065–1076, 1962.
- [14] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.
- [15] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.
- [16] A. P. Dawid, U. Kjærulff, and S. L. Lauritzen. Hybrid propagation in junction trees. In *Advances in Intelligent Computing*, pages 87–97, 1995.
- [17] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *UAI 15*, pages 324–333, 1999.
- [18] U. Kjærulff. HUGS: Combining exact inference and Gibbs sampling in junction trees. In *UAI 11*, pages 368–375, 1995.
- [19] L. D. Hernández and S. Moral. Mixing exact and importance sampling propagation algorithms in dependence graphs. *Int. J. Intelligent Systems*, 12:553–576, 1997.
- [20] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002.
- [21] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola

viola@merl.com

Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones

mjjones@crl.dec.com

Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers[6]. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

1. Introduction

This paper brings together new algorithms and insights to construct a framework for robust and extremely rapid object detection. This framework is demonstrated on, and in part motivated by, the task of face detection. Toward this end we have constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to the best published results [16, 12, 15, 11, 1]. This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly. Operating on 384 by 288 pixel images, faces are de-

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates.

There are three main contributions of our object detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections.

The first contribution of this paper is a new image representation called an *integral image* that allows for very fast feature evaluation. Motivated in part by the work of Papa-georgiou et al. our detection system does not work directly with image intensities [10]. Like these authors we use a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales we introduce the integral image representation for images. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in *constant* time.

The second contribution of this paper is a method for constructing a classifier by selecting a small number of important features using AdaBoost [6]. Within any image sub-window the total number of Harr-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. Motivated by the work of Tieu and Viola, feature selection is achieved through a simple modification of the AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a

single feature [2]. As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. AdaBoost provides an effective learning algorithm and strong bounds on generalization performance [13, 9, 10].

The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image. The notion behind focus of attention approaches is that it is often possible to rapidly determine where in an image an object might occur [17, 8, 1]. More complex processing is reserved only for these promising regions. The key measure of such an approach is the “false negative” rate of the attentional process. It must be the case that all, or almost all, object instances are selected by the attentional filter.

We will describe a process for training an extremely simple and efficient classifier which can be used as a “supervised” focus of attention operator. The term supervised refers to the fact that the attentional operator is trained to detect examples of a particular class. In the domain of face detection it is possible to achieve fewer than 1% false negatives and 40% false positives using a classifier constructed from two Harr-like features. The effect of this filter is to reduce by over one half the number of locations where the final detector must be evaluated.

Those sub-windows which are not rejected by the initial classifier are processed by a sequence of classifiers, each slightly more complex than the last. If any classifier rejects the sub-window, no further processing is performed. The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of Geman and colleagues [1, 4].

An extremely fast face detector will have broad practical applications. These include user interfaces, image databases, and teleconferencing. In applications where rapid frame-rates are not necessary, our system will allow for significant additional post-processing and analysis. In addition our system can be implemented on a wide range of small low power devices, including hand-helds and embedded processors. In our lab we have implemented this face detector on the Compaq iPaq handheld and have achieved detection at two frames per second (this device has a low power 200 mips *Strong Arm* processor which lacks floating point hardware).

The remainder of the paper describes our contributions and a number of experimental results, including a detailed description of our experimental methodology. Discussion of closely related work takes place at the end of each section.

2. Features

Our object detection procedure classifies images based on the value of simple features. There are many motivations

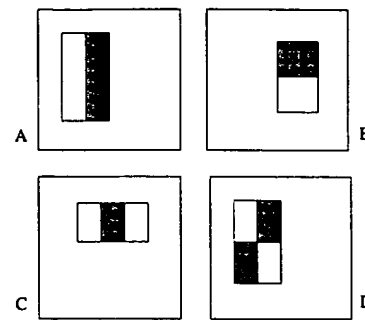


Figure 1: Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

for using features rather than the pixels directly. The most common reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. For this system there is also a second critical motivation for features: the feature based system operates much faster than a pixel-based system.

The simple features used are reminiscent of Haar basis functions which have been used by Papageorgiou et al. [10]. More specifically, we use three kinds of features. The value of a *two-rectangle feature* is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent (see Figure 1). A *three-rectangle feature* computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a *four-rectangle feature* computes the difference between diagonal pairs of rectangles.

Given that the base resolution of the detector is 24x24, the exhaustive set of rectangle features is quite large, over 180,000. Note that unlike the Haar basis, the set of rectangle features is overcomplete¹.

2.1. Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image.² The integral image at location x, y contains the sum of the pixels above and to the left of x, y , inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

¹A complete basis has no linear dependence between basis elements and has the same number of elements as the image space, in this case 576. The full set of 180,000 thousand features is many times over-complete.

²There is a close relation to “summed area tables” as used in graphics [3]. We choose a different name here in order to emphasize its use for the analysis of images, rather than for texture mapping.

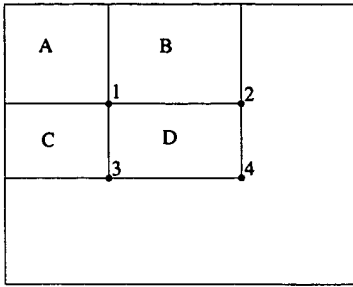


Figure 2: The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A . The value at location 2 is $A + B$, at location 3 is $A + C$, and at location 4 is $A + B + C + D$. The sum within D can be computed as $4 + 1 - (2 + 3)$.

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

(where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$) the integral image can be computed in one pass over the original image.

Using the integral image any rectangular sum can be computed in four array references (see Figure 2). Clearly the difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

2.2. Feature Discussion

Rectangle features are somewhat primitive when compared with alternatives such as steerable filters [5, 7]. Steerable filters, and their relatives, are excellent for the detailed analysis of boundaries, image compression, and texture analysis. In contrast rectangle features, while sensitive to the presence of edges, bars, and other simple image structure, are quite coarse. Unlike steerable filters the only orientations available are vertical, horizontal, and diagonal. The set of rectangle features do however provide a rich image representation which supports effective learning. In conjunction with the integral image, the efficiency of the rectangle feature set provides ample compensation for their limited flexibility.

3. Learning Classification Functions

Given a feature set and a training set of positive and negative images, any number of machine learning approaches

could be used to learn a classification function. In our system a variant of AdaBoost is used *both* to select a small set of features *and* train the classifier [6]. In its original form, the AdaBoost learning algorithm is used to boost the classification performance of a simple (sometimes called weak) learning algorithm. There are a number of formal guarantees provided by the AdaBoost learning procedure. Freund and Schapire proved that the training error of the strong classifier approaches zero exponentially in the number of rounds. More importantly a number of results were later proved about generalization performance [14]. The key insight is that generalization performance is related to the margin of the examples, and that AdaBoost achieves large margins rapidly.

Recall that there are over 180,000 rectangle features associated with each image sub-window, a number far larger than the number of pixels. Even though each feature can be computed very efficiently, computing the complete set is prohibitively expensive. Our hypothesis, which is borne out by experiment, is that a very small number of these features can be combined to form an effective classifier. The main challenge is to find these features.

In support of this goal, the weak learning algorithm is designed to select the single rectangle feature which best separates the positive and negative examples (this is similar to the approach of [2] in the domain of image database retrieval). For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier $h_j(x)$ thus consists of a feature f_j , a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Here x is a 24×24 pixel sub-window of an image. See Table 1 for a summary of the boosting process.

In practice no single feature can perform the classification task with low error. Features which are selected in early rounds of the boosting process had error rates between 0.1 and 0.3. Features selected in later rounds, as the task becomes more difficult, yield error rates between 0.4 and 0.5.

3.1. Learning Discussion

Many general feature selection procedures have been proposed (see chapter 8 of [18] for a review). Our final application demanded a very aggressive approach which would discard the vast majority of features. For a similar recognition problem Papageorgiou et al. proposed a scheme for feature selection based on feature variance [10]. They demonstrated good results selecting 37 features out of a total 1734 features.

Roth et al. propose a feature selection process based on the Winnow exponential perceptron learning rule [11]. The Winnow learning process converges to a solution where many of these weights are zero. Nevertheless a very large

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Choose the classifier, h_t , with the lowest error ϵ_t .
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Table 1: The AdaBoost algorithm for classifier learning. Each round of boosting selects one feature from the 180,000 potential features.

number of features are retained (perhaps a few hundred or thousand).

3.2. Learning Results

While details on the training and performance of the final system are presented in Section 5, several simple results merit discussion. Initial experiments demonstrated that a frontal face classifier constructed from 200 features yields a detection rate of 95% with a false positive rate of 1 in 14084. These results are compelling, but not sufficient for many real-world tasks. In terms of computation, this classifier is probably faster than any other published system, requiring 0.7 seconds to scan an 384 by 288 pixel image. Unfortunately, the most straightforward technique for improving detection performance, adding features to the classifier, directly increases computation time.

For the task of face detection, the initial rectangle features selected by AdaBoost are meaningful and easily interpreted. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region

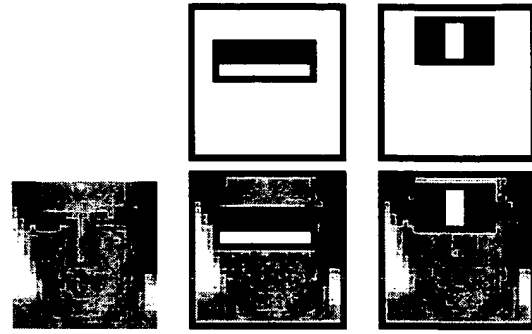


Figure 3: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

of the nose and cheeks (see Figure 3). This feature is relatively large in comparison with the detection sub-window, and should be somewhat insensitive to size and location of the face. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.

4. The Attentional Cascade

This section describes an algorithm for constructing a cascade of classifiers which achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances (i.e. the threshold of a boosted classifier can be adjusted so that the false negative rate is close to zero). Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates.

The overall form of the detection process is that of a degenerate decision tree, what we call a “cascade” (see Figure 4). A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window.

Stages in the cascade are constructed by training classifiers using AdaBoost and then adjusting the threshold to minimize false negatives. Note that the default AdaBoost threshold is designed to yield a low error rate on the training data. In general a lower threshold yields higher detec-

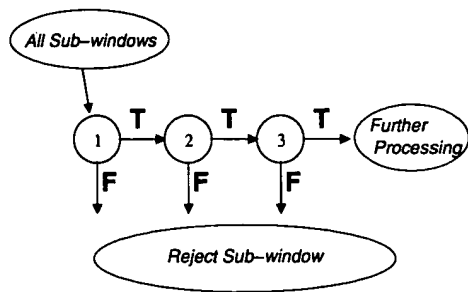


Figure 4: Schematic depiction of a the detection cascade. A series of classifiers are applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the number of sub-windows have been reduced radically. Further processing can take any form such as additional stages of the cascade (as in our detection system) or an alternative detection system.

tion rates and higher false positive rates.

For example an excellent first stage classifier can be constructed from a two-feature strong classifier by reducing the threshold to minimize false negatives. Measured against a validation training set, the threshold can be adjusted to detect 100% of the faces with a false positive rate of 40%. See Figure 3 for a description of the two features used in this classifier.

Computation of the two feature classifier amounts to about 60 microprocessor instructions. It seems hard to imagine that any simpler filter could achieve higher rejection rates. By comparison, scanning a simple image template, or a single layer perceptron, would require at least 20 times as many operations per sub-window.

The structure of the cascade reflects the fact that within any single image an overwhelming majority of sub-windows are negative. As such, the cascade attempts to reject as many negatives as possible at the earliest stage possible. While a positive instance will trigger the evaluation of every classifier in the cascade, this is an exceedingly rare event.

Much like a decision tree, subsequent classifiers are trained using those examples which pass through all the previous stages. As a result, the second classifier faces a more difficult task than the first. The examples which make it through the first stage are "harder" than typical examples. The more difficult examples faced by deeper classifiers push the entire receiver operating characteristic (ROC) curve downward. At a given detection rate, deeper classifiers have correspondingly higher false positive rates.

4.1. Training a Cascade of Classifiers

The cascade training process involves two types of trade-offs. In most cases classifiers with more features will achieve higher detection rates and lower false positive rates. At the same time classifiers with more features require more time to compute. In principle one could define an optimization framework in which: i) the number of classifier stages, ii) the number of features in each stage, and iii) the threshold of each stage, are traded off in order to minimize the expected number of evaluated features. Unfortunately finding this optimum is a tremendously difficult problem.

In practice a very simple framework is used to produce an effective classifier which is highly efficient. Each stage in the cascade reduces the false positive rate and decreases the detection rate. A target is selected for the minimum reduction in false positives and the maximum decrease in detection. Each stage is trained by adding features until the target detection and false positives rates are met (these rates are determined by testing the detector on a validation set). Stages are added until the overall target for false positive and detection rate is met.

4.2. Detector Cascade Discussion

The complete face detection cascade has 38 stages with over 6000 features. Nevertheless the cascade structure results in fast average detection times. On a difficult dataset, containing 507 faces and 75 million sub-windows, faces are detected using an average of 10 feature evaluations per sub-window. In comparison, this system is about 15 times faster than an implementation of the detection system constructed by Rowley et al.³ [12]

A notion similar to the cascade appears in the face detection system described by Rowley et al. in which two detection networks are used [12]. Rowley et al. used a faster yet less accurate network to prescreen the image in order to find candidate regions for a slower more accurate network. Though it is difficult to determine exactly, it appears that Rowley et al.'s two network face system is the fastest existing face detector.⁴

The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of Amit and Geman [1]. Unlike techniques which use a fixed detector, Amit and Geman propose an alternative point of view where unusual co-occurrences of simple image features are used to trigger the evaluation of a more complex detection process. In this way the full detection process need not be evaluated at many of the potential image locations and scales. While this basic insight

³Henry Rowley very graciously supplied us with implementations of his detection system for direct comparison. Reported results are against his fastest system. It is difficult to determine from the published literature, but the Rowley-Baluja-Kanade detector is widely considered the fastest detection system and has been heavily tested on real-world problems.

⁴Other published detectors have either neglected to discuss performance in detail, or have never published detection and false positive rates on a large and difficult training set.

is very valuable, in their implementation it is necessary to first evaluate some feature detector at every location. These features are then grouped to find unusual co-occurrences. In practice, since the form of our detector and the features that it uses are extremely efficient, the amortized cost of evaluating our detector at *every scale and location* is much faster than finding and grouping edges throughout the image.

In recent work Fleuret and Geman have presented a face detection technique which relies on a “chain” of tests in order to signify the presence of a face at a particular scale and location [4]. The image properties measured by Fleuret and Geman, disjunctions of fine scale edges, are quite different than rectangle features which are simple, exist at all scales, and are somewhat interpretable. The two approaches also differ radically in their learning philosophy. The motivation for Fleuret and Geman’s learning process is density estimation and density discrimination, while our detector is purely discriminative. Finally the false positive rate of Fleuret and Geman’s approach appears to be higher than that of previous approaches like Rowley et al. and this approach. Unfortunately the paper does not report quantitative results of this kind. The included example images each have between 2 and 10 false positives.

5 Results

A 38 layer cascaded classifier was trained to detect frontal upright faces. To train the detector, a set of face and non-face training images were used. The face training set consisted of 4916 hand labeled faces scaled and aligned to a base resolution of 24 by 24 pixels. The faces were extracted from images downloaded during a random crawl of the world wide web. Some typical face examples are shown in Figure 5. The non-face subwindows used to train the detector come from 9544 images which were manually inspected and found to not contain any faces. There are about 350 million subwindows within these non-face images.

The number of features in the first five layers of the detector is 1, 10, 25, 25 and 50 features respectively. The remaining layers have increasingly more features. The total number of features in all layers is 6061.

Each classifier in the cascade was trained with the 4916 training faces (plus their vertical mirror images for a total of 9832 training faces) and 10,000 non-face sub-windows (also of size 24 by 24 pixels) using the Adaboost training procedure. For the initial one feature classifier, the non-face training examples were collected by selecting random sub-windows from a set of 9544 images which did not contain faces. The non-face examples used to train subsequent layers were obtained by scanning the partial cascade across the non-face images and collecting false positives. A maximum of 10000 such non-face sub-windows were collected for each layer.

Speed of the Final Detector



Figure 5: Example of frontal upright face images used for training.

The speed of the cascaded detector is directly related to the number of features evaluated per scanned sub-window. Evaluated on the MIT+CMU test set [12], an average of 10 features out of a total of 6061 are evaluated per sub-window. This is possible because a large majority of sub-windows are rejected by the first or second layer in the cascade. On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds (using a starting scale of 1.25 and a step size of 1.5 described below). This is roughly 15 times faster than the Rowley-Baluja-Kanade detector [12] and about 600 times faster than the Schneiderman-Kanade detector [15].

Image Processing

All example sub-windows used for training were variance normalized to minimize the effect of different lighting conditions. Normalization is therefore necessary during detection as well. The variance of an image sub-window can be computed quickly using a pair of integral images. Recall that $\sigma^2 = m^2 - \frac{1}{N} \sum x^2$, where σ is the standard deviation, m is the mean, and x is the pixel value within the sub-window. The mean of a sub-window can be computed using the integral image. The sum of squared pixels is computed using an integral image of the image squared (i.e. two integral images are used in the scanning process). During scanning the effect of image normalization can be achieved by post-multiplying the feature values rather than pre-multiplying the pixels.

Scanning the Detector

The final detector is scanned across the image at multiple scales and locations. Scaling is achieved by scaling the detector itself, rather than scaling the image. This process makes sense because the features can be evaluated at any

False detections Detector	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

Table 2: Detection rates for various numbers of false positives on the MIT+CMU test set containing 130 images and 507 faces.

scale with the same cost. Good results were obtained using a set of scales a factor of 1.25 apart.

The detector is also scanned across location. Subsequent locations are obtained by shifting the window some number of pixels Δ . This shifting process is affected by the scale of the detector: if the current scale is s the window is shifted by $\lceil s\Delta \rceil$, where $\lceil \cdot \rceil$ is the rounding operation.

The choice of Δ affects both the speed of the detector as well as accuracy. The results we present are for $\Delta = 1.0$. We can achieve a significant speedup by setting $\Delta = 1.5$ with only a slight decrease in accuracy.

Integration of Multiple Detections

Since the final detector is insensitive to small changes in translation and scale, multiple detections will usually occur around each face in a scanned image. The same is often true of some types of false positives. In practice it often makes sense to return one final detection per face. Toward this end it is useful to postprocess the detected sub-windows in order to combine overlapping detections into a single detection.

In these experiments detections are combined in a very simple fashion. The set of detections are first partitioned into disjoint subsets. Two detections are in the same subset if their bounding regions overlap. Each partition yields a single final detection. The corners of the final bounding region are the average of the corners of all detections in the set.

Experiments on a Real-World Test Set

We tested our system on the MIT+CMU frontal face test set [12]. This set consists of 130 images with 507 labeled frontal faces. A ROC curve showing the performance of our detector on this test set is shown in Figure 6. To create the ROC curve the threshold of the final layer classifier is adjusted from $-\infty$ to $+\infty$. Adjusting the threshold to $+\infty$ will yield a detection rate of 0.0 and a false positive rate of 0.0. Adjusting the threshold to $-\infty$, however, increases both the detection rate and false positive rate, but only to a certain point. Neither rate can be higher than the rate of the detection cascade minus the final layer. In effect, a threshold of $-\infty$ is equivalent to removing that layer. Further increasing the detection and false positive rates requires decreasing the threshold of the next classifier in the cascade.

Thus, in order to construct a complete ROC curve, classifier layers are removed. We use the *number* of false positives as opposed to the *rate* of false positives for the x-axis of the ROC curve to facilitate comparison with other systems. To compute the false positive rate, simply divide by the total number of sub-windows scanned. In our experiments, the number of sub-windows scanned is 75,081,800.

Unfortunately, most previous published results on face detection have only included a single operating regime (i.e. single point on the ROC curve). To make comparison with our detector easier we have listed our detection rate for the false positive rates reported by the other systems. Table 2 lists the detection rate for various numbers of false detections for our system as well as other published systems. For the Rowley-Baluja-Kanade results [12], a number of different versions of their detector were tested yielding a number of different results they are all listed in under the same heading. For the Roth-Yang-Ahuja detector [11], they reported their result on the MIT+CMU test set minus 5 images containing line drawn faces removed.

Figure 7 shows the output of our face detector on some test images from the MIT+CMU test set.

A simple voting scheme to further improve results

In table 2 we also show results from running three detectors (the 38 layer one described above plus two similarly trained detectors) and outputting the majority vote of the three detectors. This improves the detection rate as well as eliminating more false positives. The improvement would be greater if the detectors were more independent. The correlation of their errors results in a modest improvement over the best single detector.

6 Conclusions

We have presented an approach for object detection which minimizes computation time while achieving high detection accuracy. The approach was used to construct a face detection system which is approximately 15 faster than any previous approach.

This paper brings together new algorithms, representations, and insights which are quite generic and may well

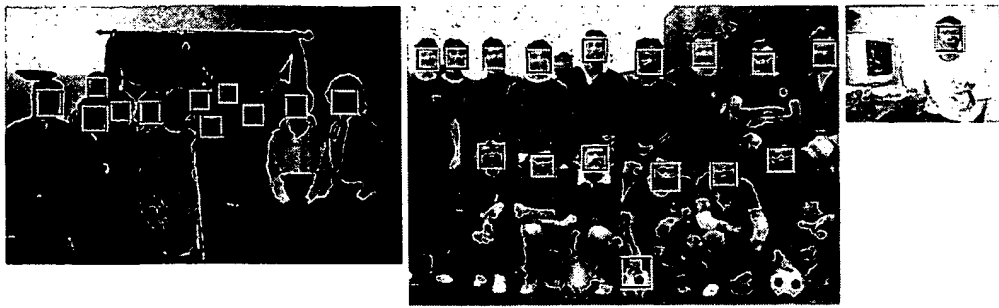


Figure 7: Output of our face detector on a number of test images from the MIT+CMU test set.

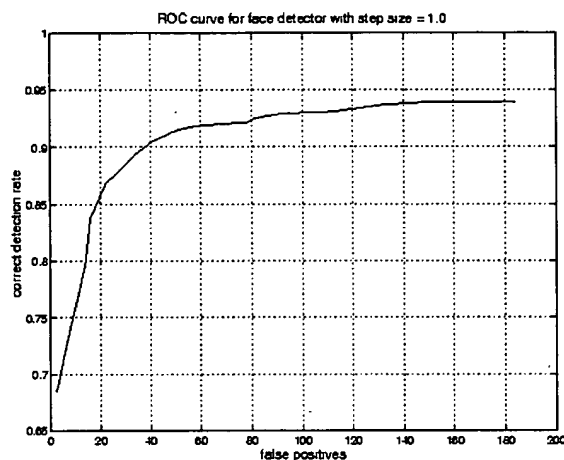


Figure 6: ROC curve for our face detector on the MIT+CMU test set. The detector was run using a step size of 1.0 and starting scale of 1.0 (75,081,800 sub-windows scanned).

have broader application in computer vision and image processing.

Finally this paper presents a set of detailed experiments on a difficult face detection dataset which has been widely studied. This dataset includes faces under a very wide range of conditions including: illumination, scale, pose, and camera variation. Experiments on such a large and complex dataset are difficult and time consuming. Nevertheless systems which work under these conditions are unlikely to be brittle or limited to a single set of conditions. More importantly conclusions drawn from this dataset are unlikely to be experimental artifacts.

References

- [1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers, 1997.
- [2] Anonymous. Anonymous. In *Anonymous*, 2000.
- [3] F. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, volume 18(3), pages 207–212, 1984.
- [4] F. Fleuret and D. Geman. Coarse-to-fine face detection. *Int. J. Computer Vision*, 2001.
- [5] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.
- [7] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C. Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994.
- [8] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Patt. Anal. Mach. Intell.*, 20(11):1254–1259, November 1998.
- [9] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
- [10] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, 1998.
- [11] D. Roth, M. Yang, and N. Ahuja. A snowbased face detector. In *Neural Information Processing 12*, 2000.
- [12] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 22–38, 1998.
- [13] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Stat.*, 26(5):1651–1686, 1998.
- [14] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.
- [15] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *International Conference on Computer Vision*, 2000.

- [16] K. Sung and T. Poggio. Example-based learning for view-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 39–51, 1998.
- [17] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y.H. Lai, N. Davis, and F. Nuflo. Modeling visual-attention via selective tuning. *Artificial Intelligence Journal*, 78(1-2):507–545, October 1995.
- [18] Andrew Webb. *Statistical Pattern Recognition*. Oxford University Press, New York, 1999.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

PAMPAS: Real-Valued Graphical Models for Computer Vision

M. Isard
 Microsoft Research
 Mountain View, CA 94043

Abstract

Probabilistic models have been adopted for many computer vision applications, however inference in high-dimensional spaces remains problematic. As the state-space of a model grows, the dependencies between the dimensions lead to an exponential growth in computation when performing inference. Many common computer vision problems naturally map onto the graphical model framework; the representation is a graph where each node contains a portion of the state-space and there is an edge between two nodes only if they are not independent conditional on the other nodes in the graph. When this graph is sparsely connected, belief propagation algorithms can turn an exponential inference computation into one which is linear in the size of the graph. However belief propagation is only applicable when the variables in the nodes are discrete-valued or jointly represented by a single multivariate Gaussian distribution, and this rules out many computer vision applications.

This paper combines belief propagation with ideas from particle filtering; the resulting algorithm performs inference on graphs containing both cycles and continuous-valued latent variables with general conditional probability distributions. Such graphical models have wide applicability in the computer vision domain and we test the algorithm on example problems of low-level edge linking and locating jointed structures in clutter.

1. Introduction

Probabilistic techniques for modelling uncertainty have found widespread success in computer vision. Their application ranges from pixel-level image generation models [13, 7] to tracking in high-level parametric spaces [11, 22] however as the size of the state-space grows the “curse of dimensionality” can cause an exponential growth in the computations necessary to perform inference. Many probabilistic vision models naturally decompose into a *graphical model* structure [14]; basic components become nodes in a graph where each node is conditionally independent of all

but its near neighbours. When the nodes are image patches the neighbours may be spatially close in an image, adjacent levels in a multi-scale representation or nearby time instants in an image sequence. Alternatively many complex objects can be decomposed into graphs where the nodes are subparts of the object and an edge indicates that two subparts are physically connected. The advantage of this representation is that some formerly exponential inference computations become linear in the size of the graph.

Exact inference on graphical models is possible only in restricted circumstances [14]; the (directed) graph must be acyclic and the joint distribution over those latent variables which are not discrete must be a single multivariate Gaussian. When a model violates these restrictions, Gibbs sampling can be used to draw approximate samples from the joint distribution [9] but for most computer vision applications this technique remains intractable. Approximate inference methods can be used for Conditional Linear Gaussian models [16] including time series [3]. Recently two methods have been widely studied in the computer vision literature to perform approximate inference on more general classes of graph; loopy belief propagation [26] (LBP) can be applied to graphs with cycles, though it still only applies to discrete or jointly Gaussian random variables; and particle filtering [5] allows the use of very general distributions over continuous-valued random variables but applies only to graphs with a simple linear chain structure.

The restriction to Gaussian latent variables is particularly onerous in computer vision applications because clutter in image generation models invariably leads to highly non-Gaussian, multi-modal posterior distributions. This effect largely accounts for the popularity of particle filtering in the computer vision tracking community. In this paper we describe the PAMPAS algorithm (Particle Message PASSing) which combines LBP with ideas from particle filtering. It has wide applicability in the computer vision domain, and increases the range of complex continuous-valued models which admit tractable probabilistic inference. We investigate the behaviour of the algorithm on test problems including illusory-contour finding and locating a jointed structure. Additionally we argue that an algorithm which performs

tractable inference over continuous-valued graphical models is a promising tool for research into unified computer vision models implementing a single probabilistic framework all the way from the pixel level to a semantic description.

2. Belief propagation

Belief propagation (BP) has been used for many years to perform inference in Bayesian networks [14] and has recently been applied to graphs with cycles under the name of loopy belief propagation [26]. The method consists in passing *messages* between the nodes of the graph. When all the latent variables are discrete-valued the messages can be encoded as matrices, and when the continuous-valued variables are jointly Gaussian the messages can be summarised using a mean and covariance matrix.

When using models with more complex continuous-valued marginal distributions the data representation for messages becomes problematic since as they are propagated their complexity grows exponentially [18]. Techniques exist for pruning the representations when the potentials and likelihoods are exponential distributions [16, 19, 2] and while these can be used to perform inference over Gaussian mixture models like those in this paper, they are not well suited to the complex likelihoods which arise in computer vision applications. When the graph is a chain a particle filter [5] can be used which represents the marginals with a non-parametric form, the *particle set*. Particle filtering has been widely used in computer vision largely because it performs well with common image likelihood models.

This section begins with a brief description of standard BP and goes on to describe the PAMPAS algorithm which modifies BP to use particle sets as messages and thus permits approximate inference over general continuous-valued graphical models. Sudderth et al [23, 24] have independently developed an almost identical algorithm and a comparison is presented in section 2.5.

2.1. Discrete Belief Propagation

Belief propagation can in general be stated using a pairwise MRF formulation [26]. Consider a set of latent-variable nodes $\mathcal{X} = \{X_1, \dots, X_M\}$ paired with a set of observed nodes $\mathcal{Y} = \{Y_1, \dots, Y_M\}$. The conditional independence structure of the problem is expressed in the neighbourhood set $\mathcal{P}$. A pair of indices $(i, j) \in \mathcal{P}$ if the hidden node X_j is not conditionally independent of X_i given the other nodes in the graph, and in this case we say X_i is a parent of X_j and by slight abuse of notation $i \in \mathcal{P}(j)$.

For tutorial simplicity, when the X_i are discrete-valued random variables we assume each takes on some value $x_i \in \{1, \dots, L\} \equiv \mathcal{L}$. Given a fixed assignment to $\mathcal{Y}$ the observed and hidden nodes are related by observation

functions $\phi_i(x_i, y_i) \equiv \phi_i(x_i) : \mathcal{L} \rightarrow \mathbb{R}$. Adjacent hidden nodes are related by the correlation functions $\psi_{ij}(x_j, x_i) : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}$. The joint probability over $\mathcal{X}$ is given by

$$P(\mathcal{X}) = \frac{1}{Z} \left(\prod_{(i,j) \in \mathcal{P}} \psi_{ij}(x_j, x_i) \right) \left(\prod_i \phi_i(x_i) \right). \quad (1)$$

The belief propagation algorithm introduces variables such as $m_{ij}(x_j)$ which can intuitively be understood as a *message* from hidden node i to hidden node j about what state node j should be in. Messages are computed iteratively using the update algorithm

$$m_{ij}(x_j) \leftarrow \sum_{x_i \in \mathcal{L}} \left(\phi_i(x_i) \psi_{ij}(x_j, x_i) \prod_{k \in \mathcal{P}(i) \setminus j} m_{ki}(x_i) \right) \quad (2)$$

and the marginal distribution of X_i (the node's *belief*) is given by

$$b_i(x_i) \propto \phi_i(x_i) \prod_{j \in \mathcal{P}(i)} m_{ji}(x_i). \quad (3)$$

2.2. Belief propagation in particle networks

When a problem calls for continuous-valued random variables $\mathbf{X}_i \in \mathbb{R}^D$ then $\phi_i(\mathbf{x}_i) : \mathbb{R}^D \rightarrow \mathbb{R}$ and $\psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ and (1) now defines a probability density. In the continuous analogues of (2) and (3) now $m_{ij}(\cdot)$ and $b_i(\cdot)$ are functions rather than L -element arrays; in fact we will assume that they are normalisable functions which can be interpreted as probability density functions. Now (2) becomes

$$m_{ij}(\mathbf{x}_j) \leftarrow \int_{\mathbb{R}^D} \psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) \phi_i(\mathbf{x}_i) \prod_{k \in \mathcal{P}(i) \setminus j} m_{ki}(\mathbf{x}_i) d\mathbf{x}_i. \quad (4)$$

If the structure of the network is such that the marginal distribution at each node is Gaussian, this integral can be performed exactly [19]. As noted above, if any of the m_{ij} or ϕ_i violate this conditional linear Gaussian structure exact inference is intractable and the m_{ij} must be replaced by an approximation; the PAMPAS algorithm uses particle sets.

At the heart of any algorithm for belief propagation with particle sets is a Monte-Carlo approximation to the integral in (4). We set out here a general algorithmic framework for performing this approximation and in section 2.4 describe the PAMPAS algorithm which is a specific implementation suitable for common computer vision models.

The function

$$p_{ij}^M(\mathbf{x}_i) \equiv \frac{1}{Z_{ij}} \phi_i(\mathbf{x}_i) \prod_{k \in \mathcal{P}(i) \setminus j} m_{ki}(\mathbf{x}_i) \quad (5)$$

is denoted the *foundation* of message $m_{ij}(\cdot)$ where Z_{ij} is a constant of proportionality which turns $p_{ij}^M(\cdot)$ into a probability density. A Monte-Carlo integration of (4) yields $\tilde{m}_{ij}$, an approximation of m_{ij} , by drawing N samples from the foundation $s_{ij}^n \sim p_{ij}^M(\cdot)$ and setting

$$\tilde{m}_{ij}(\mathbf{x}_j) = \frac{1}{N} \sum_{n=1}^N \psi_{ij}(\mathbf{x}_j, s_{ij}^n). \quad (6)$$

In general it is possible to pick an *importance function* $g_{ij}(\mathbf{x}_i)$ from which to generate samples $s_{ij}^n \sim g_{ij}(\cdot)$ with (unnormalised) importance weights $\pi_{ij}^n \propto p_{ij}^M(s_{ij}^n)/g(s_{ij}^n)$ in which case the $\tilde{m}_{ij}$ is a weighted mixture

$$\tilde{m}_{ij}(\mathbf{x}_j) = \frac{1}{\sum_{k=1}^N \pi_{ij}^k} \sum_{n=1}^N \pi_{ij}^n \psi_{ij}(\mathbf{x}_j, s_{ij}^n). \quad (7)$$

When the marginal distribution over $\mathbf{x}_i$ is required, samples s_i^n can be drawn from the estimated belief distribution

$$\tilde{b}_i(\mathbf{x}_i) = \frac{1}{Z_i} \phi_i(\mathbf{x}_i) \prod_{j \in \mathcal{P}(i)} \tilde{m}_{ji}(\mathbf{x}_i) \quad (8)$$

either directly or by importance sampling as above and these samples can be used to compute expectations over the belief, for example its mean and higher moments. By fixing a set of samples s_i^n at each node a Gibbs sampler can be used to generate samples from the joint distribution over the graph and this is described in section 2.4.

2.3. Choice of importance function

Some limitations of the approach in section 2.2 can be seen by considering a simple example. Suppose that we wish to perform inference on a three-node network $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ which has the chain structure:

$$\mathcal{P}(1) = (2), \quad \mathcal{P}(2) = (1, 3), \quad \mathcal{P}(3) = (2).$$

Two distinct inference chains are being sampled: $\mathbf{X}_1 \rightarrow \mathbf{X}_2 \rightarrow \mathbf{X}_3$ and $\mathbf{X}_3 \rightarrow \mathbf{X}_2 \rightarrow \mathbf{X}_1$ but the information is not combined in any of the messages; there is no “mixing” between the forward and backward directions. This situation also arises whenever there is a layered structure to the graph with a set of low-level nodes passing information up to higher levels and high-level information propagating back down. Of course the belief does correctly incorporate information from both directions, but the efficiency of a Monte Carlo algorithm is strongly dependent on the sample positions so if possible we would like to use all the available information when choosing these positions.

One solution is to use importance sampling for some of the particles [16] and a natural choice of function is the current belief estimate $g(\mathbf{x}_i) = \tilde{b}_i(\mathbf{x}_i)$ in which case the importance weights are $\pi_{ij}^n = 1/\tilde{m}_{ji}(s_{ij}^n)$. In the context of the

simple chain given above this algorithm amounts to *smoothing* the distribution. Smoothing has been proposed for particle filters [12] but previous algorithms merely updated the particle weights in a backward pass rather than generating new particle positions localised in regions of high belief.

2.4. The PAMPAS algorithm

A property of many computer vision models is that the potentials $\psi_{ij}(\mathbf{x}_j, \mathbf{x}_i)$ can be written as a mixture of Gaussians and that the likelihoods $\phi_i(\mathbf{x}_i)$ are complex and difficult to sample from but can be evaluated up to a multiplicative constant. We now describe the PAMPAS algorithm which is specialised to perform belief propagation for this type of model; section 2.5 includes a brief sketch of a more general algorithm [24] for which these Gaussian restrictions do not apply.

For notational simplicity we describe the case that each potential is a single Gaussian plus a Gaussian outlier process though the extension to mixtures of Gaussians is straightforward. Setting λ^o to be the fixed outlier probability and μ_{ij} and Λ_{ij} the parameters of the outlier process,

$$\psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) = (1 - \lambda^o) \mathcal{N}(\mathbf{x}_j; f_{ij}(\mathbf{x}_i), G_{ij}(\mathbf{x}_i)) + \lambda^o \mathcal{N}(\mathbf{x}_j; \mu_{ij}, \Lambda_{ij}) \quad (9)$$

where $f_{ij}(\cdot)$ and $G_{ij}(\cdot)$ are deterministic functions, potentially non-linear, respectively computing the mean and covariance matrix of the conditional Gaussian distribution. Each message approximation $\tilde{m}_{ij}$ is now a mixture of N Gaussians:

$$\tilde{m}_{ij}(\mathbf{x}_j) = \frac{(1 - \lambda^o)}{\sum_{k=1}^{N'} \pi_{ij}^k} \sum_{n=1}^{N'} \pi_{ij}^n \mathcal{N}(\mathbf{x}_j; f_{ij}(s_{ij}^n), G_{ij}(s_{ij}^n)) + \lambda^o \mathcal{N}(\mathbf{x}_j; \mu_{ij}, \Lambda_{ij}) \quad (10)$$

where $N' = N - 1$ is the number of samples drawn during the Monte-Carlo integration. The message is summarised by N triplets:

$$\tilde{m}_{ij} = \{(\pi_{ij}^n, s_{ij}^n, \Lambda_{ij}^n) : 1 \leq n \leq N\}. \quad (11)$$

where π_{ij}^N , s_{ij}^N and Λ_{ij}^N correspond to the outlier process and are fixed throughout the algorithm. A potential which encodes a mixture of J Gaussians with K Gaussian outliers will lead to $N' = (N - K)/J$ which is only practical for fairly small values of J .

As indicated in section 2.3 it makes sense to importance-sample some fraction of the message particles from the approximate belief distribution. In fact, because ϕ_i is assumed to be hard to sample from but easy to evaluate, we will use importance sampling for all the particles but with two

distinct importance distributions and corresponding weight functions. A fraction $(1 - \nu)N'$ of the particles, by slight abuse of terminology, are denoted direct samples:

$$\begin{aligned} \mathbf{s}_{ij}^n &\sim \frac{1}{\bar{Z}_{ij}} \prod_{k \in \mathcal{P}(i) \setminus j} \tilde{m}_{ki}(\mathbf{x}_i) \\ \pi_{ij}^n &= \phi_i(\mathbf{s}_{ij}^n) \end{aligned} \quad (12)$$

and the remaining $\nu N'$ particles we will refer to as importance samples:

$$\begin{aligned} \mathbf{s}_{ij}^n &\sim \frac{1}{\bar{Z}_{ij}} \prod_{k \in \mathcal{P}(i)} \tilde{m}_{ki}(\mathbf{x}_i) \\ \pi_{ij}^n &= \phi_i(\mathbf{s}_{ij}^n) / \tilde{m}_{ji}(\mathbf{s}_{ij}^n). \end{aligned} \quad (13)$$

Note that if the network is a forward chain where $\mathcal{P}(i) \equiv i - 1$ and $\nu = 0$ the algorithm reduces exactly to a standard form of particle filtering [11].

Both (12) and (13) require sampling from a foundation F which is the product of D mixtures indexed by a label vector $L = (l_1, \dots, l_D)$:

$$F(\cdot) = \frac{1}{Z} \sum_L \eta_L \mathcal{N}(\cdot; \mu_L, \Lambda_L) \quad (14)$$

where from the standard Gaussian product formula

$$\Lambda_L^{-1} = \sum_{d=1}^D (\Lambda_d^{l_d})^{-1} \quad \Lambda_L^{-1} \mu_L = \sum_{d=1}^D (\Lambda_d^{l_d})^{-1} \mu_d^{l_d} \quad (15)$$

$$\eta_L = \frac{\prod_{d=1}^D \pi_d^{l_d} \mathcal{N}(\cdot; \mu_d^{l_d}, \Lambda_d^{l_d})}{\mathcal{N}(\cdot; \mu_L, \Lambda_L)}. \quad (16)$$

This product F is a Gaussian mixture with N^D components so direct sampling is effectively infeasible for $D > 3$ for reasonable N . Methods for overcoming this difficulty are discussed in sections 2.5 and 4. The PAMPAS message update algorithm is given in figure 1.

When N belief samples $\mathbf{s}_i^n$ have been drawn for each node in $\mathcal{X}$ they can be used to sample from the joint distribution over the graph. This can be done by Gibbs sampling from the discrete probability distribution over labels $L' = (l_1, \dots, l_M)$ where L' indexes a sample $\mathbf{s}_m^{l_m}$ at each of the M nodes. The Gibbs sampler updates each component of this label vector in turn where the marginal probability of choosing label l_d when the other labels are fixed is

$$P(l_d = n) = \phi_d(\mathbf{s}_i^n) \prod_{j \in \mathcal{P}(i)} \psi_{ji}(\mathbf{s}_i^n, \mathbf{s}_j^{l_j}). \quad (17)$$

2.5. Comparison with the NBP algorithm

Sudderth et al [23, 24] have independently developed an almost identical algorithm for performing belief propagation with the aid of particle sets, which they call Non-Parametric Belief Propagation, or NBP. In order to deal with

1. Draw samples from the incoming message product.
 - (a) For $1 \leq n \leq (1 - \nu)(N - 1)$:
 - i. Draw $\tilde{\mathbf{s}}_{ij}^n \sim \prod_{k \in \mathcal{P}(i) \setminus j} \tilde{m}_{ki}(\cdot)$.
 - ii. Set $\tilde{\pi}_{ij}^n = 1/(N - 1)$.
 - (b) For $(1 - \nu)(N - 1) < n \leq N - 1$:
 - i. Draw $\tilde{\mathbf{s}}_{ij}^n \sim \prod_{k \in \mathcal{P}(i)} \tilde{m}_{ki}(\cdot)$.
 - ii. Set $\gamma_{ij}^n = 1/\tilde{m}_{ji}(\tilde{\mathbf{s}}_{ij}^n)$.
 - (c) For $(1 - \nu)(N - 1) < n \leq N - 1$:
 - i. Set $\tilde{\pi}_{ij}^n = \nu \gamma_{ij}^n / \sum_{k=1+(1-\nu)(N-1)}^{N-1} \gamma_{ij}^k$.
2. Apply importance correction from likelihood. For $1 \leq n \leq N - 1$:
 - (a) Set $\tilde{\pi}_{ij}^n = \tilde{\pi}_{ij}^n \phi_i(\tilde{\mathbf{s}}_{ij}^n)$
3. Store normalised weights and mixture components. For $1 \leq n \leq N - 1$:
 - (a) Set $\pi_{ij}^n = (1 - \pi_{ij}^N) \tilde{\pi}_{ij}^n / \sum_{k=1}^{N-1} \tilde{\pi}_{ij}^k$
 - (b) Set $\mathbf{s}_{ij}^n = f_{ij}(\tilde{\mathbf{s}}_{ij}^n)$.
 - (c) Set $\Lambda_{ij}^n = G_{ij}(\tilde{\mathbf{s}}_{ij}^n)$.

Figure 1. The message update algorithm.

the exponential blowup in the message product as D increases, they introduce a Gibbs sampler which is sketched below. We have used this Gibbs sampler for the central node in the jointed object model of section 3.3 which has 4 incoming messages.

The main remaining difference between the algorithms is the way in which the message foundations are interpreted. In [24] N samples $\mathbf{r}_{ij}^n$ are generated from $p_{ij}^M(\mathbf{x}_i)$ and then for each $\mathbf{r}_{ij}^n$ a single sample $\mathbf{s}_{ij}^n$ is generated by sampling from the potential $\psi_{ij}(\cdot, \mathbf{r}_{ij}^n)$. The message mixture $\tilde{m}_{ij}(\mathbf{x}_j)$ is then formed by placing identical diagonal-covariance Gaussian kernels about the $\mathbf{s}_{ij}^n$. This additional smoothing kernel leads to variance estimates which are biased upwards of their true values. This step is unnecessary when the potentials are small mixtures of Gaussians so we omit it; thus we achieve unbiased kernel estimates as well as generating less noisy samples from the message distribution, and in addition we do not have the problem of selecting a kernel width. The advantage of the technique in [24] is that it allows more general potentials which are not mixtures of Gaussians. Note that if the potentials are complex (for example a mixture of $J = 100$ Gaussians) neither approach will adequately represent the product unless N is very large. The two algorithms also differ in the application of ϕ_i and NBP is specialised to models where ϕ_i is slowly

varying with respect to the kernel bandwidth.

Sudderth et al [23, 24] propose a Gibbs sampler which performs approximate sampling from the message product F in $O(KDN)$ operations per sample, where K is the number of iterations of the sampler. A sample is drawn from F by first choosing a label vector L using a Gibbs sampler and then generating a random sample from the Gaussian F_L . The Gibbs sampler works by sampling l_d with all the other labels in L held fixed for each d in turn. With all of L but l_d fixed the marginal product-component weights are given by

$$\eta_d^n \propto \frac{\pi_d^n \mathcal{N}(\cdot; \mu_d^n, \Lambda_d^n)}{\mathcal{N}(\cdot; \mu_{L^n}, \Lambda_{L^n})} \quad (18)$$

where $L^n = (l_1, \dots, l_{d-1}, n, l_{d+1}, \dots, l_D)$.

We use this sampler with $K = 100$ in section 3.3 for one message product where $D = 4$. Generating an entire message using the Gibbs sampler takes $O(KDN^2)$ operations which is preferable to brute force sampling when $D > 3$ in cases where KD has the same order of magnitude as N . While [23] and [24] demonstrate that the Gibbs sampler is very effective for some applications, we have found that in clutter the algorithm may generate samples in regions with very low probability mass. This can lead to problems when importance sampling since some of the $\tilde{m}_{ji}(s_{ij}^n)$ in (13) may be very small relative to other samples leading to a high weight π_{ij}^n which suppresses the other samples in the set. As a workaround we adopted a heuristic to smooth these importance weights by replacing any $\pi_{ij}^n > 0.25\nu$ by

$$\pi_{ij}^n \leftarrow (\nu - \pi_{ij}^n)/6 \quad (19)$$

and renormalising (recall from step 1c in figure 1 that $\pi_{ij}^n < \nu$). We believe that the problem of generating good samples from a product of mixtures when D is large is an open research question, and discuss this further in section 4.

3. Results

3.1. Edge linking models

We exercised the algorithm using a simple edge model. In these experiments each node in the graph corresponds to an "edgel" $\mathbf{x}_i = (x_i, y_i, \alpha_i, \lambda_i) \in \mathbb{R}^4$ where (x_i, y_i) is an image coordinate, α_i is an orientation angle and λ_i is the distance between adjacent edgels. We use a simple diagonal Gaussian conditional distribution model for $j = i+1$ where

$$\begin{aligned} \psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) = & \mathcal{N}(x_j; x_i + \lambda_i \cos \alpha_i, \sigma_p^2) \times \mathcal{N}(y_j; y_i + \lambda_i \sin \alpha_i, \sigma_p^2) \times \\ & \mathcal{N}(\alpha_j; \alpha_i, \sigma_a^2) \times \mathcal{N}(\lambda_j; \lambda_i, \sigma_\lambda^2) \end{aligned}$$

and the model for $j = i-1$ is the same but in reverse. A similar model has been used for pixel-based illusory contour

completion [1, 25] as well as finding segmentation edges using particle filtering [21], and if the ϕ_i are Gaussian it is very similar to the standard snake model [15]. Figure 2 demonstrates the effectiveness of importance sampling using the belief distribution. The top images show messages

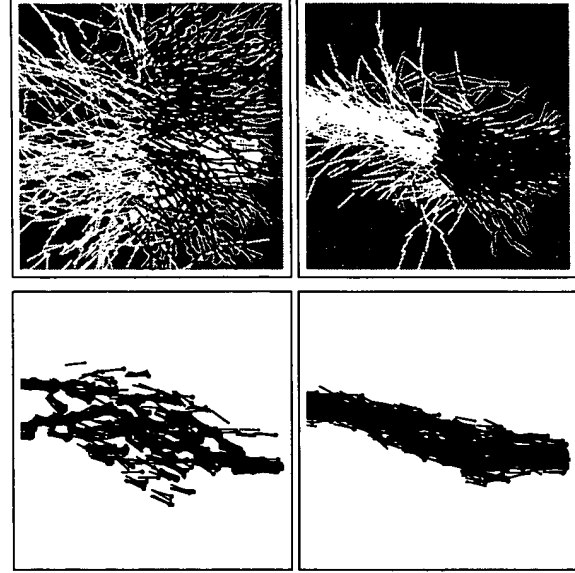


Figure 2. Importance sampling mixes information from forward and backward chains. The top images show the messages passed forwards and backwards along a chain in black and white respectively, and the bottom images show the resulting beliefs. Each displayed edgel is the mean of a Gaussian mixture component. On the left no importance sampling is used and the forward and backward messages do not interact. With importance sampling ($\nu = 0.5$) the message particles are guided to areas of high belief; this is shown on the right after 5 iterations of the algorithm.

and the bottom beliefs, and the ϕ_i are uniform to emphasise the sampling behaviour in the absence of image data. The left column shows a run of the PAMPAS algorithm with no importance sampling ($\nu = 0$) on a 16-link chain where each end of the chain is initialised from a different prior distribution. Since there is no interaction between the forward and backward messages the particles spread away from each end in a random walk producing the noisy belief distribution shown below. The right column shows the same particle sets after 5 iterations of the algorithm now using importance sampling with $\nu = 0.5$. The message particles become con-

centrated in areas of high belief making the belief estimates below less noisy.

In the absence of image measurements the above model is Gaussian and so the belief could be estimated by a variety of methods. The cluttered image data used in figure 3 causes the marginals to become highly non-Gaussian; the algorithm converges on a globally plausible solution despite strong local distractors. Here the ϕ_i used are simple robust gradient-affinity operators. The incorporation of information along the chain in both directions can be expected to improve the performance of e.g. the Jetstream algorithm [21] which is a particle-based edge finder.

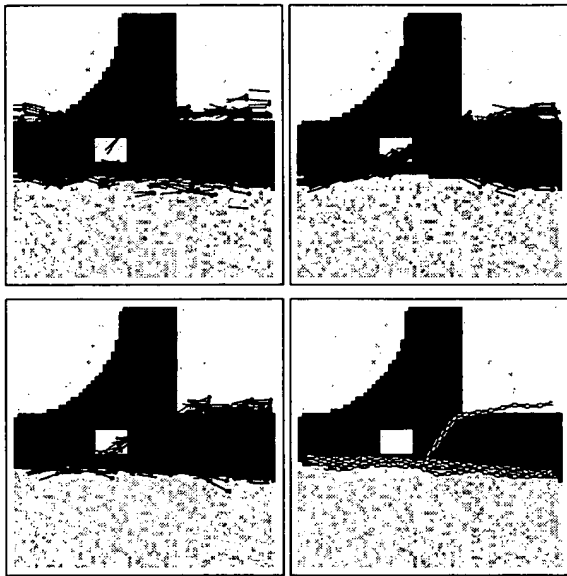


Figure 3. An edge linking model finds global solutions. The images show (in raster order) the belief estimates after 1, 5, and 10 iterations, and 7 samples from the joint graph after 10 iterations. The belief stabilises on a bimodal distribution with most of the weight of the distribution on the continuous edge as can be seen on the bottom right image.

3.2. Illusory contour matching

We tested the algorithm on an illusory-contour finding problem shown in figure 4. The input image has C “pacman”-like corner points where the open mouth defines two edge directions leading away from the corner separated by an angle C_α . The task is to match up the edges with smooth contours. We construct $2C$ chains each of four links, and each is anchored at one end with a prior start-

ing at its designated corner pointing in one of the two edge directions. At the untethered end of a chain from corner c is a “connector” edgel which is attached to (conditionally dependent on) the connectors at the ends of the $2C - 2$ other chains not starting from corner c . We use the same edgel model as in section 3.1 but now each node state-vector $\mathbf{x}_i$ is augmented with a discrete label v_i denoting which of the $2C - 2$ partner chains it matches, and the potentials at the connector are modified so that $\psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) = 0$ when v_i and v_j do not match. This special form of potential allows us to form the message product in $O(N^2C)$ operations rather than $O(N^{2C-1})$ as a naive approach would dictate. Space does not permit a full description of the algorithm here but it is very similar to the Mixed Markov Model approach [8].

Figure 4 shows the model being applied to two scenarios with $C = 6$ in both cases. First, $C_\alpha = 11\pi/36$ which leads to two overlapping triangles with slightly concave edges when the chains are linked by the algorithm. Secondly, $C_\alpha = 7\pi/9$ which leads to a slightly convex regular hexagon. The same parameters are used for the models in both cases, so the algorithm must adapt the edgel lengths to cope with the differing distances between target corners.

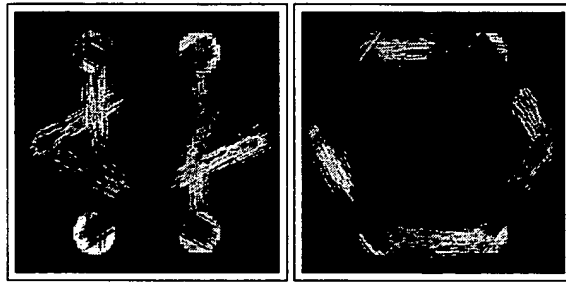


Figure 4. The PAMPAS algorithm performs illusory contour completion. Depending on the angle at which the chains leave the corner positions, the belief converges on either two overlapping triangles or a single hexagon.

3.3. Finding a jointed object in clutter

Finally we constructed a graph representing a jointed object and applied the PAMPAS algorithm (combined with the Gibbs sampler from [24]) to locating the object in a cluttered scene. The model consists of 9 nodes; a central circle with four jointed arms each made up of two rectangular links. The circle node $\mathbf{x}_1 = (x_0, y_0, r_0)$ encodes a position and radius. Each arm node $\mathbf{x}_i = (x_i, y_i, \alpha_i, w_i, h_i)$, $2 \leq i \leq 9$ encodes a position, angle, width and height and prefers one of the four compass directions (to break symmetry). The arms pivot around their inner joints, so the po-

tential to go from an inner arm $2 \leq i \leq 5$ to the outer arms $j = i + 4$ is given by

$$\begin{aligned}x_j &= x_i + h_i \cos(\alpha_i) + \mathcal{N}(\cdot; 0, \sigma_p^2) \\y_j &= y_i + h_i \sin(\alpha_i) + \mathcal{N}(\cdot; 0, \sigma_p^2) \\ \alpha_j &= \alpha_i + \mathcal{N}(\cdot; 0, \sigma_\alpha^2) \\w_j &= w_i + \mathcal{N}(\cdot; 0, \sigma_s^2) \quad h_j = h_i + \mathcal{N}(\cdot; 0, \sigma_s^2)\end{aligned}$$

and the $\psi_{1i}(\mathbf{x}_i, \mathbf{x}_1)$ from the circle outwards are similar. Going from the outer to the inner arm, the potential is not Gaussian but we approximate it with the following:

$$\begin{aligned}x_i &= x_j - h_j \cos(\alpha_i) + \mathcal{N}(\cdot; 0, h_j \sigma_s) \sin((i-2)\pi/2) |\sigma_p^2| \\y_i &= y_j - h_j \sin(\alpha_i) + \mathcal{N}(\cdot; 0, h_j \sigma_s) \cos((i-2)\pi/2) |\sigma_p^2| \\ \alpha_i &= \alpha_j + \mathcal{N}(\cdot; 0, \sigma_\alpha^2) \\w_i &= w_j + \mathcal{N}(\cdot; 0, \sigma_s^2) \quad h_i = h_j + \mathcal{N}(\cdot; 0, \sigma_s^2).\end{aligned}$$

The $\psi_{i1}(\mathbf{x}_1, \mathbf{x}_i)$ are straightforward:

$$\begin{aligned}x_1 &= x_i + \mathcal{N}(\cdot; 0, \sigma_p^2) \quad y_1 = y_i + \mathcal{N}(\cdot; 0, \sigma_p^2) \\r_1 &= 0.5(w_i/\delta_w + h_i/\delta_h) + \mathcal{N}(\cdot; 0, \sigma_s^2).\end{aligned}$$

The object has been placed in image clutter in figure 5. The clutter is made up of 12 circles and 100 rectangles placed randomly in the image. The messages are initialised with a simulated specialised feature detector: $\mathbf{x}_1$ is sampled from positions near the circles in the image and the arms are sampled to be near the rectangles, with rectangles closer to the preferred direction of each arm sampled more frequently. Each node contains $N = 75$ particles so the space is undersampled. After initialisation the PAMPAS algorithm (using the Gibbs sampler from [24] for the circle's messages and belief) is iterated without further recourse to the "feature detector" information. All of the potentials include a Gaussian outlier covering the whole image which allows samples to be generated far from the initialisation particles. After two or three iterations the belief distribution has converged on the object. In this case it might be argued that it would be more straightforward to simply detect the circles and perform an exhaustive search near each. The second example of figure 5 demonstrates the power of the probabilistic modelling approach, since now the circle at the centre of the object has not been displayed, simulating occlusion. Of course no $\mathbf{x}_1$ initialisation samples are generated near the "occluded" circle, but even after one iteration the belief at that node has high weight at the correct location due to the agreement of the nearby arm nodes. After a few iterations the belief has converged on the correct object position despite the lack of the central circle.

4. Conclusion

We have concentrated in this paper on examples with synthetic data in order to highlight the working of the PAM-

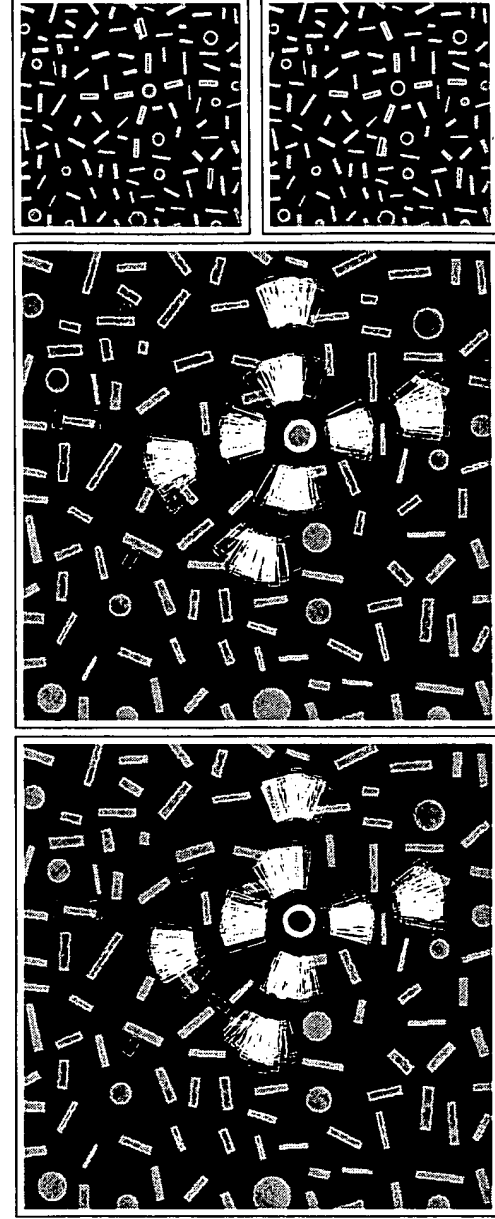


Figure 5. A jointed object is located in clutter. Two experiments are shown, with and without a simulated occlusion. The top two images show a sample from the joint distribution for each experiment, while the lower images show the full distributions. The second experiment models an occluded object with no circle present in the image where the belief has placed it. This experiment uses the Gibbs sampler from [24] to form the message product for the central node.

PAS algorithm. The applications we have chosen are highly suggestive of real-world problems, however, and we hope the promising results will stimulate further research into applying continuous-valued graphical models to a variety of computer vision problems. Sections 3.1 and 3.2 suggest a variety of edge completion applications. Section 3.3 indicates that the PAMPAS algorithm may be effective in locating articulated structures in images, in particular people. Person-finding algorithms exist already for locating and grouping subparts [20] but they do not naturally fit into a probabilistic framework. Recently several researchers have had promising results finding jointed structures in images using discretised graphical model representations [6, 10, 4]. Both discretising and using the PAMPAS algorithm result in approximations, and the tradeoffs between the two approaches remain to be investigated.

One obvious advantage of using a graphical model representation is that it extends straightforwardly to tracking, simply by increasing the size of the graph and linking nodes in adjacent timesteps. Another area of great interest is linking the continuous-valued graph nodes to lower-level nodes which may be discrete-valued as in [13]. In fact, researchers in the biological vision domain have recently proposed a hierarchical probabilistic structure to model human vision which maps closely onto our algorithmic framework [17].

We are performing current research, to be published elsewhere, into the problem of sampling from products of Gaussian mixtures. The Gibbs sampler of [24] is very flexible and straightforward to implement and performs well on some important models, but for cluttered distributions of the type we have encountered K must be set prohibitively large to get reliable samples. We have developed heuristic algorithms to address this clutter problem which perform well on our test applications, but are only practical for $D < 9$. The design of an efficient, generally effective sampler which scales to large D remains an open problem.

Acknowledgements

Helpful suggestions to improve this work were contributed by Michael Black, David Mumford, Mike Burrows, Martin Abadi, Mark Manasse and Ian Reid.

References

- [1] J. August and S.W. Zucker. The curve indicator random field: curve organization via edge correlation. In *Perceptual Organization for Artificial Vision Systems*. Kluwer Academic, 2000.
- [2] C.M. Bishop, D. Spiegelhalter, and J. Winn. Vibes: A variational inference engine for bayesian networks. In *NIPS*, 2002.
- [3] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, 1998.
- [4] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proc. 7th European Conf. Computer Vision*, 2002.
- [5] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [6] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 66–73, 2000.
- [7] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000.
- [8] A. Fridman. Mixed markov models. In *Proc. Second International ICSC Symposium on Neural Computation*, 2000.
- [9] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [10] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *Proc. 8th Int. Conf. on Computer Vision*, volume 1, pages 690–695, 2001.
- [11] M. Isard and A. Blake. Condensation — conditional density propagation for visual tracking. *Int. J. Computer Vision*, 28(1):5–28, 1998.
- [12] M.A. Isard and A. Blake. A smoothing filter for Condensation. In *Proc. 5th European Conf. Computer Vision*, volume 1, pages 767–781, 1998.
- [13] N. Jojic, N. Petrovic, B. J. Frey, and T. S. Huang. Transformed hidden markov models: Estimating mixture models of images and inferring spatial transformations in video sequences. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2000.
- [14] M.I. Jordan, T.J. Sejnowski, and T. Poggio, editors. *Graphical Models: Foundations of Neural Computation*. MIT Press, 2001.
- [15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, 1987.
- [16] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid bayes nets. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence*, pages 324–333, 1999.
- [17] T.S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *Journal of the Optical Society of America A*, 2002 (submitted).
- [18] U. Lerner. *Hybrid Bayesian Networks for Reasoning about Complex Systems*. PhD thesis, Stanford University, October 2002.
- [19] T.P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, January 2001.
- [20] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. 7th European Conf. Computer Vision*, pages 666–680, 2002.
- [21] P. Perez, A. Blake, and M. Gangnet. Jetstream: Probabilistic contour extraction with particles. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 524–531, 2001.
- [22] H. Sidenbladh, M.J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. 7th European Conf. Computer Vision*, volume 1, pages 784–800, 2002.
- [23] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. Technical Report P-2551, MIT Laboratory for Information and Decision Systems, 2002.
- [24] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.
- [25] L.R. Williams and D.W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.
- [26] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR2001-22, MERL, 2001.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

BEST AVAILABLE COPY

MITSUBISHI ELECTRIC RESEARCH LABORATORIES

<http://www.merl.com>

Detecting Pedestrians Using Patterns of Motion and Appearance

Paul Viola
Michael Jones
Daniel Snow

TR2003-90 July 2003

Abstract

This paper describes a pedestrian detection system that integrates image intensity information with motion information. We use a detection style algorithm that scans a detector over two consecutive frames of a video sequence. The detector is trained (using AdaBoost) to take advantage of both motion and appearance information to detect a walking person. Past approaches have built detectors based on motion information or detectors based on appearance information, but ours is the first to combine both sources of information in a single detector. The implementation described runs at about 4 frames/second, detects pedestrians at very small scales (as small as 20x15 pixels), and has a very low false positive rate.

Our approach builds on the detection work of Viola and Jones. Novel contributions of this paper include: i) development of a representation of image motion which is extremely efficient, and ii) implementation of a state of the art pedestrian detection system which operates on low resolution images under difficult conditions (such as rain and snow).

*To appear in the Proceedings of the International Conference on Computer Vision (ICCV),
October 13, 2003, Nice, France.*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2003
201 Broadway, Cambridge, Massachusetts 02139

Submitted March 2003.

Detecting Pedestrians Using Patterns of Motion and Appearance

Paul Viola Michael J. Jones Daniel Snow
Microsoft Research Mitsubishi Electric Research Labs Mitsubishi Electric Research Labs
viola@microsoft.com mjones@merl.com snow@merl.com

Abstract

This paper describes a pedestrian detection system that integrates image intensity information with motion information. We use a detection style algorithm that scans a detector over two consecutive frames of a video sequence. The detector is trained (using AdaBoost) to take advantage of both motion and appearance information to detect a walking person. Past approaches have built detectors based on motion information or detectors based on appearance information, but ours is the first to combine both sources of information in a single detector. The implementation described runs at about 4 frames/second, detects pedestrians at very small scales (as small as 20x15 pixels), and has a very low false positive rate.

Our approach builds on the detection work of Viola and Jones. Novel contributions of this paper include: i) development of a representation of image motion which is extremely efficient, and ii) implementation of a state of the art pedestrian detection system which operates on low resolution images under difficult conditions (such as rain and snow).

1 Introduction

Pattern recognition approaches have achieved measurable success in the domain of visual detection. Examples include face, automobile, and pedestrian detection [14, 11, 13, 1, 9]. Each of these approaches use machine learning to construct a detector from a large number of training examples. The detector is then scanned over the entire input image in order to find a pattern of intensities which is consistent with the target object. Experiments show that these systems work very well for the detection of faces, but less well for pedestrians, perhaps because the images of pedestrians are more varied (due to changes in body pose and clothing). Detection of pedestrians is made even more difficult in surveillance applications, where the resolution of the images is very low (e.g. there may only be 100-200 pixels on the target). Though improvement of pedestrian detection using better functions of image intensity is a valuable pursuit, we take a different approach.

This paper describes a pedestrian detection system that integrates intensity information with motion information. The pattern of human motion is well known to be readily distinguishable from other sorts of motion. Many recent papers have used motion to recognize people and in some cases to detect them ([8, 10, 7, 3]). These approaches have a much different flavor from the face/pedestrian detection approaches mentioned above. They typically try to track moving objects over many frames and then analyze the motion to look for periodicity or other cues.

Detection style algorithms are fast, perform exhaustive search over the entire image at every scale, and are trained using large datasets to achieve high detection rates and very low false positive rates. In this paper we apply a detection style approach using information about motion *as well as* intensity information. The implementation described is very efficient, detects pedestrians at very small scales (as small as 20x15 pixels), and has a very low false positive rate. The system is trained on full human figures and does not currently detect occluded or partial human figures.

Our approach builds on the detection work of Viola and Jones [14]. Novel contributions of this paper include: i) development of a representation of image motion which is extremely efficient, and ii) implementation of a state of the art pedestrian detection system which operates on low resolution images under difficult conditions (such as rain and snow).

2 Related Work

The field of human motion analysis is quite broad and has a history stretching back to the work of Hoffman and Flinchbaugh [6]. Much of the field has presumed that the moving object has been detected, and that the remaining problem is to recognize, categorize, or analyze the long-term pattern of motion. Interest has recently increased because of the clear application of these methods to problems in surveillance. An excellent overview of related work in this area can be found in the influential paper by Cutler and Davis [3]. Cutler and Davis describe a system that is more direct than most, in that it works directly on images which can be of low resolution and poor quality. The system measures

periodicity robustly and directly from the tracked images. Almost all other systems require complex intermediate representations, such as points on the tracked objects or the segmentation of legs. Detection failures for these intermediates will lead to failure for the entire system.

In contrast our system works directly with images extracting short term patterns of motion, as well as appearance information, to detect all instances of potential objects. There need not be separate mechanisms for tracking, segmentation, alignment, and registration which each involve parameters and adjustment. One need only select a feature set, a scale for the training data, and the scales used for detection. All remaining tuning and adjustment happens automatically during the training process.

Since our analysis is short term, the absolute false positive rate of our technique is unlikely to be as low as might be achieved by long-term motion analysis such as Cutler and Davis. Since the sources of motion are large, random processes may generate plausible "human" motion in the short term. In principle the two types of techniques are quite complimentary, in that hypothetical objects detected by our system could be verified using a long-term analysis.

The field of object detection is equally broad. To our knowledge there are no other systems which perform direct detection of pedestrians using both intensity and motion information. Key related work in the area use static intensity images as input. The system of Gavrilu and Philomen [4] detects pedestrians in static images by first extracting edges and then matching to a set of exemplars. This is a highly optimized and practical system, and it appears to have been a candidate for inclusion in Mercedes automobiles. Nevertheless published detection rates were approximately 75% with a false positive rate of 2 per image.

Other related work includes that of Papageorgiou et. al [9]. This system detects pedestrians using a support vector machine trained on an overcomplete wavelet basis. Because there is no widely available testing set, a direct comparison with our system is not possible. Based on the published experiments, the false positive rate of this system was significantly higher than for the related face detection systems. We conjecture that Papageorgiou et al.'s detection performance on static images would be very similar to our experiments on static images alone. In this case the false positive rate for pedestrian detection on static images is approximately 10 times higher than on motion pairs.

3 Detection of Motion Patterns

The dynamic pedestrian detector that we built is based on the simple rectangle filters presented by Viola and Jones [14] for the static face detection problem. We first extend these filters to act on motion pairs. The rectangle filters proposed by Viola and Jones can be evaluated extremely

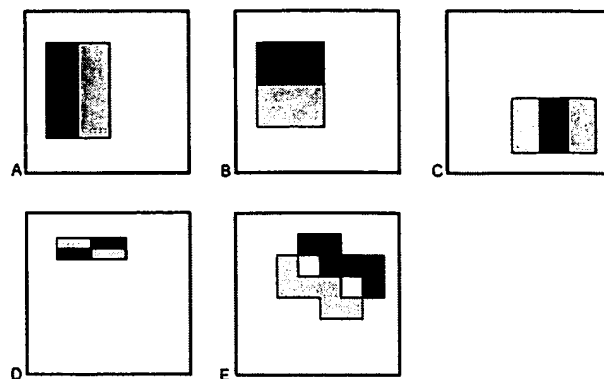


Figure 1: Example rectangle filters shown relative to the enclosing detection window. The sum of the pixels which lie within the lighter rectangles are subtracted from the sum of pixels in the darker rectangles. For three-rectangle filters the sum of pixels in the darker rectangle is multiplied by 2 to account for twice as many lighter pixels.

rapidly at any scale (see figure 1). They measure the differences between region averages at various scales, orientations, and aspect ratios. While these features are somewhat limited, experiments demonstrate that they provide useful information that can be boosted to perform accurate classification.

Motion information can be extracted from pairs or sequences of images in various ways, including optical flow and block motion estimation. Block motion estimation requires the specification of a comparison window, which determines the scale of the estimate. This is not entirely compatible with multi-scale object detection. In the context of object detection, optical flow estimation is typically quite expensive, requiring 100s or 1000s of operations per pixel.

Let us propose a natural generalization of the Viola-Jones features which operate on the differences between pairs of images in time. Clearly some information about motion can be extracted from these differences. For example, regions where the sum of the absolute values of the differences is large correspond to motion. Information about the direction of motion can be extracted from the difference between shifted versions of the second image in time with the first image.

Motion filters operate on 5 images:

$$\Delta = abs(I_t - I_{t+1})$$

$$U = abs(I_t - I_{t+1} \uparrow)$$

$$L = abs(I_t - I_{t+1} \leftarrow)$$

$$R = abs(I_t - I_{t+1} \rightarrow)$$

$$D = abs(I_t - I_{t+1} \downarrow)$$

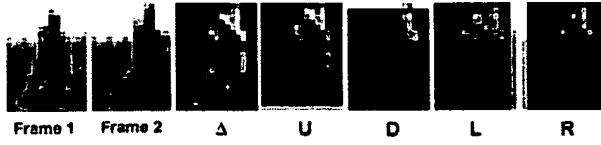


Figure 2: An example of the various shifted difference images used in our algorithm. The first two images are two typical frames with a low resolution pedestrian pictured. The following images show the Δ , U , D , L and R images described in the text. Notice that the right-shifted image difference (R) which corresponds to the direction of motion shown in the two frames has the lowest energy.

where I_t and I_{t+1} are images in time, and $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ are image shift operators ($I_t \uparrow$ is I_t shifted up by one pixel). See Figure 2 for an example of these images.

One type of filter compares sums of absolute differences between Δ and one of $\{U, L, R, D\}$

$$f_i = r_i(\Delta) - r_i(S)$$

where S is one of $\{U, L, R, D\}$ and $r_i(\cdot)$ is a single box rectangular sum within the detection window. These filters extract information related to the likelihood that a particular region is moving in a given direction.

The second type of filter compares sums within the same motion image:

$$f_j = \phi_j(S)$$

where ϕ_j is one of the rectangle filters shown in figure 1. These features measure something closer to motion shear.

Finally, a third type of filter measures the magnitude of motion in one of the motion images:

$$f_k = r_k(S)$$

where S is one of $\{U, L, R, D\}$ and $r_k(\cdot)$ is a single box rectangular sum within the detection window.

We also use appearance filters which are simply rectangle filters that operate on the first input image, I_t :

$$f_m = \phi(I_t)$$

The motion filters as well as appearance filters can be evaluated rapidly using the “integral image” [14, 2] of $\{I_t, \Delta, U, L, R, D\}$.

Because the filters shown in figure 1 can have any size, aspect ratio or position as long as they fit in the detection window, there are very many possible motion and appearance filters. The learning algorithm selects from this huge library of filters to build the best classifier for separating positive examples from negative examples.

A classifier, C , is a thresholded sum of features:

$$C(I_t, I_{t+1}) = \begin{cases} 1 & \text{if } \sum_{i=1}^N F_i(I_t, \Delta, U, L, R, D) > \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

A feature, F , is simply a thresholded filter that outputs one of two votes.

$$F_i(I_t, I_{t+1}) = \begin{cases} \alpha & \text{if } f_i(I_t, \Delta, U, L, R, D) > t_i \\ \beta & \text{otherwise} \end{cases} \quad (2)$$

where $t_i \in \mathcal{R}$ is a feature threshold and f_i is one of the motion or appearance filters defined above. The real-valued α and β are computed during AdaBoost learning (as is the filter, filter threshold t_i and classifier threshold θ).

In order to support detection at multiple scales, the image shift operators $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ must be defined with respect to the detection scale. This ensures that measurements of motion velocity are made in a scale invariant way. Scale invariance is achieved during the training process simply by scaling the training images to a base resolution of 20 by 15 pixels. The scale invariance of the detection is achieved by operating on image pyramids. Initially the pyramids of I_t and I_{t+1} are computed. Pyramid representations of $\{\Delta, U, L, R, D\}$ are computed as follows:

$$\Delta^l = \text{abs}(I_t^l - I_{t+1}^l)$$

$$U^l = \text{abs}(I_t^l - I_{t+1}^l \uparrow)$$

$$L^l = \text{abs}(I_t^l - I_{t+1}^l \leftarrow)$$

$$R^l = \text{abs}(I_t^l - I_{t+1}^l \rightarrow)$$

$$D^l = \text{abs}(I_t^l - I_{t+1}^l \downarrow)$$

where X^l refers to the l -th level of the pyramid. Classifiers (and features) which are learned on scaled 20x15 training images, operate on each level of the pyramid in a scale invariant fashion. In our experiments we used a scale factor of 0.8 to generate each successive layer of the pyramid and stopped when the image size was less than 20x15 pixels.

4 Training Process

The training process uses AdaBoost to select a subset of features and construct the classifier. In each round the learning algorithm chooses from a heterogeneous set of filters, including the appearance filters, the motion direction filters, the motion shear filters, and the motion magnitude filters. The AdaBoost algorithm also picks the optimal threshold for each feature as well as the α and β votes of each feature. The output of the AdaBoost learning algorithm is a classifier that consists of a linear combination of the selected features. For details on AdaBoost, the reader is referred to [12, 5]. The important aspect of the resulting classifier to

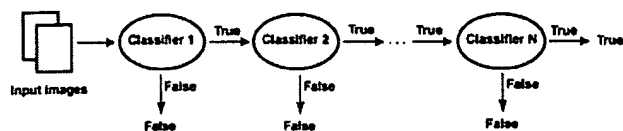


Figure 3: Cascade architecture. Input is passed to the first classifier with decides true or false (pedestrian or not pedestrian). A false determination halts further computation and causes the detector to return false. A true determination passes the input along to the next classifier in the cascade. If all classifiers vote true then the input is classified as a true example. If any classifier votes false then computation halts and the input is classified as false. The cascade architecture is very efficient because the classifiers with the fewest features are placed at the beginning of the cascade, minimizing the total required computation.

note is that it mixes motion and appearance features. Each round of AdaBoost chooses from the total set of the various motion and appearance features, the feature with lowest weighted error on the training examples. The resulting classifier balances intensity and motion information in order to maximize detection rates.

Viola and Jones [14] showed that a single classifier for face detection would require too many features and thus be too slow for real time operation. They proposed a cascade architecture to make the detector extremely efficient (see figure 3). We use the same cascade idea for pedestrian detection. Each classifier in the cascade is trained to achieve very high detection rates, and modest false positive rates. Simpler detectors (with a small number of features) are placed earlier in the cascade, while complex detectors (with a large number of features are placed later in the cascade). Detection in the cascade proceeds from simple to complex.

Each stage of the cascade consists of a classifier trained by the AdaBoost algorithm on the true and false positives of the previous stage. Given the structure of the cascade, each stage acts to reduce both the false positive rate and the detection rate of the previous stage. The key is to reduce the false positive rate more rapidly than the detection rate.

A target is selected for the minimum reduction in false positive rate and the maximum allowable decrease in detection rate. Each stage is trained by adding features until the target detection and false positives rates are met on a validation set. Stages of the cascade are added until the overall target for false positive and detection rate is met.



Figure 5: A small sample of positive training examples. A pair of image patterns comprise a single example for training.

5 Experiments

5.1 Dataset

We created a set of video sequences of street scenes with all pedestrians marked with a box in each frame. We have eight such sequences, each containing around 2000 frames. One frame of each sequence used for training along with the manually marked boxes is shown in figure 4.

5.1.1 Training Set

We used six of the sequences to create a training set from which we learned both a dynamic pedestrian detector and a static pedestrian detector. The other two sequences were used to test the detectors. The dynamic detector was trained on consecutive frame pairs and the static detector was trained on static patterns and so only uses the appearance filters described above. The static pedestrian detector uses the same basic architecture as the face detector described in [14].

Each stage of the cascade is a boosted classifier trained using a set of 2250 positive examples and 2250 negative examples. Each positive training example is a pair of 20 x 15 pedestrian images taken from two consecutive frames of a video sequence. Negative examples are similar image

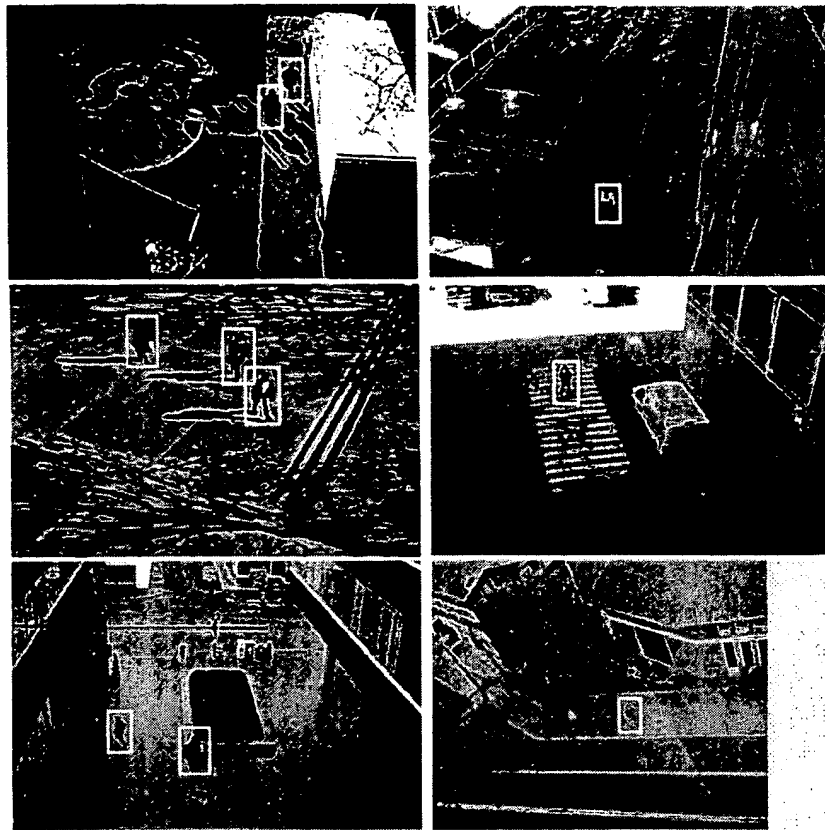


Figure 4: Sample frames from each of the 6 sequences we used for training. The manually marked boxes over pedestrians are also shown.

pairs which do not contain pedestrians. Positive examples are shown in figure 5. During training, all example images are variance normalized to reduce contrast variations. The same variance normalization computation is performed during testing.

Each classifier in the cascade is trained using the original 2250 positive examples plus 2250 false positives from the previous stages of the cascade. The resulting classifier is added to the current cascade to construct a new cascade with a lower false positive rate. The detection threshold of the newly added classifier is adjusted so that the false negative rate is very low. The threshold is set using a validation set of image pairs.

Validation is performed using full images which contain marked positive examples. The validation set for these experiments contains 200 frame pairs. The threshold of the newly added classifier is set so that at least 99.5% of the pedestrians that were correctly detected after the last stage are still correctly detected while at least 10% of the false positives after the last stage are eliminated. If this target cannot be met then more features are added to the current

classifier.

The cascade training algorithm also requires a large set of image pairs to scan for false positives. These false positives form the negative training examples for the subsequent stages of the cascade. We use a set of 4600 full image pairs which do not contain pedestrians for this purpose. Since each full image contains about 50,000 patches of 20x15 pixels, the effective pool of negative patches is larger than 20 million.

The static pedestrian detector is trained in the same way on the same set of images. The only difference in the training process is the absence of motion information. Instead of image pairs, training examples consist of static image patches.

5.2 Training the Cascade

The dynamic pedestrian detector was trained using 54,624 filters which were uniformly subsampled from the much larger set of all filters that fit in a 20 x 15 pixel window.

The static detector was trained using 24,328 filters also

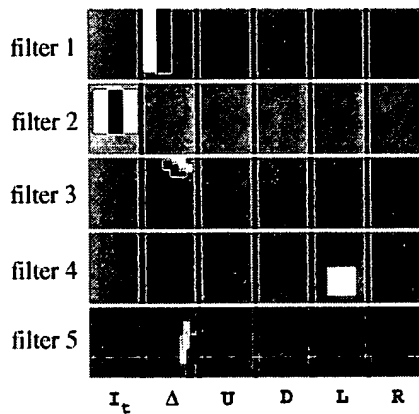


Figure 6: The first 5 filters learned for the dynamic pedestrian detector. The 6 images used in the motion and appearance representation are shown for each filter.

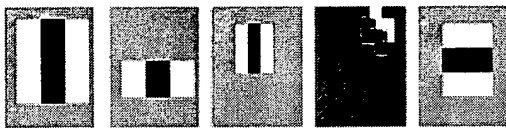


Figure 7: The first 5 filters learned for the static pedestrian detector.

uniformly subsampled from the total possible set. There are fewer filters in the static set because the various types of motion filters are not used.

The first 5 features learned for the dynamic detector are shown in figure 6. It is clear from these that the detector is taking advantage of the motion information in the examples. The first filter, for example, is looking for a difference in the motion near the left edge of the detection box compared to motion in the interior. This corresponds to the fact that our examples have the pedestrian roughly centered and so the motion is greatest in the interior of the detection box. The second filter is acting on the first image of the input image pair. It corresponds to the fact that pedestrians tend to be in the middle of the detection box and stand out from the background. Four of the first 5 filters act on one of the motion images. This confirms the importance of using motion cues to detect pedestrians.

The first 5 features learned for the static detector are shown in figure 7. The first filter is similar to the second filter learned in the dynamic case. The second filter is similar but focuses on the legs. The third filter focuses on the torso.

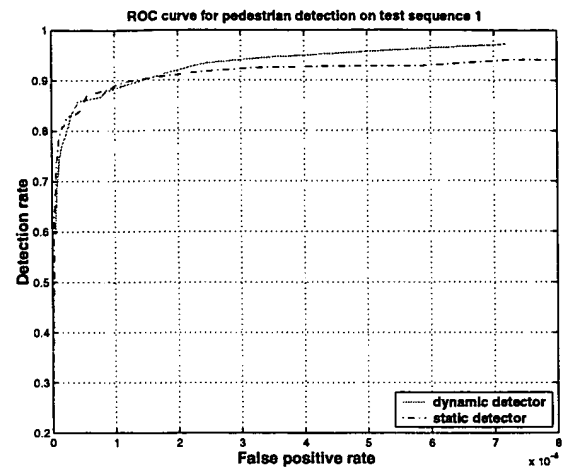


Figure 8: ROC curve on test sequence 1 for both the dynamic and static pedestrian detectors. Both detectors achieve a detection rate of about 80% with a false positive rate of about 1/400,000 (which corresponds to about 1 false positive every 2 frames for the 360x240 pixel frames of this sequence).

5.3 Detection Results

Some example detections for the dynamic detector are shown in figure 10. The detected boxes are displayed on the first frame of each image pair. Example detections for the static case are shown in figure 11. It is clear from these examples that the static detector usually has many more false positives than the dynamic detector. The top row of examples are taken from the PETS 2001 dataset. The PETS sequences were acquired independently using a different camera and a different viewpoint from our training sequences. The detector seems to generalize to these new sequences quite well. The bottom row shows test performed using our own test data. The left two image sequences were not used during training. The bottom right image is a frame from the same camera and location used during training, though at a different time. This image shows that the dynamic detector works well under difficult conditions such as rain and snow, though these conditions did not occur in the training data.

We also ran both the static and dynamic detectors over test sequences for which ground truth is available. From these experiments the false positive rate (the number of false positives across all the frames divided by the total number of patches tested in all frames) and a false negative rate (the total number of false negatives in all frames divided by the total number of faces in all frames) were estimated.

Computation of a full ROC curve is not a simple matter, since the false positive and negative rates depend on the threshold chosen for all layers of the cascade. By adjust-

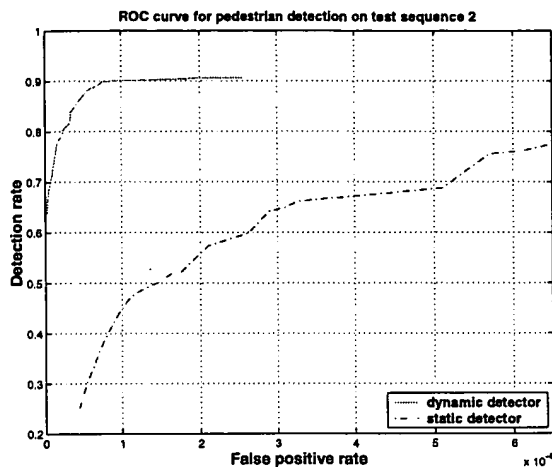


Figure 9: ROC curve on test sequence 2 for both the dynamic and static pedestrian detectors. The dynamic detector has much greater accuracy in this case. At a detection rate of 80%, the dynamic detector has a false positive rate of about 1/400,000 while the static detector has a false positive rate of about 1/15,000.

ing these thresholds one at a time we get a ROC curve for both the dynamic and static detectors. These ROC curves are shown in figures 8 and 9. The ROC curve for the dynamic case is an order of magnitude better on sequence 2 than the static case. On sequence 1, the dynamic detector is only slightly better than the static one. This is probably due to the fact that sequence 2 has some highly textured areas such as the tree and grass that are more likely to cause false positives in the static case.

6 Conclusions

We have presented a detection style algorithm which combines motion and appearance information to build a robust model of walking humans. This is the first approach that we are aware of that combines both motion and appearance in a single model. Our system robustly detects pedestrians from a variety of viewpoints with a low false positive rate.

The basis of the model is an extension of the rectangle filters from Viola and Jones to the motion domain. The advantage of these simple filters is their extremely low computation time. As a result, the pedestrian detector is very efficient. It takes about 0.25 seconds to detect all pedestrians in a 360 x 240 pixel image on a 2.8 GHz P4 processor. About 0.1 seconds of that time is spent actually scanning the cascade over all positions and scales of the image and 0.15 seconds are spent creating the pyramids of difference images. Using optimized image processing routines we be-

lieve this can be further improved.

The idea of building efficient detectors that combine both motion and appearance cues will be applicable to other problems as well. Candidates include other types of human motion (running, jumping), facial expression classification, and possible lip reading.

References

- [1] S. Avidan. Support vector tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
- [2] F. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, volume 18(3), pages 207–212, 1984.
- [3] R. Cutler and L. Davis. Robust real-time periodic motion detection: Analysis and applications. In *IEEE Patt. Anal. Mach. Intell.*, volume 22, pages 781–796, 2000.
- [4] V. Philomin D. Gavrilu. Real-time object detection for "smart" vehicles. In *IEEE International Conference on Computer Vision*, pages 87–93, 1999.
- [5] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.
- [6] D. D. Hoffman and B. E. Flinchbaugh. The interpretation of biological motion. *Biological Cybernetics*, pages 195–204, 1982.
- [7] L. Lee. Gait dynamics for recognition and classification. Mit ai lab memo aim-2001-019, MIT, 2001.
- [8] F. Liu and R. Picard. Finding periodicity in space and time. In *IEEE International Conference on Computer Vision*, pages 376–383, 1998.
- [9] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, 1998.
- [10] R. Polana and R. Nelson. Detecting activities. *Journal of Visual Communication and Image Representation*, June 1994.
- [11] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 22–38, 1998.
- [12] R. Schapire and Y. Singer. Improving boosting algorithms using confidence-rated predictions, 1999.
- [13] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *International Conference on Computer Vision*, 2000.
- [14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

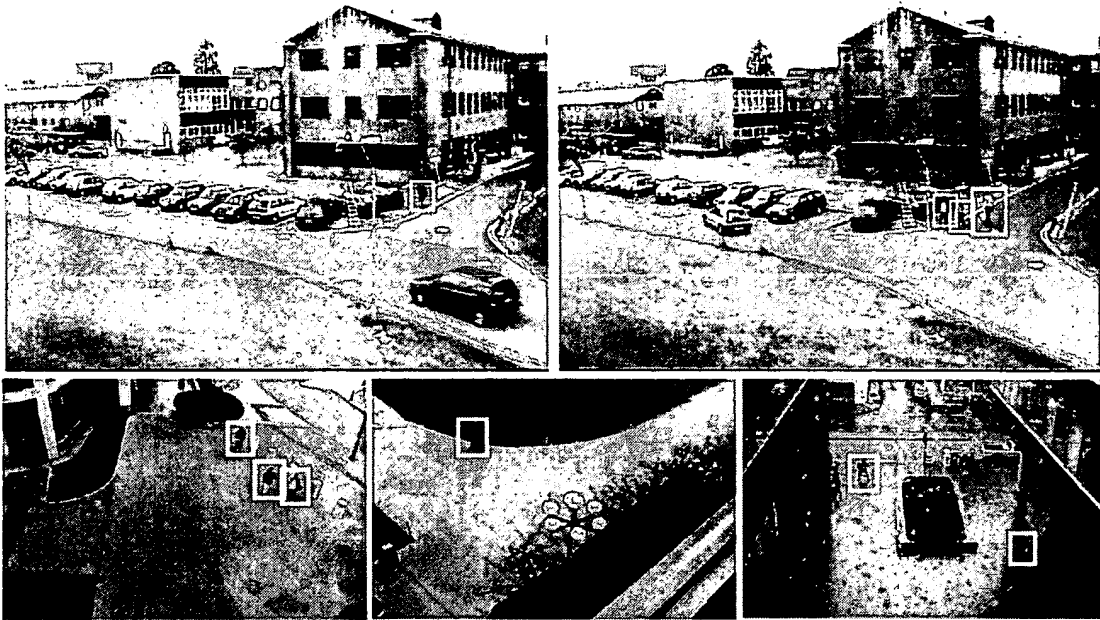


Figure 10: Example detections for the dynamic detector. Note the rain and snow falling in the image on the lower right.

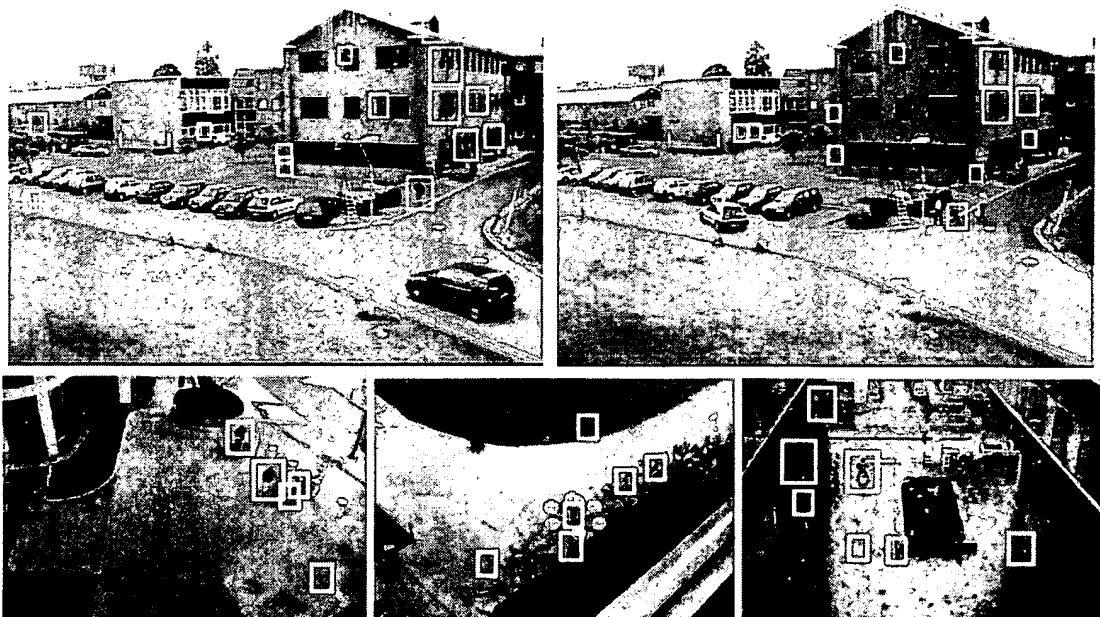


Figure 11: Example detections for the static detector.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.



052305

17589

U.S.

PTO

Express Mail Label No.: EV336791321US

Date of Deposit: 23.May.2005

Mail Stop PATENT APPLICATION
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

PATENT

Case Docket No.: 2004P09036 US01

Date: 23.May.2005

113014 U.S. PTO
11/135210

052305

Sir:

Transmitted herewith for filing is the patent application of:

Inventor(s):

Dorin Comaniciu

Leonid Sigal

Ying Zhu

For: GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

This application includes:

25 pages of specifications, including 5 claims and 1 abstract
1 title page
5 sheets of drawings, formal/ X informal containing 5 Figures
 photographs (design patent application only)

Also enclosed is:

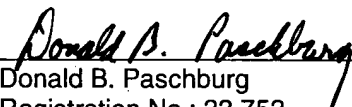
 Declaration and Power of Attorney
 An assignment of the invention to Siemens Corporate Research Inc..
 A certified copy of a application.
X Information Disclosure Statement pursuant to 37 CFR 1.56.

The filing fee has been calculated as shown below:

FOR:	(Col. 1) NO. FILED	(Col. 2) NO. EXTRA	OTHER THAN A SMALL ENTITY RATE FEE
BASIC FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 300.00
SEARCH FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 500.00
EXAMINATION FEE	XXXXXXXXXX	XXXXXXXXXX	\$ 200.00
TOTAL CLAIMS	16 - 20 =	0 x \$ 50.00 =	\$.00
INDEP CLAIMS	2 - 3 =	0 x \$ 200.00 =	\$.00
* MULTIPLE DEPENDENT CLAIM PRESENTED +\$ 360.00 =			\$.00
TOTAL			\$ 1000.00

- ☒ Please charge my Deposit Account No. 19-2179 in the amount of \$ 1000.00 duplicate copy of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 19-2179. A duplicate copy of this sheet is enclosed.
- ☒ Any additional filing fees required under 37 CFR 1.16.
- ☒ Any patent application processing fees under 37 CFR 1.17.
- ☒ The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 19-2179. A duplicate copy of this sheet is enclosed.
- ☒ Any patent application processing fees under 37 CFR 1.17.
- ☒ Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Dated: 23.May.2005
Siemens Corporation
Customer No. 28524
Tel. +1 (732) 321 3191


Donald B. Paschburg
Registration No.: 33,753

VR



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/135,210	05/23/2005	Dorin Comaniciu	2004P09036 US01

28524
 SIEMENS CORPORATION
 INTELLECTUAL PROPERTY DEPARTMENT
 170 WOOD AVENUE SOUTH
 ISELIN, NJ 08830

CONFIRMATION NO. 3651
 FORMALITIES LETTER



OC000000016327745

Date Mailed: 06/20/2005

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is missing. *A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.*
Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(f) of \$130 for a non-small entity, must be submitted with the missing items identified in this letter.

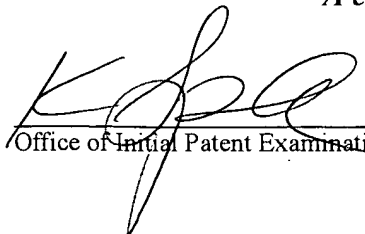
SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **\$130** for a Large Entity

- **\$130** Late oath or declaration Surcharge.

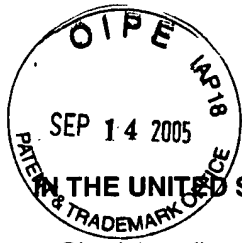
Replies should be mailed to: Mail Stop Missing Parts
 Commissioner for Patents
 P.O. Box 1450
 Alexandria VA 22313-1450

A copy of this notice MUST be returned with the reply.



Office of Initial Patent Examination (703) 308-1202

PART 3 - OFFICE COPY



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Sigal (et al)

Serial No.: 11/135,210

Filed: 05.23.05

Title: GRAPHICAL OBJECT MODELS FOR DETECTION . . . Date: 12.Sep.2005

Docket No.: 2004P09036US01

Examiner:

Group Art Unit: 2621

CERTIFICATE OF MAILING

I hereby certify that correspondence is being deposited on 12.Sep.2005 with the U.S. Postal Service with sufficient postage as first class mail on the below-indicated date in an envelope addressed to: Assistant Commissioner for Patents, Alexandria, VA 22313-1450

Donald B. Paschburg
Donald B. Paschburg, Reg. No. 33,753

Attention: Application Branch
Manager

RESPONSE TO NOTICE TO FILE MISSING PARTS OF APPLICATION --
FILING DATE GRANTED

Mail Stop Missing Parts
Commissioner for Patents
P.O.Box 1450
Alexandria, VA 22313-1450

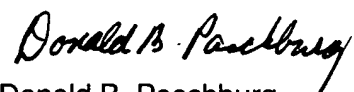
Sir:

This is a response to the Notice to File Missing Parts of Application having a mailing date of 06/20/2005 and for which a period for response of two (2) months from the mailing date of said Notice expired 08/20/2005.

Enclosed are 2 Declarations by the present Inventors. Also enclosed is a Petition and Fee for Extension of Time and a copy of the Notice to File Missing Parts of Application.

Please charge Deposit Account No. 19-2179 for the surcharge of \$ 130.00 in accordance with 37 CFR 1.16(f). A copy of this Response is enclosed. Issuance of a formal Filing Receipt is respectfully solicited.

Respectfully submitted,

A handwritten signature in black ink that reads "Donald B. Paschburg". The signature is written in a cursive style with a prominent, sweeping flourish at the end of the last name.

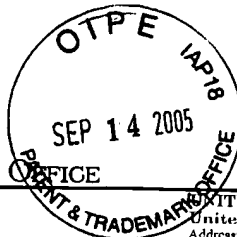
Donald B. Paschburg
Reg. No.: 33,753

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, NJ 08830
Tel.: +1 (732) 321 3191

VR



UNITED STATES PATENT AND TRADEMARK OFFICE

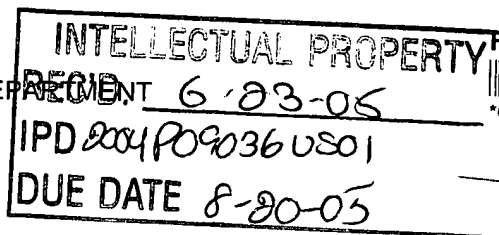


UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/135,210	05/23/2005	Dorin Comaniciu	2004P09036 US01

28524

SIEMENS CORPORATION
 INTELLECTUAL PROPERTY DEPARTMENT
 170 WOOD AVENUE SOUTH
 ISELIN, NJ 08830



CONFIRMATION NO. 3651



Date Mailed: 06/20/2005

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The oath or declaration is missing. *A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.*
Note: If a petition under 37 CFR 1.47 is being filed, an oath or declaration in compliance with 37 CFR 1.63 signed by all available joint inventors, or if no inventor is available by a party with sufficient proprietary interest, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(f) of \$130 for a non-small entity, must be submitted with the missing items identified in this letter.

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is \$130 for a Large Entity

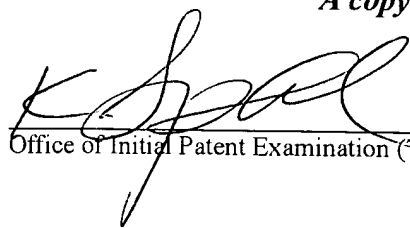
- \$130 Late oath or declaration Surcharge.

Replies should be mailed to: Mail Stop Missing Parts
 Commissioner for Patents
 P.O. Box 1450
 Alexandria VA 22313-1450

09/15/2005 JBALINAN 00000049 192179 11135210

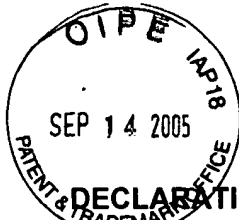
01 FC:1051 130.00 DA

*A copy of this notice **MUST** be returned with the reply.*

A handwritten signature in black ink, appearing to be 'K. Spall', is written over a horizontal line.

Office of Initial Patent Examination (703) 308-1202

PART 1 - ATTORNEY/APPLICANT COPY



**DECLARATION FOR PATENT APPLICATION, POWER OF
ATTORNEY & DESIGNATION OF CORRESPONDENCE ADDRESS**

As below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

the specification of which (check one)

 is attached hereto, and

 X was filed on 05.23.05 as Application Serial No. 11/135,210.

I hereby state that I have reviewed and understand the contents of the above-identified specification including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with 37 CFR §1.56(a).

**PRIOR FOREIGN/PCT APPLICATION(S) FILED WITHIN 12 MONTHS (6 MONTHS FOR
DESIGN) PRIOR TO THIS APPLICATION AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. §
119(a)-(d)**

I hereby claim foreign priority benefits under 35 USC §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Filing Date	Yes	No

**CLAIM FOR BENEFIT OF PRIOR U.S. PROVISIONAL APPLICATION(S)
(35 U.S.C. § 119(e))**

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below:

Provisional Application Number	Filing Date
60/574,799	05.27.04

**CLAIM FOR BENEFIT OF EARLIER US/PCT APPLICATION(S)
UNDER 35 U.S.C. 120**

I hereby claim the benefits under 35 USC §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of 35 USC §112, I acknowledge the duty to disclose material information as defined in 37 CFR §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Appl. No.	Filing Date	Patented, Pending, Abandoned

Power of Attorney:

I hereby appoint practitioners associated with customer number **28524** to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

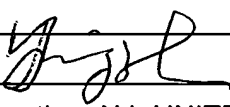
Send Correspondence to:

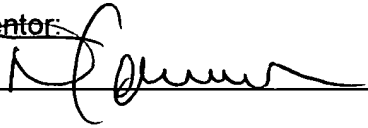
Customer No. 28524
Siemens Corporation

Direct inquiries to:
Elsa Keller, Legal Administrator
Tel. 732-321-3026
Fax 732-321-3014

I hereby declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under 18 USC §1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full Name of First Inventor: Leonid Sigal	
Inventor's Signature	Date:
Residence: Providence, RI, UNITED STATES OF AMERICA	
Citizenship: US	
Post Office Address: 115 Waterman Street, 4th Floor, Providence, RI 02912, UNITED STATES OF AMERICA	

Full Name of Second Inventor: Ying Zhu	
Inventor's Signature 	Date: 9/9/05
Residence: Monmouth Junction, NJ, UNITED STATES OF AMERICA	
Citizenship: CN	
Post Office Address: 88 Kingsland Circle, Monmouth Junction, NJ 08852, UNITED STATES OF AMERICA	

Full Name of Third Inventor: Dorin Comaniciu 	
Inventor's Signature	Date: 9/9/05
Residence: Princeton Jct., NJ, UNITED STATES OF AMERICA	
Citizenship: RO	
Post Office Address: 2 Stuart In. West, Princeton Jct., NJ 08550, UNITED STATES OF AMERICA	



**DECLARATION FOR PATENT APPLICATION, POWER OF
ATTORNEY & DESIGNATION OF CORRESPONDENCE ADDRESS**

As below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

the specification of which (check one)

☐ is attached hereto, and

☒ was filed on 05.23.05 as Application Serial No. 11/135,210.

I hereby state that I have reviewed and understand the contents of the above-identified specification including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with 37 CFR §1.56(a).

**PRIOR FOREIGN/PCT APPLICATION(S) FILED WITHIN 12 MONTHS (6 MONTHS FOR
DESIGN) PRIOR TO THIS APPLICATION AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. §
119(a)-(d)**

I hereby claim foreign priority benefits under 35 USC §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Filing Date	Yes	No

CLAIM FOR BENEFIT OF PRIOR U.S. PROVISIONAL APPLICATION(S)
(35 U.S.C. § 119(e))

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below:

Provisional Application Number	Filing Date
60/574,799	05.27.04

CLAIM FOR BENEFIT OF EARLIER US/PCT APPLICATION(S)
UNDER 35 U.S.C. 120

I hereby claim the benefits under 35 USC §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of 35 USC §112, I acknowledge the duty to disclose material information as defined in 37 CFR §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Appl. No.	Filing Date	Patented, Pending, Abandoned

Power of Attorney:

I hereby appoint practitioners associated with customer number **28524** to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Send Correspondence to:

Customer No. 28524
Siemens Corporation

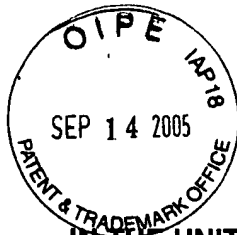
Direct inquiries to:
Elsa Keller, Legal Administrator
Tel. 732-321-3026
Fax 732-321-3014

I hereby declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under 18 USC §1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full Name of First Inventor: Leonid Sigal	
Inventor's Signature <i>Leonid Sigal</i>	Date: <i>7/25/2005</i>
Residence: Providence, RI, UNITED STATES OF AMERICA	
Citizenship: US	
Post Office Address: 115 Waterman Street, 4th Floor, Providence, RI 02912, UNITED STATES OF AMERICA	

Full Name of Second Inventor: Ying Zhu	
Inventor's Signature	Date:
Residence: Monmouth Junction, NJ, UNITED STATES OF AMERICA	
Citizenship: CN	
Post Office Address: 88 Kingsland Circle, Monmouth Junction, NJ 08852, UNITED STATES OF AMERICA	

Full Name of Third Inventor: Dorin Comaniciu	
Inventor's Signature	Date:
Residence: Princeton Jct., NJ, UNITED STATES OF AMERICA	
Citizenship: RO	
Post Office Address: 2 Stuart In. West, Princeton Jct., NJ 08550, UNITED STATES OF AMERICA	



PATENT

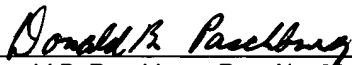
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Comaniciu (et al)
Serial No.: 11/135,210
Filed: 05.23.05
Title: GRAPHICAL OBJECT MODELS FOR DETECTION
AND TRACKING

Docket No.: 2004P09036US01
Examiner: unassigned
Group Art Unit: 2621
Date: 12.Sep.2005

CERTIFICATE OF MAILING

I hereby certify that correspondence is being deposited on
12.Sep.2005 with the U.S. Postal Service with sufficient postage as
first class mail on the below-indicated date in an envelope addressed
to: Assistant Commissioner for Patents, Alexandria, VA 22313-1450


Donald B. Paschburg, Reg. No. 33,753

Mail Stop AF
Assistant Commissioner for Patents
P.O.Box 1450
Alexandria, VA 22313-1450

PETITION AND FEE FOR EXTENSION OF TIME
SUBMITTED PURSUANT TO 37 CFR 1.136(a)

Sir:

On 06/20/2005

X an Notice to File Missing Parts of Nonprovisional Application was issued in the above-entitled
case, setting a three month shortened statutory period for response.
— a Notice of Appeal was filed.

Enclosed herewith is:

X a response to the outstanding Office Action.
— an Appeal Brief.
— a Continued Prosecution Application of the above-referenced application.

This is a petition for a one (1) month extension of time pursuant to 37 CFR 1.136(a). The
required fee should be charged to Deposit Account No. 19-2179, and a copy of this Petition is
enclosed for accounting purposes.

2038 / V: 03.05.20 / B: Val
10/15/2005 JBALINAN 00000049 192179 11135210
FC:1251 120.00 DA

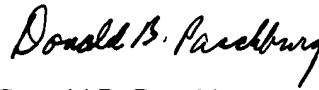
The fee is believed to be

\$ 120.00 per 37 CFR 1.17(a)(2)
-\$ _____ paid with previous petition dated

\$ 120.00 Balance due PTO.

If this Petition is for an incorrect time period or if the above balance due is incorrect, please treat this paper as a petition for the appropriate period and fee and charge/credit any fee deficiencies/overpayments to Deposit Account No. 19-2179.

Respectfully submitted,

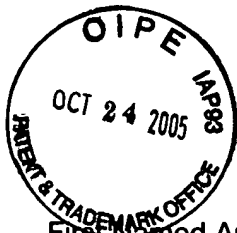


Donald B. Paschburg
Reg. No.: 33,753

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, NJ 08830
Tel.No.: +1 (732) 321 3191

Enclosure:

VR



1FW
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Sigal

Serial No.: 11/135,210

Filed: May 23, 2005

Title: GRAPHICAL OBJECT MODELS FOR DETECTION
AND TRACKING

Docket No.: 2004P09036 US01

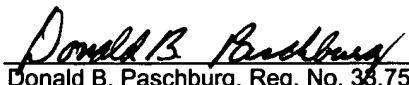
Examiner: Not yet assigned

Group Art Unit: 2621

Date: October 20, 2005

CERTIFICATE OF MAILING

I hereby certify that correspondence is being deposited on October 20, 2005 with the U.S. Postal Service with sufficient postage as first class mail on the below-indicated date in an envelope addressed to:
Mail Stop DD, Commissioner for Patents, PO Box 1450, Alexandria,
VA 22313-1450


Donald B. Paschburg, Reg. No. 33,753

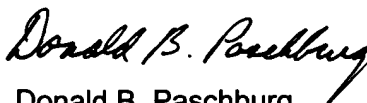
Mail Stop DD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Sir:

The documents listed on the accompanying Form PTO/SB/08a, and of which copies are enclosed, are hereby cited pursuant to 37 CFR §1.56, §1.97 and §1.98, for consideration in the examination of the above-identified application and for the purpose of having them made of record. These documents were cited in an International Search Report (copy enclosed) in a related application.

Respectfully submitted,


Donald B. Paschburg
Reg. No.: 33,753

Siemens Corporation
Intellectual Property Department
186 Wood Avenue South
Iselin, NJ 08830
(732) 321-3191

Substitute for form 1449A/PTO

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**

(use as many sheets as necessary)

of

Complete if Known

Application Number	11/135,210
Filing Date	May 23, 2005
First Named Inventor	Leonid Sigal
Group Art Unit	2621
Examiner Name	Not yet assigned
Attorney Docket Number	2004P09036 US01

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials *	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²
	3	Lan et al., "A unified spatio-temporal articulated model for tracking", Computer Vision and Pattern Recognition, 2004, Proceedings of the 2004 IEEE Computer Society Conference on Washington, DC, 27 June – 2 July 2004, Piscataway, NJ, Vol. 1, pp. 722-729.	
	4	Donohoe, "Combining Segmentation and Tracking for the Classification of Moving Objects in Video Scenes", Conference Record, 22 nd Asilomar Conference on Signals, Systems and Computers (Cat. No. 88CH2835-7), Maple Press San Jose, CA, USA, Vol. 2, 31 October 1988, pp. 533-538.	

Examiner
SignatureDate
Considered

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 120 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 217 572 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:

26.06.2002 Bulletin 2002/26

(51) Int Cl.7: G06K 9/00

(21) Application number: 01204728.8

(22) Date of filing: 06.12.2001

(84) Designated Contracting States:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR

Designated Extension States:

AL LT LV MK RO SI

(72) Inventors:

- Chen, Shoupu
Rochester, New York 14650-2201 (US)
- Ray, Lawrence A.
Rochester, New York 14650-2201 (US)

(30) Priority: 19.12.2000 US 740422

(71) Applicant: EASTMAN KODAK COMPANY
Rochester, New York 14650 (US)

(74) Representative:

Nunney, Ronald Frederick Adolphe et al
Kodak Limited,
Patents, W92-3A,
Headstone Drive
Harrow, Middlesex HA1 4TY (GB)

(54) Digital image processing method and computer program product for detecting human irises in an image

(57) A digital image processing method is taught for detecting human irises in a digital image. The method comprises the steps measuring the red intensity of the pixels in the image, determining the probability that each pixel is an iris based upon the red intensity of the pixel, determining the probability that each pixel is not an iris based upon the red intensity of the pixel; and determining whether the pixel is an iris by analyzing the relationship between the probability that the pixel is an iris and the probability that the pixel is not an iris. In one embodiment of the present invention, the determination as to whether a pixel is an iris pixel is then made based upon

the application of a Bayes model to the probability that the pixel is not an iris, the probability of the occurrence of an iris in the identified region and probability of the occurrence of a non-iris pixel in the identified region. In another embodiment of the present invention, the method comprises the steps of finding an oval shaped skin color region, detecting iris color pixels in the oval shaped skin color region, detecting iris color pixels in the oval shaped region using a Bayes model and locating eye positions based upon the detected iris color pixels.

A computer program product for performing these methods is also taught.

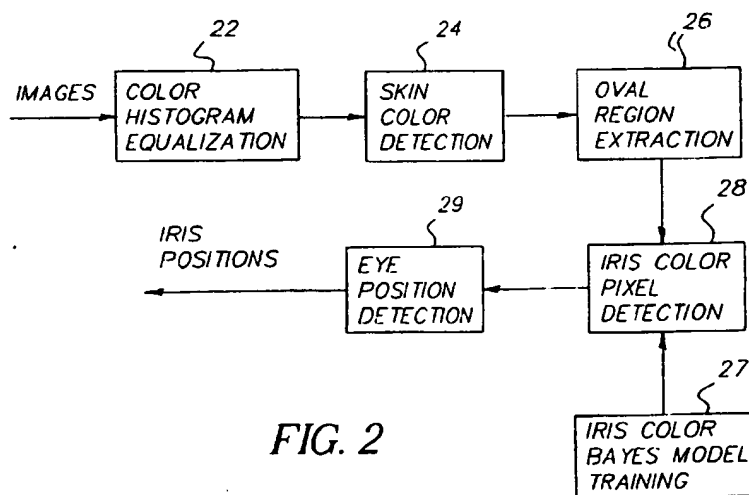


FIG. 2

Description

[0001] The present invention relates to digital image processing methods for detecting facial features and more particularly to methods of detecting human irises in a digital image.

[0002] In digital image processing it is often useful to detect human irises in an image. This information is used, for example, to locate other features in the image such as human eyes. This information can also be used for other purposes such as finding the orientation of a human face in the image.

[0003] Methods are known in the art for detecting human eyes. For example, U.S. Patent 6,072,892 discloses the use of a thresholding method to detect the position of human eyes in a digital image. In this method, a scanning window scans across the entire image using a raster scanning method. A histogram extractor extracts intensity histograms from the window as it scans across the image. Each intensity histogram is examined by a peak detector to find three peaks in the histogram representing the skin, the white of the eye, and the black of the pupil located within the iris. A histogram having the three peaks identifies a location in an image that potentially defines an eye position. Eye position is determined from among the potential locations by calculating the area under the histogram associated with each potential location and by selecting the location associated with the histogram having the largest area.

[0004] One of the problems with this approach is that the entire image must be scanned on a pixel-by-pixel basis. Thus, a search window must be positioned at each pixel and a histogram must be assembled at each pixel location. Further, the area under each histogram must be calculated and stored. It will be appreciated that this method consumes enormous amounts of computing power and reduces the rate at which images can be processed. This method can also produce a high rate of false positives.

[0005] Methods are also known to detect eyes that have abnormally high red content. Such abnormal red content is commonly associated with a photographic phenomenon known as red eye. Red eye is typically caused by a flash of light that is reflected by a pupil. As is described in commonly assigned and co-pending U.S. Patent application No. 08/919,560, it is known to search for images for pixels having the high red content that is indicative of red eye. Similarly, commonly assigned U.S. Patent 5,432,863 describes a user interactive method for detecting pixels in an image that have color characteristic of red eye. It will be recognized that these methods detect eyes only where red eye is present.

[0006] Thus, there is a need for a method of locating human irises in a digital image that is accurate and efficient.

[0007] The need is met according to the present invention by a digital image processing method for detect-

ing human irises in a digital image. This method comprises the steps of: measuring the red intensity of the pixels in the image, determining the probability that each pixel is an iris based upon the red intensity of the pixel, determining the probability that each pixel is not an iris based upon the red intensity of the pixel; and determining whether the pixel is an iris by analyzing the relationship between the probability that the pixel is an iris and the probability that the pixel is not an iris. In one embodiment of the present invention, the determination as to whether a pixel is an iris pixel is made based upon the application of a Bayes model to the probability that the pixel is an iris, to the probability that the pixel is not an iris, the probability of the occurrence of an iris in the image and probability of the occurrence of a non-iris pixel in the image.

[0008] In another embodiment of the present invention, skin color regions are detected in the image and only the pixels within the skin color regions are measured for red intensity. The determination as to whether a pixel is an iris pixel is made based upon the application of a Bayes model to the probability that the pixel is an iris, the probability that the pixel is not an iris, the probability of the occurrence of an iris in the skin color region and the probability of a non-iris in the skin color region.

[0009] The need is also met according to the present invention by a method for detecting irises in a digital image comprising the steps of finding an oval shaped skin color region, detecting iris color pixels in the oval shaped region using a Bayes iris model and locating eye positions based upon the detected iris color pixels.

[0010] The need is also met according to the present invention by a computer program product for detecting irises in a digital image comprising: a computer readable storage medium having a computer program stored thereon for performing the method of the present invention.

[0011] The present invention is effective to obtain iris positions in a frontal face image and has the advantage of reducing the region of the image that must be searched, thereby greatly reducing the computation required to locate an eye, and reducing the incidence of eye detection false positives.

Figure 1 is a schematic diagram of an image processing system useful in practicing the present invention;

Figure 2 is a flowchart illustrating the eye detection method of the present invention;

Figure 3 is an illustration showing the relationship between certain geometric parameters and an oval shaped skin colored region in an image;

Figure 4 is an illustration showing, the conditional probability that a given pixel is an iris pixel stated as a function of a specific red intensity and the conditional probability that a given pixel is a non-iris pixel as a function of a specific red intensity I.

Figure 4a is a flowchart presenting the process of

developing a statistical model representing the conditional probability that a given pixel is an iris pixel as a function of a specific red intensity level and the a statistical model representing the conditional probability that a given pixel is an iris pixel as a function of a specific red intensity level.

Figure 5 is a flowchart illustrating the process of eye position detection after detecting the iris color pixel clusters;

Figure 6 is an illustration showing the iris color pixel clusters;

Figure 7 is a flowchart illustrating the details of the process of applying the method of summation of squared difference to eye position detection; and

Figure 8 shows an eye template and a search window centered at the center of a cluster.

[0012] Fig. 1, shows an image processing system useful in practicing the present invention including a color digital image source 10, such as a film scanner, digital camera, or digital image storage device such as a compact disk drive with a Picture CD. The digital image from the digital image source 10 is provided to an image processor 12, such as a programmable personal computer, or digital image processing work station such as a Sun Sparc workstation. The image processor 12 may be connected to a CRT display 14, an operator interface such as a keyboard 16 and a mouse 18. Image processor 12 is also connected to computer readable storage medium 17. The image processor 12 transmits processed digital images to an output device 19. Output device 19 can comprise a hard copy printer, a long-term image storage device, a connection to another processor, or an image telecommunication device connected for example to the internet.

[0013] In the following description, a preferred embodiment of the present invention will be described as a method. However, in another preferred embodiment, the present invention comprises a computer program product for detecting human eyes and irises in a digital image in accordance with the method described. In describing the present invention, it should be apparent that the computer program of the present invention can be utilized by any well-known computer system, such as the personal computer of the type shown in Fig. 1. However, many other types of computer systems can be used to execute the computer program of the present invention. Consequently, the computer system will not be discussed in further detail herein.

[0014] It will be understood that the computer program product of the present invention may make use of image manipulation algorithms and processes that are well known. Accordingly, the present description will be directed in particular to those algorithms and processes forming part of, or cooperating more directly with, the method of the present invention. Thus, it will be understood that the computer program product embodiment of the present invention may embody algorithms and

processes not specifically shown or described herein that are useful for implementation. Such algorithms and processes are conventional and within the ordinary skill in such arts.

[0015] Other aspects of such algorithms and systems, and hardware and/or software for producing and otherwise processing the images involved or cooperating with the computer program product of the present invention, are not specifically shown or described herein and may be selected from such algorithms, systems, hardware, components and elements known in the art.

[0016] The computer program for performing the method of the present invention may be stored in a computer readable storage medium. This medium may comprise, for example; magnetic storage media such as a magnetic disk (such as a hard drive or a floppy disk) or magnetic tape; optical storage media such as an optical disc, optical tape, or machine readable bar code; solid state electronic storage devices such as random access memory (RAM), or read only memory (ROM); or any other physical device or medium employed to store a computer program. The computer program for performing the method of the present invention may also be stored on computer readable storage medium that is connected to the image processor by way of the internet or other communication medium. Those skilled in the art will readily recognize that the equivalent of such a computer program product may also be constructed in hardware.

[0017] Turning now to Fig. 2, the method of the present invention will be described in greater detail. Fig. 2 is a flow chart illustrating one embodiment of the iris detection method of the present invention. In the embodiment shown in Fig. 2, iris detection is accomplished by first detecting skin colored regions in the image and then identifying iris pixels from the skin colored regions.

[0018] The first step in skin color detection is color histogram equalization shown in Fig. 2 as step 22. Color Histogram Equalization step 22 receives images to be processed and ensures that the images are in a form that will permit skin color detection. This step is made necessary because human skin may take on any number of colors in an image because of lighting conditions, flash settings or other circumstances. This makes it difficult to automatically detect skin in such images. In Color Histogram Equalization step 22, a statistical analysis of each image is performed. If the statistical analysis suggests that the image may contain regions of skin that have had their appearance modified by lighting conditions, flash settings or other circumstances then such images are modified so that skin colored regions can be detected.

[0019] After the Color Histogram Equalization step 22, the image is searched for skin color regions in Skin Color Detection step 24. While it is possible to detect skin in a digital image in a number of ways, a preferred method for detecting skin in a digital image is to separated skin color pixels from other pixels in an image by defining a working color space that contains a range of

possible skin colors collected from a large population of images. A pixel is then identified as skin color pixel if the pixel has a color that is within the working color space. [0020] Skin Color Detection step 24 identifies a region of skin color pixels in the image. This region can be defined in any number of ways. In one embodiment, the skin color region is defined by generating a set of pixel locations identifying the pixels in the image having skin colors. In another embodiment, a modified image is generated that contains only skin color pixels. In yet another embodiment, Skin Color Detection step 24 defines boundaries that confine the skin color region in the image. It will be recognized that more than one skin color region can be identified in the image.

[0021] Oval Region Extraction step 26 examines the skin color regions detected by the Skin Color Detection step 24 to locate skin color regions that may be indicative of a face. Because the human face has a roughly oval shape, the skin color regions are examined to locate an oval shaped skin color region. When an oval shaped skin color region is found, the Oval Region Extraction step 26 measures the geometric properties of the oval shaped skin color region. The Oval Region Extraction step 26 uses these measurements to define parameters that describe the size of the face and the location of the face within the image.

[0022] Fig. 3 is an illustration of the relationship between the geometric parameters used to define an oval shaped skin color region in the image. As is shown in Fig. 3, these parameters include Oval_top 300, Oval_bottom 302, Oval_left 304, Oval_right 306, Oval_center_row 308, and Oval_center_column 310. These parameters are can be used in subsequent processing of the image. It will be recognized that the method of the present invention can be practiced using skin color detection regions that have shapes that are other than oval and that other geometric parameters can be defined in association with such shapes. It will also be recognized that it is not necessary to detect an oval or other shaped area in the image. In such case, the skin color region is examined to detect iris color pixels. Further, in such a case, still other parameters describing the skin color region are defined for use in the eye detection process.

[0023] After the oval region extraction has been performed, Iris Color Pixel Detection step 28 examines the pixels in the oval shaped skin color region to detect iris color pixels. In the method of the present invention, Iris Color Pixel Detection step 28 determines whether a pixel is an iris by measuring the red intensity of the pixel. Red intensity levels are measured because it has been observed that that a human iris has a low red intensity level as compared to human skin which has a relatively high red intensity level. However, the method of the present invention does not use a red level thresholding method to determine whether a pixel is to be classified as an iris or as a non-iris.

[0024] Instead, the present invention classifies a pixel

as an iris or a non-iris pixel on the basis of a probability analysis. This probability analysis applies an iris statistical model and a non-iris statistical model. The iris statistical model defines the probability that a given pixel is an iris pixel based upon the red intensity level of the pixel. Similarly, the non-iris statistical model defines the probability that a given pixel is not an iris pixel based upon the red intensity level of the pixel. The relationship between these models is non-linear as is shown by way of example in Fig. 4 which is an illustration of the conditional probability 402 that a given pixel is an iris pixel stated as a function of a specific red intensity and the conditional probability 404 that a given pixel is a non-iris pixel as a function of a specific red intensity I.

[0025] The probability analysis can take many forms. For example, the probabilities can be combined in various ways with a pixel being classified as an iris or not on the basis of the relationship between these probabilities. However, in a preferred embodiment, a mathematical construct known as a Bayes model is used to combine the probabilities to produce the conditional probability that a pixel having a given red intensity belongs to an iris.

[0026] In this embodiment, the Bayes model is applied as follows:

$$P(Iris|I) = \frac{P(I|Iris)P(Iris)}{P(I|Iris)P(Iris) + P(I|noniris)P(noniris)}$$

where $P(Iris|I)$ is the conditional probability that a given pixel intensity belongs to an iris; $P(I|Iris)$ is the conditional probability that a given iris pixel has a specific intensity I; $P(Iris)$ is the probability of the occurrence of an iris in the face oval region; $P(I|noniris)$ is the conditional probability that a given non-iris pixel has a specific intensity I; and $P(noniris)$ is the probability of the occurrence of a non-iris pixel in the face oval region. The Bayes model further applies the probability of the occurrence of an iris in a face oval region and the probability of the occurrence of a non-iris pixel in the face oval region. Using a probability analysis based on the Bayes model, a pixel is classified as an iris if the conditional probability that a pixel having a given red intensity belongs to an iris is greater than, for example, 0.05.

[0027] In the embodiment described above, only those pixels in the oval shaped skin color region defined by Oval_top 300, Oval_bottom 302, Oval_left 304, and Oval_right 306 are examined. Confining the pixels to be examined to those in the oval shaped skin color region reduces number of pixels to be examined and decreases the likelihood that pixels that are not irises will be classified as such. It will be understood that confining the pixels to be examined to those in the skin color region, as is described in other embodiments of the present invention, conveys similar advantages. It will also be understood that it is not necessary to detect skin color regions and that the present invention can be practiced by measuring the red intensity of each pixel in the

image and determining whether a pixel is an iris based upon the probability analysis described above.

[0028] Fig. 4a shows a flow chart illustrating the processes used in the Iris Color/Bayes Model Training step 27 of Fig. 2 for developing the statistical models used to classify the pixels. This step will be performed before the method for detecting irises is used to detect iris pixels. As is shown, a large sample of frontal face images are collected and examined. All iris pixels and non-iris pixels in the face region of each image are then manually identified 40, 42. Next, the conditional probability that a given iris pixel has a specific red intensity I , $P(I|iris)$ is computed and the probability of the occurrence of an iris in the face oval region, $P(iris)$ 44 is computed; then the conditional probability that a given noniris pixel has a specific red intensity I , $P(I|noniris)$ is computed and finally the probability of the occurrence of a non-iris pixel in the face oval region, $P(noniris)$ 46 is computed. The computed statistical models of iris and non-iris are used in the Bayes formula to produce the conditional probability that a given pixel intensity belongs to an iris, $P(iris|I)$ 48. In application, the Bayes model can be used to generate a look-up table to be used in Iris Color Pixel Detection step 28.

[0029] The Iris Color Pixel Detection step 28 identifies the location of the iris color pixels in the image. In some cases, it will be desirable to ensure that the iris color pixels that are detected are associated with an eye. This is done by performing the step of eye detection. Eye Position Detection step 27 is used to estimate the eye positions. It will be appreciated that there are many ways to determine whether an iris pixel is associated with an eye in the image. In one preferred embodiment of the present invention the iris color pixel locations are used to facilitate the process of determining whether an iris pixel is associated with an eye in the image.

[0030] Fig. 5 shows a flow chart illustrating the process of eye position detection using the iris color pixels. As is shown in Fig. 5, the eye position detection process starts with an Iris Pixel Clustering step 50. If iris color pixels are detected, then the iris pixels must be assigned to a cluster. A cluster is a non-empty set of iris color pixels with the property that any pixel within the cluster is also within a predefined distance to another pixel in the cluster. One example of a predefined distance is one thirtieth of the digital image height. The Iris Pixel Clustering step 50 of Fig. 5 groups iris color pixels into clusters based upon this definition of a cluster. However, it will be understood that pixels may be clustered on the basis of other criteria.

[0031] Under certain circumstances, a cluster of pixels may not be valid. Accordingly, the optional step of validating the clusters is shown in Fig. 5 as Cluster Validation step 51. A cluster may be invalid because, if it contains too many iris color pixels or because the geometric relationship of the pixels in the cluster suggests that the cluster is not indicative of an iris. For example, if the ratio is greater than two, then this cluster is invalid.

Invalid iris pixel clusters are removed from further consideration by the method of the present invention. Accordingly, in the portions of the description that follow, valid iris pixel clusters will be referred to simply as iris pixel clusters.

[0032] After the clustering operation, a center for each of the clusters is calculated in Cluster Centering step 52. The center of a cluster is determined as the center of mass of the cluster. The center position of the clusters is calculated with respect to the origin of the image coordinate system. The origin of the image coordinate system for a digital image is defined as at the upper left corner of the image boundary.

[0033] Oval division step 54 uses the Oval_center_column 310 parameter to separate the oval shaped skin color region into a left-half region and a right-half region. As is shown in Fig. 6 iris pixel clusters 602 and the center positions 600 of the iris pixel clusters 602 are positioned in either the left-half or right-half regions 604 and 606 separated by the Oval_Center_Column 310.

[0034] To locate eyes in the image using the iris pixel clusters, a left-eye position search 56 is conducted in the left-half region using a method known as the Summation of the Squared Difference. A right-eye position search 58 is also conducted in the right-half region based on the same Summation of the Squared Difference method. In general, the summation of the squared difference method involves calculating the summation of the squared difference of the intensity values of the corresponding pixels in an eye template and a patch of the image that has the same size as the template. In this method, each pixel in the patch of pixels has a corresponding pixel in the template. The difference between the intensity level of each of the corresponding pixels is calculated. Each difference is then squared. The sum of each of the squared differences for each of the pixels in the set is then calculated. This summation of the squared differences provides a relative measure of the degree of correspondence between each of the pixel sets measured and the template. The eye template itself is generated by averaging a large number of sample eye images.

[0035] As applied in the present invention, summation of the squared difference values are calculated for each pixel in each window in each half-region. These values are compared and the pixel having the lowest relative summation of the squared difference value is identified as an eye location for the half-region. This process is performed separately on the clusters of the left and the right-half regions of the image in the manner described below.

[0036] It will be noted that while the present invention has been described as using the summation of the squared difference method to identify the best relative match between the average eye template and each of the pixels, other methods to determine the degree of relative correspondence can be used. In particular, the

mean-squared error method can be used in place of the summation of the squared difference method.

[0037] Referring to Figs. 7 and 8, the eye position search process 56 or 58 is started with centering 70 a window 800 at the center of each cluster 802 in a respective half-region. The operation of calculating the summation of the squared differences 72 is then performed, separately, using a patch of pixels centered on each of the pixels in each of windows 74. The position of the pixel having the lowest summation of squared difference value in each window 800 is recorded 76. When this process has been completed for every cluster center 802 in a half region 78, the position of the pixel having the lowest summation of squared difference value is recorded 79. This position is the eye position for the half-region.

[0038] It will be appreciated that the summation of the squared difference method of steps 56 and 58 of Fig. 5, can also be performed without the use of oval region extraction. In such an embodiment, the skin colored region can be divided into a left-half region and a right-half region. Iris pixel clusters can then be divided into left-half region and right-half region clusters. The summation of the squared difference method can then be applied.

[0039] The subject matter of the present invention relates to digital image understanding technology, which is understood to mean technology that digitally processes a digital image to recognize and thereby assign useful meaning to human understandable objects, attributes or conditions and then to utilize the results obtained in the further processing of the digital image.

[0040] In this manner, the present invention provides an efficient method for detecting normally appearing human irises in a digital image.

Claims

1. A digital image processing method for detecting human irises in a digital image formed from pixels, the method comprising the steps of:

measuring a red intensity of the pixels in the image;
determining a probability that each pixel is an iris based upon the red intensity of the pixel;
determining a probability that each pixel is not an iris based upon the red intensity of the pixel;
and
determining whether each pixel is an iris by analyzing the relationship between the probability that the pixel is an iris and probability that the pixel is not an iris.

2. The method of claim 1, wherein the step of determining whether each pixel is an iris is based upon the application of a Bayes model to the probability

that the pixel is an iris, the probability that a pixel is not an iris, the probability of the occurrence of a non-iris pixel in the image and the probability of the occurrence of a non-iris pixel in the image.

3. The method of claim 1 further comprising the step of detecting skin color regions in the image and wherein the step of measuring the red intensity comprises measuring the red intensity of only the pixels in the skin color region.
4. The method of claim 3 further comprising the step of performing color equalization on the image prior to the step of detecting skin color regions in the image.
5. The method of claim 4 wherein the step of determining whether each pixel is an iris is based upon the application of a Bayes model to the probability that the pixel is an iris, the probability that the pixel is not an iris, the probability of the occurrence of an iris in a skin color region, and the probability of the occurrence of a non-iris pixel in a skin color region.
6. The method of claim 4 further comprising the step of finding oval shaped skin colored regions in the image and wherein the step of measuring the red intensity of the pixels in the skin color region comprises measuring only those pixel within an oval shaped skin color region.
7. The method of claim 6 wherein the step of determining whether each pixel is an iris is based upon the application of a Bayes model to the probability that the pixel is an iris, the probability that the pixel is not an iris, the probability of the occurrence of an iris in an oval shaped skin color region, and the probability of the occurrence of a non-iris pixel in an oval shaped skin color region.
8. A computer program product for detecting human irises and eyes in a digital image, the computer program product comprising a computer readable storage medium having a computer program stored thereon for performing the steps of:
finding a skin colored region;
detecting iris color pixels in the skin colored region using a Bayes model; and
locating eye positions based upon the detected iris color pixels.
9. The computer program product of claim 8, wherein the step of locating eye positions based upon the detected iris color pixels comprises the steps of:
clustering iris color pixels;
finding the center of each cluster;

dividing the skin colored region into a left-half
and a right-half;
locating the most likely left eye position in the
left-half region using the summation of squared
difference method; and
locating the most likely right eye position in the
right-half region using the summation of
squared difference method.

5

10. The computer program product of claim 9, wherein
the step of locating the most likely left eye position
is performed using the summation of squared dif-
ference method comprises the steps of:

10

defining a window of pixels surrounding the
center of each of the clusters in the image;
dividing the iris pixel clusters into left-half iris
pixel clusters and right-half pixel clusters; and
locating the most likely left eye position based
on the summation of squared difference be-
tween an average eye and patch of the image
centered at each of the pixels in each of the win-
dows surrounding a left-half iris pixel cluster.

15

20

25

30

35

40

45

50

55

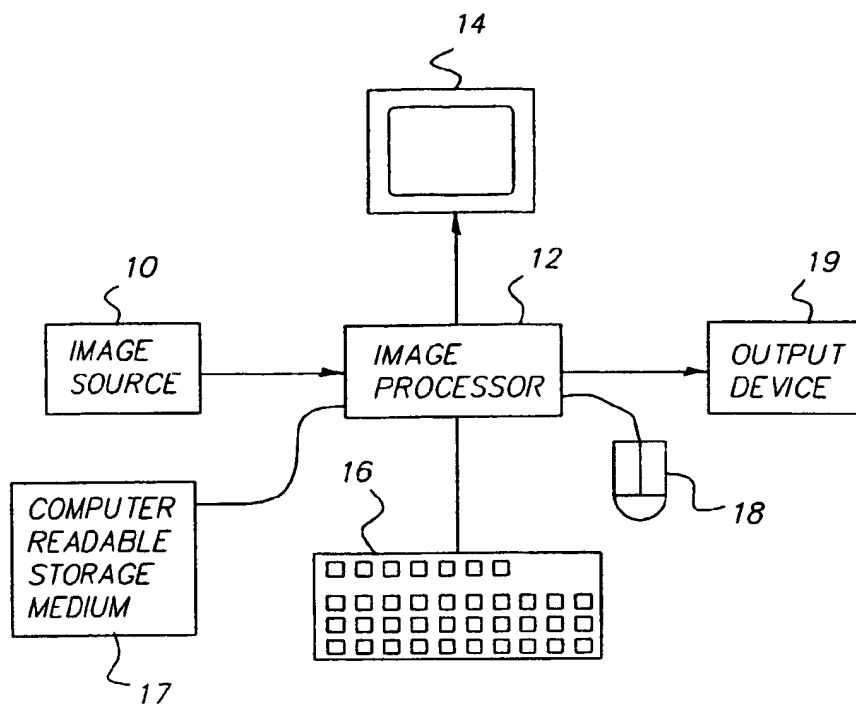


FIG. 1

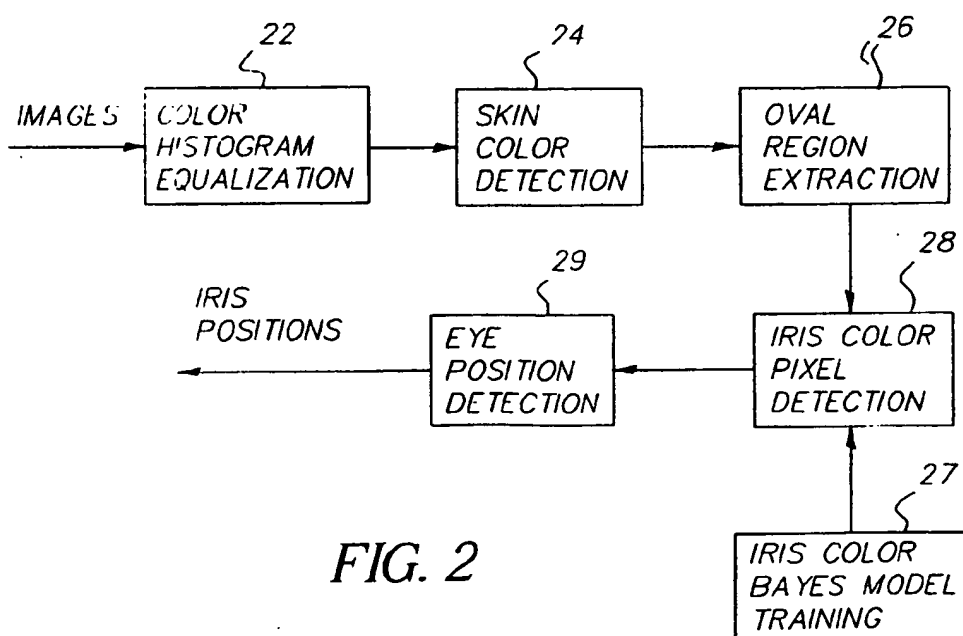


FIG. 2

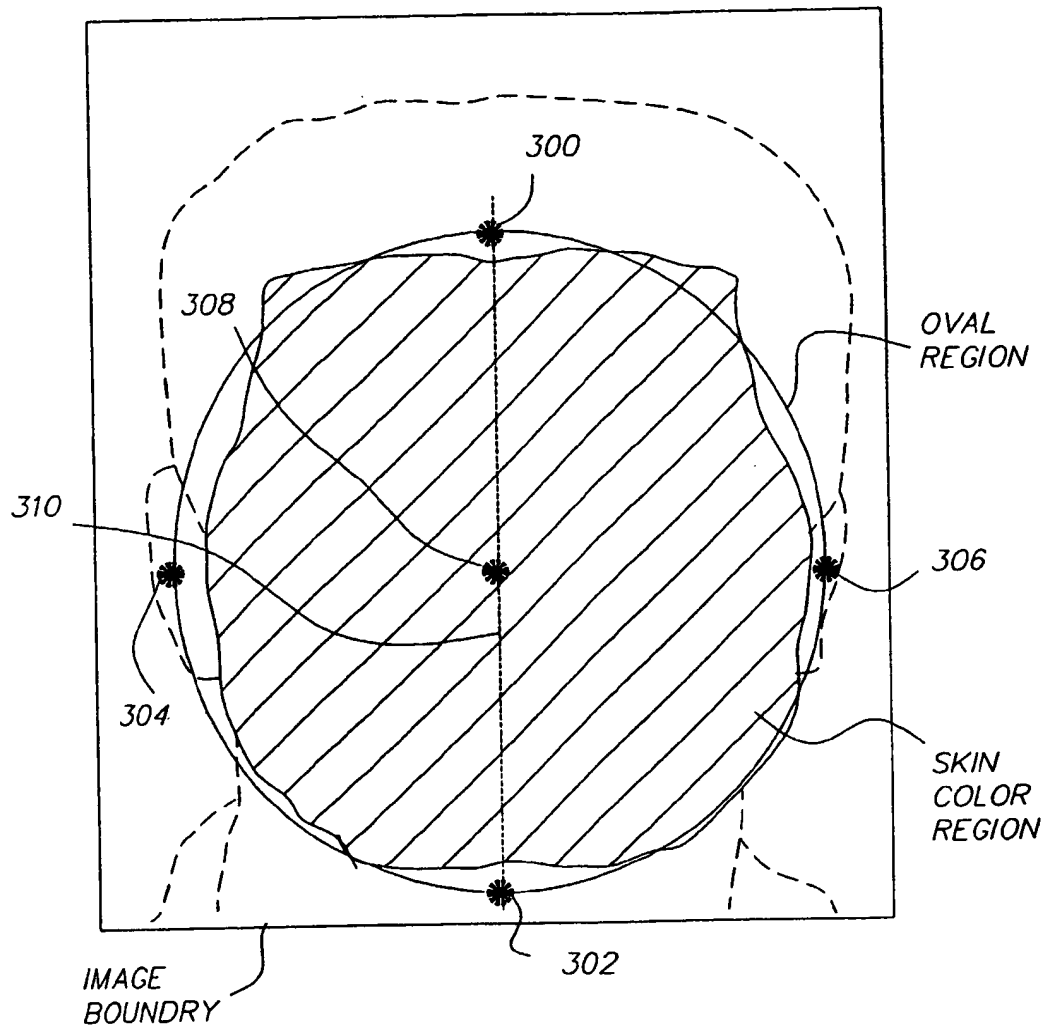


FIG. 3

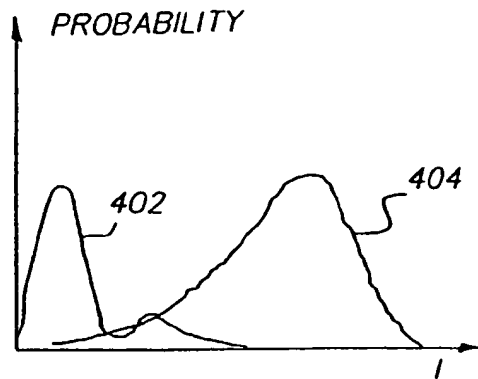


FIG. 4A

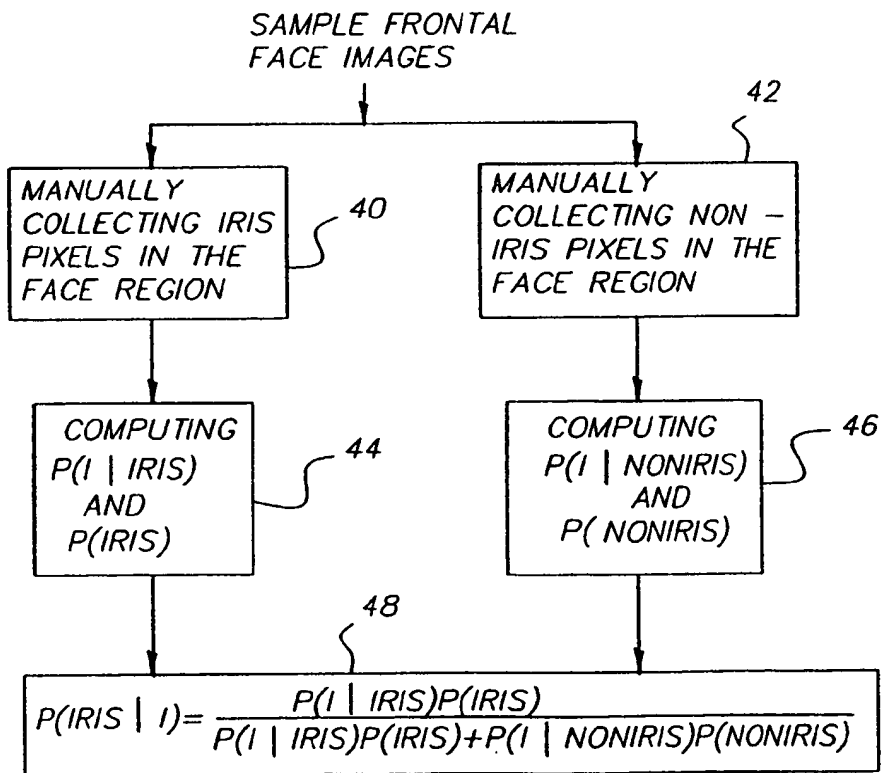


FIG. 4B

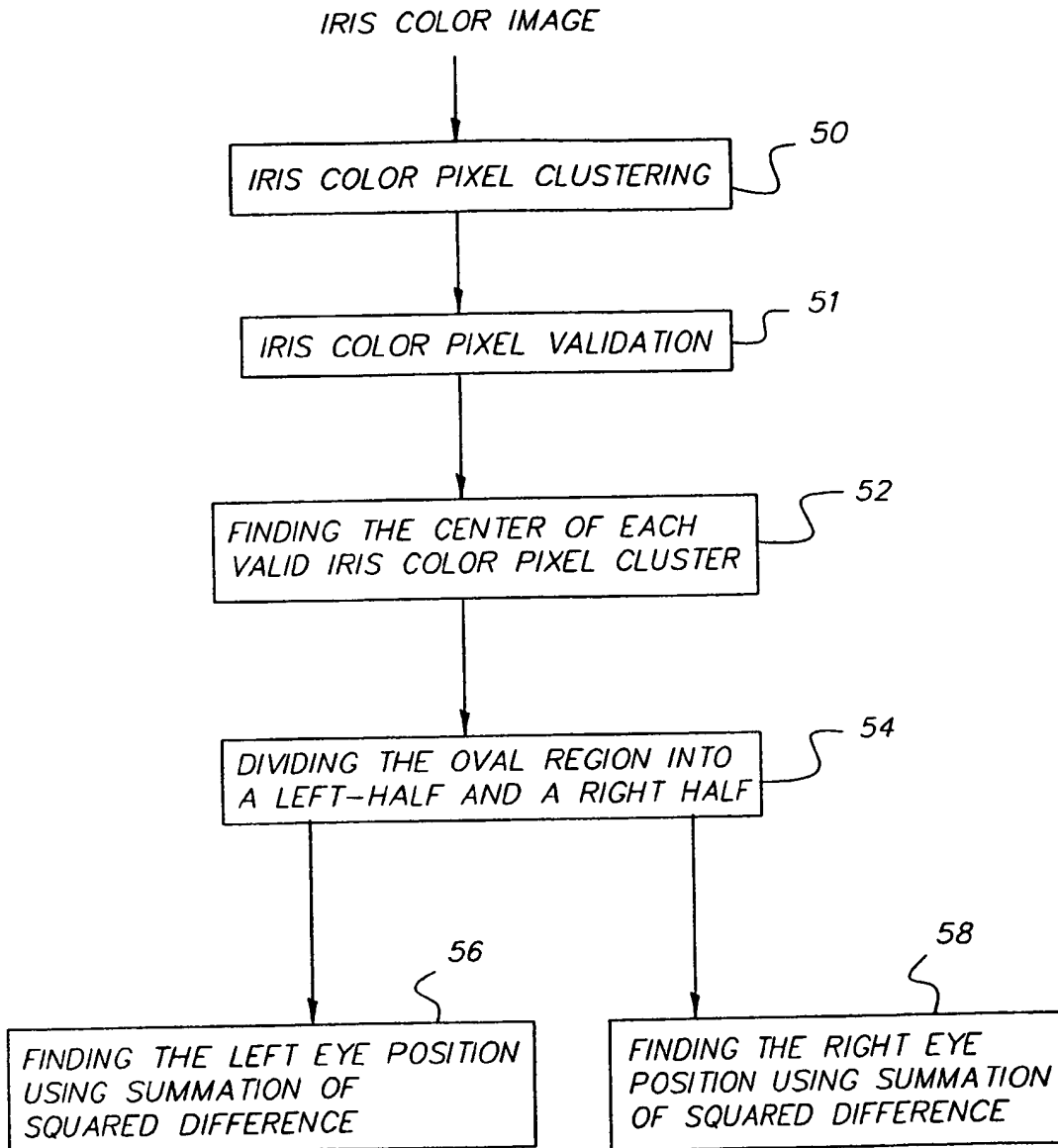


FIG. 5

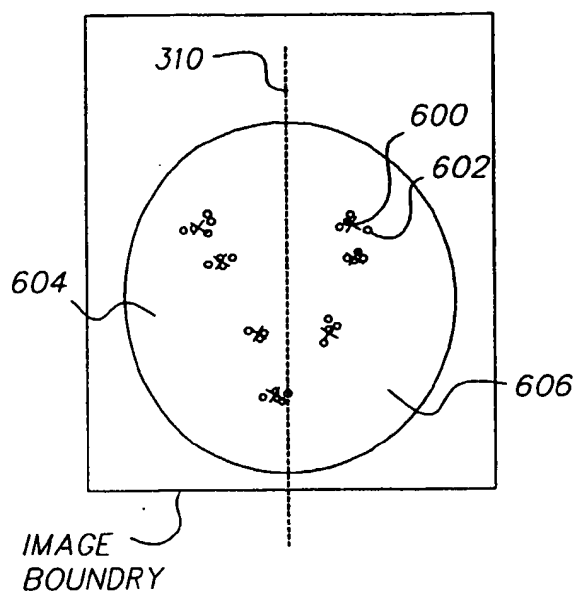


FIG. 6

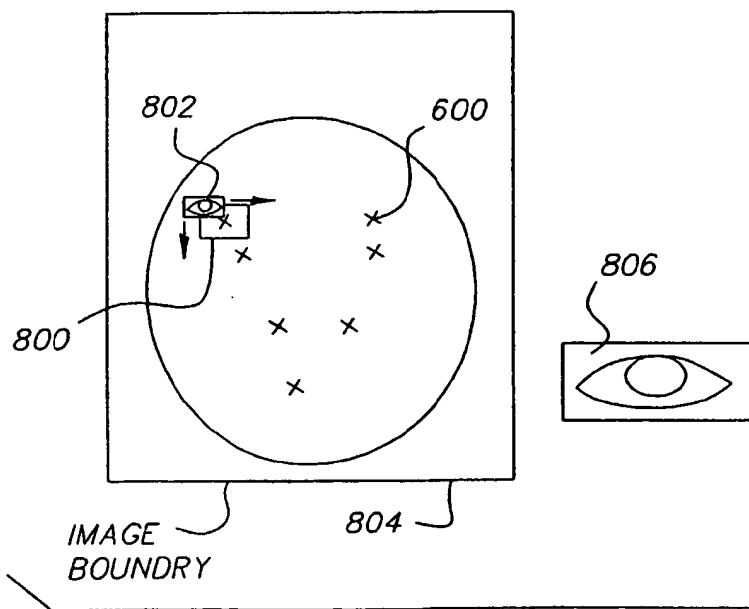
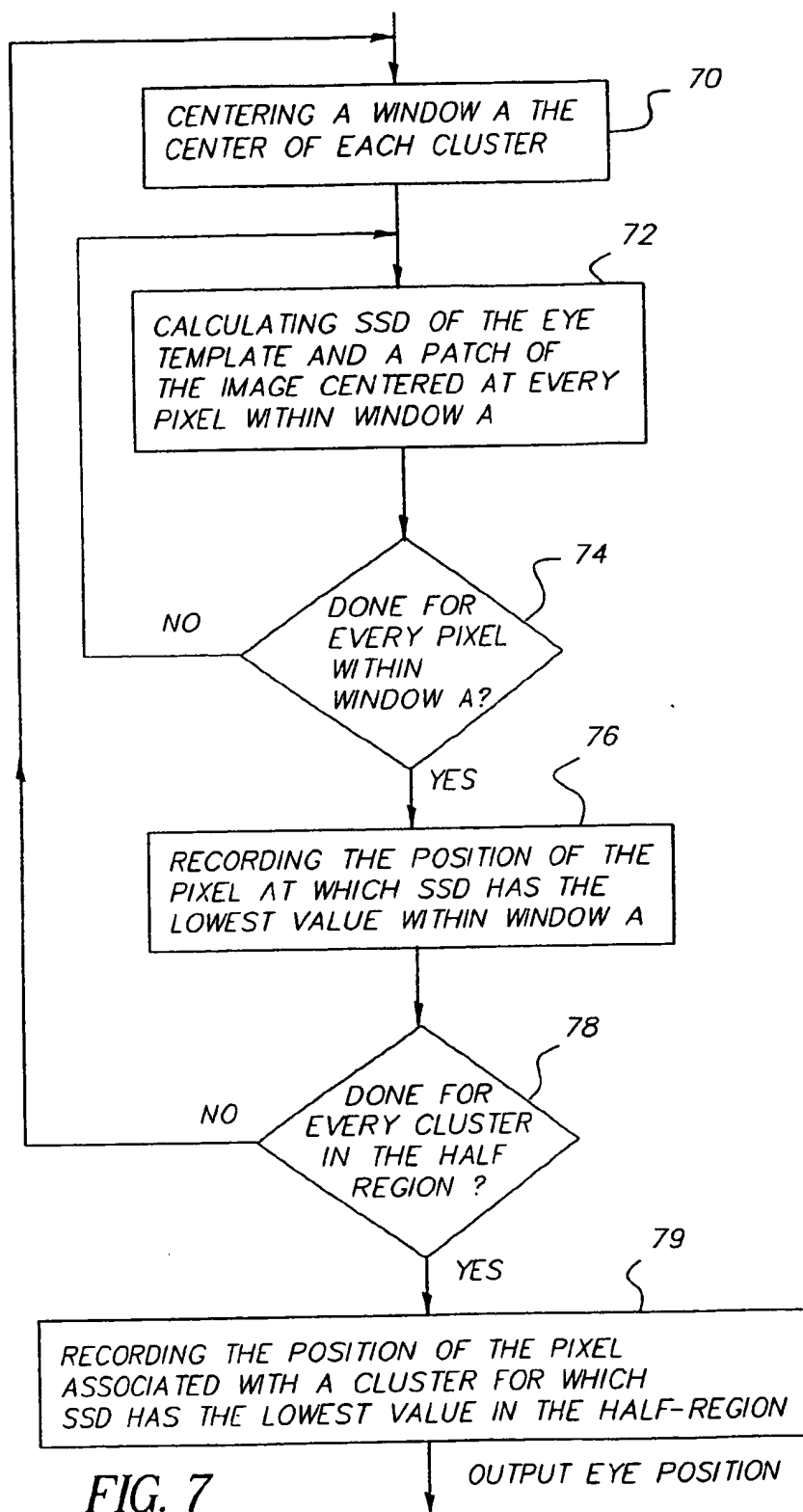
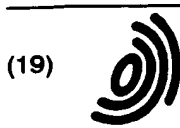


FIG. 8





(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
14.01.2004 Bulletin 2004/03

(51) Int Cl.7: G06K 9/00

(43) Date of publication A2:
26.06.2002 Bulletin 2002/26

(21) Application number: 01204728.8

(22) Date of filing: 06.12.2001

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
• Chen, Shoupu
Rochester, New York 14650-2201 (US)
• Ray, Lawrence A.
Rochester, New York 14650-2201 (US)

(30) Priority: 19.12.2000 US 740422

(71) Applicant: EASTMAN KODAK COMPANY
Rochester, New York 14650 (US)

(74) Representative: Halle, Helen Cynthia et al
Kodak Limited
Patent, W92-3A,
Headstone Drive
Harrow, Middlesex HA1 4TY (GB)

(54) Digital image processing method and computer program product for detecting human irises in an image

(57) A digital image processing method is taught for detecting human irises in a digital image. The method comprises the steps measuring the red intensity of the pixels in the image, determining the probability that each pixel is an iris based upon the red intensity of the pixel, determining the probability that each pixel is not an iris based upon the red intensity of the pixel; and determining whether the pixel is an iris by analyzing the relationship between the probability that the pixel is an iris and the probability that the pixel is not an iris. In one embodiment of the present invention, the determination as to whether a pixel is an iris pixel is then made based upon

the application of a Bayes model to the probability that the pixel is not an iris, the probability of the occurrence of an iris in the identified region and probability of the occurrence of a non-iris pixel in the identified region. In another embodiment of the present invention, the method comprises the steps of finding an oval shaped skin color region, detecting iris color pixels in the oval shaped skin color region, detecting iris color pixels in the oval shaped region using a Bayes model and locating eye positions based upon the detected iris color pixels.

A computer program product for performing these methods is also taught.

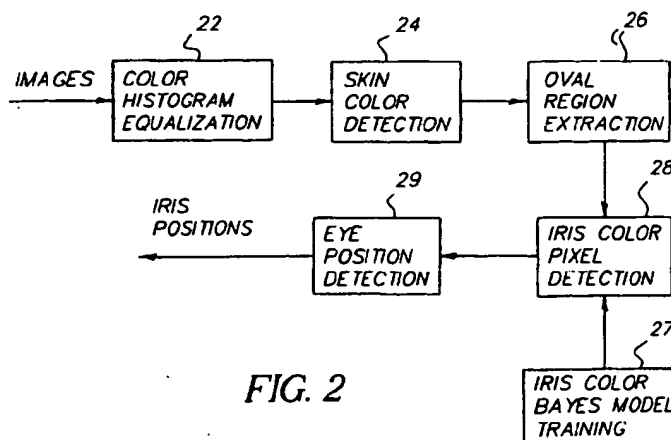


FIG. 2



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 20 4728

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	WO 99 23600 A (CHANG SHIH FU ;UNIV COLUMBIA (US); WANG HUALU (US)) 14 May 1999 (1999-05-14) * page 10, column 5 - column 20 *	1,2,5, 7-10	G06K9/00
A	CHANG M M ET AL: "ADAPTIVE BAYESIAN SEGMENTATION OF COLOR IMAGES" JOURNAL OF ELECTRONIC IMAGING, SPIE + IS&T, US, vol. 3, no. 4, 1 October 1994 (1994-10-01), pages 404-414, XP000478117 ISSN: 1017-9909 * paragraph [0002] *	1,2,5, 7-10	
A	EP 0 899 680 A (EASTMAN KODAK CO) 3 March 1999 (1999-03-03) * paragraph [0023] - paragraph [0024] * * paragraph [0025] - paragraph [0032] *	3,4,6, 8-10	
A	WO 97 21188 A (SARNOFF DAVID RES CENTER) 12 June 1997 (1997-06-12) * page 17, line 25 - page 18, line 2 * * page 34, line 16 - line 22 * * page 43, line 29 - page 44, line 4 *	3,6,8-10	TECHNICAL FIELDS SEARCHED (Int.Cl.7) G06K
A	PAPPAS T N ET AL: "AN ADAPTIVE CLUSTERING ALGORITHM FOR IMAGE SEGMENTATION" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COMPUTER VISION. TAMPA, DEC. 5 - 8, 1988, WASHINGTON, IEEE COMP. SOC. PRESS, US, vol. CONF. 2, 5 December 1988 (1988-12-05), pages 310-315, XP000079956 * paragraph [0002] *	1,2,5, 7-10	
The present search report has been drawn up for all claims			
Place of search MUNICH		Date of completion of the search 17 November 2003	Examiner Rimassa, S
CATEGORY OF CITED DOCUMENTS X: particularly relevant if taken alone Y: particularly relevant if combined with another document of the same category A: technological background O: non-written disclosure P: intermediate document		T: theory or principle underlying the invention E: earlier patent document, but published on, or after the filing date D: document cited in the application L: document cited for other reasons a: member of the same patent family, corresponding document	

EPO FORM 1503 03 02 (P04001)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 01 20 4728

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

17-11-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9923600 A	14-05-1999	WO 9923600 A1	14-05-1999
EP 0899680 A	03-03-1999	US 6151403 A	21-11-2000
		EP 0899680 A2	03-03-1999
		JP 11149559 A	02-06-1999
WO 9721188 A	12-06-1997	AU 1127197 A	27-06-1997
		EP 0865637 A1	23-09-1998
		WO 9721188 A1	12-06-1997

EPO FORM P0158

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference 2004 P 09036 WO	FOR FURTHER ACTION see Form PCT/ISA/220 as well as, where applicable, item 5 below.	
International application No. PCT/US2005/018373	International filing date (day/month/year) 25/05/2005	(Earliest) Priority Date (day/month/year) 27/05/2004
Applicant SIEMENS CORPORATE RESEARCH, INC.		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 4 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. Basis of the report

- a. With regard to the language, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ The international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

- b. ☐ With regard to any nucleotide and/or amino acid sequence disclosed in the international application, see Box No. I.

2. ☐ Certain claims were found unsearchable (See Box II).

3. ☐ Unity of invention is lacking (see Box III).

4. With regard to the title,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the abstract,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the drawings,

- a. the figure of the drawings to be published with the abstract is Figure No. 5

☒ as suggested by the applicant.

☐ as selected by this Authority, because the applicant failed to suggest a figure.

☐ as selected by this Authority, because this figure better characterizes the invention.

- b. ☐ none of the figures is to be published with the abstract.

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06T7/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	XIANGYANG LAN ET AL: "A unified spatio-temporal articulated model for tracking" COMPUTER VISION AND PATTERN RECOGNITION, 2004. CVPR 2004. PROCEEDINGS OF THE 2004 IEEE COMPUTER SOCIETY CONFERENCE ON WASHINGTON, DC, USA 27 JUNE - 2 JULY 2004, PISCATAWAY, NJ, USA, IEEE, vol. 1, 27 June 2004 (2004-06-27), pages 722-729. XP010708631 ISBN: 0-7695-2158-4 the whole document	1-16
A	US 6 597 801 B1 (CHAM TAT-JEN ET AL) 22 July 2003 (2003-07-22) abstract; figures 1-16 column 1, line 16 - column 4, line 57 column 5, line 51 - column 19, line 56 ----- -/--	1-16

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- * & * document member of the same patent family

Date of the actual completion of the international search

13 September 2005

Date of mailing of the international search report

05/10/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Herter, J

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DONOHUE G W: "Combining Segmentation And Tracking For The Classification Of Moving Objects In Video Scenes" CONFERENCE RECORD. TWENTY-SECOND ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS (CAT. NO.88CH2835-7) MAPLE PRESS SAN JOSE, CA, USA, vol. 2, 31 October 1988 (1988-10-31), pages 533-538, XP010325202 the whole document -----	1-16
A	EP 1 217 572 A (EASTMAN KODAK COMPANY) 26 June 2002 (2002-06-26) abstract; figures 1-7 paragraph '0001! - paragraph '0039! -----	1-16

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US2005/018373

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6597801	B1	22-07-2003	NONE
EP 1217572	A	26-06-2002	JP 2002230547 A 16-08-2002
		US 2002106112 A1	08-08-2002

A Unified Spatio-Temporal Articulated Model for Tracking

Xiangyang Lan Daniel P. Huttenlocher
{xylan,dph}@cs.cornell.edu
Cornell University
Ithaca NY 14853

Abstract

Tracking articulated objects in image sequences remains a challenging problem, particularly in terms of the ability to localize the individual parts of an object given self-occlusions and changes in viewpoint. In this paper we propose a two-dimensional spatio-temporal modeling approach that handles both self-occlusions and changes in viewpoint. We use a Bayesian framework to combine pictorial structure spatial models with hidden Markov temporal models. Inference for these combined models can be performed using dynamic programming and sampling methods. We demonstrate the approach for the problem of tracking a walking person, using silhouette data taken from a single camera viewpoint. Walking provides both strong spatial (kinematic) and temporal (dynamic) constraints, enabling the method to track limb positions in spite of simultaneous self-occlusion and viewpoint change.

1. Introduction

We consider the problem of detecting and tracking an articulated object, such as a person in a video sequence, using a two-dimensional parameterized model. The main focus of our work is on handling self-occlusions and changes in viewpoint, which remain challenging problems in the tracking of articulated objects. Our approach combines pictorial structure spatial models and hidden Markov temporal models into a single unified framework. A pictorial structure [5] consists of a set of rigid parts where certain parts are connected by springs to allow for bending and stretching. The parts and their constraints are two-dimensional, similar to the cardboard people models used in [7] for tracking people, but within a more general spatial modeling framework. Hidden Markov models (HMM's) have been used extensively for tracking as well as for gait and gesture recognition. In our approach the states of the HMM correspond to parameterized pictorial structure models that capture spatial characteristics of representative views or "key frames". The transitions between states capture temporal dynamics

based on interpolation between representative views.

The main distinguishing characteristic of our work is the integration of parameterized spatial models and temporal models into a common framework. When the states of an HMM are themselves parameterized models, it is generally intractable to perform inference – e.g., to find the best sequence of states for a given input sequence, or even to find good such state sequences. Here we provide a good inference procedure for object detection and tracking, when the states of the HMM are pictorial structure spatial models. The resulting combined model provides both strong spatial (or kinematic) constraints from the pictorial structures and strong dynamic constraints from the HMM. The model clearly delineates between the spatial and dynamic constraints, yet the two are brought together in a common estimation framework.

There are a wide range of approaches to detection and tracking, many of which are surveyed in [6]. In contrast with our approach, most prior work that uses parameterized-state HMM's for tracking articulated objects is based on having linear dynamic system (LDS) models for the states (e.g., [1, 8]). Such methods have strong dynamic constraints but relatively weak spatial constraints, because the LDS models (and associated Kalman filtering techniques) capture primarily dynamic rather than spatial characteristics. HMM's have also been used in conjunction with template spatial models, for problems in gesture, gait and action recognition (e.g., [3]). These methods differ from our work in that each state is a template rather than a parameterized model, making it difficult to adapt such methods to tracking the actual configuration of parts for an articulated object. Parametric HMM's have been used in other contexts, such as gesture recognition (e.g., [12]). But their work is more focused on model learning, and difficult to apply to high-dimensional problems such as tracking articulated objects.

Pictorial structure models have recently been successfully applied to tracking humans [10], but without explicit models of the temporal constraints. Our work extends those results by using an HMM temporal model, and as a consequence is able to handle changes in viewpoint and self-



Figure 1: The default configuration for a simple pictorial structure model corresponding to a side view of a person walking. Red denotes the person's left limbs, blue their right limbs and green their head and torso.

occlusions. The focus in [10] is on learning good appearance models for the parts of an object, for use in tracking, whereas our focus is on combining spatial and temporal constraints. Thus the two pieces of work are complementary. Another recent approach to tracking articulated objects uses a tree-based filtering method [11]. They demonstrate the method for hand tracking but it could also be applied to other articulated objects. In their approach a large number of views are clustered together and the focus of the work is on using a tree structure to rapidly search for the best matching view. In contrast we use a small number of representative views for accurate tracking of multiple parts, by having a highly parameterized pictorial structure model.

2. An Integrated Spatio-Temporal Articulated Model

We now discuss the specifics of the pictorial structure and HMM models, and how they fit together in an integrated modeling framework. A pictorial structure is a parametric spatial model composed of multiple parts, with spring-like connections between certain pairs of parts. An instance of such a model consists of the parts $P = (p_1, \dots, p_n)$ and the connections $C = \{c_{ij}\}$, where each c_{ij} represents the spatial constraints between parts p_i and p_j . Each part p_i has parameters representing the appearance of that part. For instance the appearance of a part may be characterized by local oriented filter responses, template models, feature detectors, etc. Each connection c_{ij} has parameters that encode both the ideal relative spatial configuration of two parts and spring-like constants controlling deformations from this ideal configuration. For instance the spatial relations may specify a relative position and orientation together with covariances that characterize the degree of bending and stretching between the parts.

If we let $\mu = (P, C)$ denote such a pictorial structure model, then a *configuration* of the model is given by $\Theta = (\theta_1, \dots, \theta_n)$ where the θ_i are parameters specifying the location of each part p_i in an image coordinate frame. In a Bayesian framework the prior distribution of configurations for a given model, $P(\Theta|\mu)$, characterizes the probability of different configurations occurring, independent of

any observed data. This prior is governed by the connection parameters C , as described in [4]. The highest probability such configuration can be thought of as the preferred or "default" configuration of the model. Such a configuration is illustrated in Figure 1 for a model corresponding to a side view of a person walking, when their arms and legs are maximally extended. We use the color red to denote the left arm and leg, blue to denote the right arm and leg, and green to denote the torso and head. Deviations from this maximal prior probability configuration can be thought of as incurring a cost, in that such configurations have a lower chance of occurring.

The posterior distribution of the configuration parameters,

$$P(\Theta|J, \mu),$$

characterizes the probability of different configurations given a particular model μ and image J . There are several standard definitions of what constitute good configurations. One definition is the *MAP estimate*, which is a set of parameter values that maximize the posterior probability. Another definition is any set of parameter values for which the posterior probability is high. Such values are generally computed using *sampling* methods, where configurations are selected at random, weighted by their probability. Generally some other technique is used to select the best of these sampled configurations. Following [4] we use the Chamfer distance as a selection criterion.

Using Bayes' rule, the posterior can be expressed in terms of the product of a likelihood and prior,

$$P(J|\Theta, \mu)P(\Theta|\mu). \quad (1)$$

In general for a high-dimensional parameter space Θ it is not tractable to find the Θ^* that maximizes this probability, or to sample values of Θ that have high probability. In the case of pictorial structure models where the parts have a tree-like skeletal structure, such as a human body or hand, both these problems can be solved efficiently using dynamic programming methods as described in [4].

We now turn from the spatial to the temporal component of the models. Let $S = \{s_1, \dots, s_n\}$ be a set of representative views of an object. Representative views are analogous to "key frames" in animation, in that a video sequence can be summarized by a sequence of representative views. Intermediate frames between two representative views can be generated by interpolation of those views. As noted in the introduction, while representative view temporal models have been used in other work, in these models the views are generally templates or exemplars and are thus not well suited to tracking the configuration of parts of articulated objects. In our case each representative view is a parameterized pictorial structure model rather than a template image.

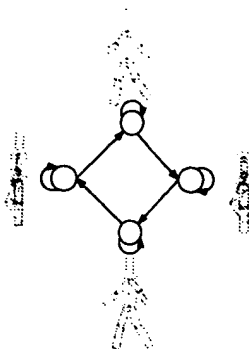


Figure 2: Four state model corresponding to a side view of a person walking.

A hidden Markov model (HMM) is a stochastic finite automaton, where each state generates (or accepts) some observation. Let Q_t represent the hidden state and Y_t represent the observation at time t , $1 \leq t \leq T$, where the states take on values in some set S and the observations take on values in some set O . An HMM is characterized by the tuple $\lambda = (A, B, \pi)$ where the *transition probabilities* are $A(i, j) = P(Q_t = j | Q_{t-1} = i)$ for $i, j \in S$, the *observation probabilities* are $B(i, k) = P(Y_t = k | Q_t = i)$ for $k \in O$ and $i \in S$, and the *initial probabilities* are $\pi(i) = P(Q_1 = i)$ for $i \in S$. For a good tutorial on HMM's see [9]. In our case the observation is the image at each time t , and each state is a pictorial structure model corresponding to some representative view.

Given an observed sequence of images $Y = (Y_1, \dots, Y_T)$ and an HMM $\lambda = (A, B, \Pi)$ we want to determine a corresponding best sequence of representative views $Q^* = (Q_1^*, \dots, Q_T^*)$. This temporal estimation problem is commonly expressed as that of finding a maximum posterior probability (MAP) state sequence

$$Q^* = \operatorname{argmax}_Q P(Q|Y, \lambda),$$

and can be solved in time $O(Tn^2)$ for a given model λ and observation sequence Y using the Viterbi algorithm.

It is common to consider an HMM in terms of its *state space* graph, where the nodes correspond to states and the arcs correspond to non-zero entries in $A(i, j)$. By way of illustration, Figure 2 shows such a graph for the case of a side view of a person walking (we focus more in depth on walking in Section 3). The Figure shows each state together with the highest prior probability configuration of the corresponding pictorial structure model for that state. The pictorial structures for all four states of this model consist of the same set of parts, P . Only the spatial connection parameters, C , differ between states, as the only differences in the views are in the expected part locations. The states shown at the top and bottom of the Figure correspond to maximal extension of the arms and legs, and differ only in whether the

right arm and left leg are forward or vice versa. The states shown at the sides of the Figure correspond to minimal extension of the arms and legs, and differ only in whether the right leg is down and the left leg raised or vice versa.

Note that in this model the two states corresponding to minimal extension of the limbs cannot be distinguished on the basis of a single observed image, because they differ only in which leg is bent. Similarly for the two states corresponding to maximal extension of the limbs. Over time (and given a starting state) these ambiguities can be resolved based on the allowable state transitions indicated by the arrows, thus making it possible to distinguish between states and thereby determine which parts are occluded. This illustrates how the combination of simple parameterized spatial models and temporal models can be used to eliminate ambiguity and localize parts that are occluded by other parts.

2.1. Combining the Spatial and Temporal Models

When the states of an HMM are parameterized models, it is often not tractable to determine a best state sequence, Q^* , because the observation probabilities cannot be reasonably estimated. For HMM's with LDS state models, approximate inference procedures have been developed (e.g. [8]). Here we consider the case of HMM's with pictorial structure state models, where it turns out there is a simple inference procedure. In this combined spatio-temporal model, the observation probability, $B(i, k) = P(Y_t = k | Q_t = i)$, results from the parameterized pictorial structure spatial model, μ_i . This is simply the probability over all possible configurations, which can be determined by integrating over the model parameters,

$$P(Y_t = k | \mu_i) = \int P(Y_t = k | \Theta_i, \mu_i) P(\Theta_i | \mu_i) d\Theta_i. \quad (2)$$

Computing this integral when Θ_i is a high-dimensional parameter vector, as is the case here, is generally intractable. A standard approach to approximating such integrals is to sample from the distribution. In our case this sampling can be done efficiently because the distribution is the posterior over configuration parameters in equation (1), which as noted above can be sampled efficiently for a tree-structured model. Moreover, as a byproduct of the sampling methods in [4] the integral in (2) can actually be computed directly without needing to sample configurations.

Given an image sequence $Y = (Y_1, \dots, Y_T)$, the best state sequence $Q^* = (Q_1^*, \dots, Q_T^*)$ can thus be computed using the Viterbi algorithm, once the observation probabilities $B(i, k) = P(Y_t = k | Q_t = i)$ for each image and state have been computed using the corresponding distribution of the form in (1). For the best state Q_t^* at each time t , sampling can be used to find parameters Θ^* that correspond to a high probability configuration of the model. Thus the best

state sequence naturally specifies a corresponding sequence of configuration parameter values $\Theta_1^*, \dots, \Theta_T^*$, specifying a high probability configuration of the best matching model μ_i at each time t . These configurations take into account both the spatial constraints in matching individual pictorial structure models to images and temporal constraints in selecting the best state sequence Q^* .

The above sequence of configuration parameters is based on matching the best representative view model at each time frame. When parts are occluded, interpolation between successive representative views can be used to predict better part locations than are obtained by simply using one of the representative view models. This can be handled easily in the current framework in one of several ways. One approach is to simply introduce additional states into the HMM that are intermediate between representative views. These views have connection parameters, C that are based on interpolation between adjacent representative views. The main disadvantage of this approach is that it increases the number of states and pictorial structure models, which slows down processing.

An alternative approach to obtaining better estimates of the locations of occluded parts is to re-estimate the parameters Θ_i at each time, after first finding the best state sequence without the addition of any interpolated states to the HMM. The observation probability $P(Y_t = k | Q_t^* = i)$ reflects how well the best model μ_i accounts for the observed image at time t . This probability will be lower for time frames that lie between two representative views compared with those that are a good match to a single representative view. This can be used to determine the degree to which an image is intermediate between representative views and then generate a corresponding interpolated model. The configuration can then be estimated by matching this interpolated model to the image rather than the model for the best state. We use this approach to obtain better estimates for the locations of occluded parts.

As described above the transition probabilities A are stationary, and do not vary with time. In our context it could further be useful to have transition probabilities that vary with time. For instance, one could envision stronger dynamic models where the transition probabilities vary. Given such additional estimates the same Viterbi method can be applied as above except $A = (A_1, \dots, A_t)$ is varying with time.

A drawback of the Viterbi approach is that it operates on an entire sequence of frames at once, whereas in interactive tracking applications it is desirable to do incremental processing one frame at time rather than batch processing on an entire sequence. There are several standard ways of handling this including using the forward probabilities from the forward-backward algorithm, rather than the Viterbi method. This corresponds to summing rather than

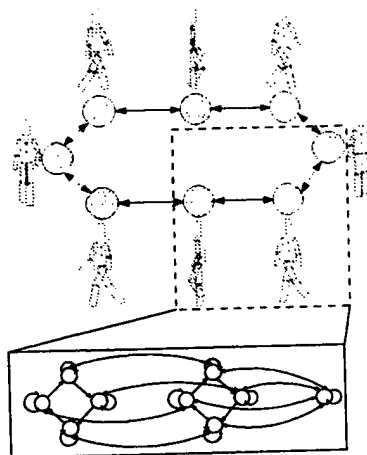


Figure 3: Modeling transitions between viewpoints.
maximizing products of probabilities.

3. Application to Human Walking

In the remainder of the paper we consider the problem of tracking a walking person, including determining the locations of their limbs, using silhouette data taken from a single camera viewpoint. Human walking has relatively simple temporal characteristics, yet this problem is still challenging due to single-frame ambiguities, self-occlusion and changes in viewpoint. This section will show how the pictorial structure based HMM framework introduced above can be used to tackle these difficulties effectively.

3.1. A Model of Walking

When only a single viewpoint is considered the simple four state model introduced above, and shown for a side view in Figure 2, provides a generic description of walking as a cyclical gait between four representative views. Similar four-state models have been proposed and used in a wide variety of contexts, both in computer vision and in studies of human walking.

To allow for changes in viewpoint, we consider the eight possible viewing directions of front, back, left, right and 45-degree views halfway between them (front-left, front-right, back-left and back-right). The only allowable state transitions between viewpoints are to stay at the same viewpoint or to change to one of the immediately adjacent viewpoints. These states and transitions are illustrated in the graph at the top of Figure 3. Each node in this graph actually corresponds to an entire cyclical gait model. We use the four-state gait model in Figure 2 for the two side views, a similar model for the four 45-degree views, and a one-state model for the front and rear views, as illustrated in the expanded

box at the bottom of Figure 3. The 45-degree gait models differ from the side models in having less occlusion of the arms and legs in the minimum extent configuration and a smaller spread of the arms and legs in the maximum extent configuration. For the front and back views the gait model consists of just a single state because the positions of the arms and legs do not change appreciably during the walking cycle from these viewpoints.

The full walking model thus has 26 states, four states for each of the side views (left and right), four states for each of the 45-degree views, and one state each for the front and back views. Conceptually there are three different types of transition edges in this graph: *self transitions* from a given representative view to itself, *gait transitions* from one state in a gait cycle to the next but maintaining the same viewpoint, and *view transitions* where the viewpoint changes but the state in the gait cycle stays the same. One could further imagine a combined transition that simultaneously goes from one viewpoint to the next and one gait state to the next, but we do not consider that here.

3.2. Estimating Model Parameters

Given the graph structure there are a number of parameters to be estimated. We use the maximum likelihood estimation approach in [4] to learn pictorial structure spatial models from a set of labeled training images. Each training image specifies the viewpoint and the locations of the ten model parts (torso, head, and upper and lower left and right arms and legs). All spatial relations between the parts are learned from the examples, including covariances capturing the degree of change in the relative locations of the parts for a given representative view. In practice many of the viewpoints have the similar spatial relations between their parts, so training need not be done separately for each viewpoint. For instance in the 26 view model above there are only 5 distinct spatial arrangements of parts: the minimum and maximum extent side views, the minimum and maximum extent 45 degree views, and the front/back view.

For the transition probabilities in the HMM we simply use constants corresponding to each of the three transition types identified above: τ_s for self transitions, τ_g for gait transitions and τ_v for view transitions. The transition probabilities for a given node are then obtained by normalizing these values so that the total over all outbound edges from the node is 1.0. We set these constants such that $\tau_s > \tau_g > \tau_v$, and use the same values for all experiments.

In order to generate more accurate locations of occluded parts we interpolate the pictorial structure model parameters between the representative view states, as discussed above. We use linear interpolation of the parameters, although for walking a more accurate model would reflect that near the representative views the change is slower than halfway be-

tween such views. The number of intermediate views is determined by the rate of walking. In principle this can be estimated from sample video sequences, but we simply use a fixed value at the moment.

This modeling approach provides a framework for learning parameters such as the transition probabilities. Thus possible extensions to this work include clustering images to automatically identify distinct views, and estimation of HMM parameters using EM.

3.3. Approximations

While pictorial structure matching is efficient given the large number of parameters to be estimated, it still takes approximately half a minute to match a ten-part person model to an image using a 2 GHz processor. With one pictorial structure model per state, the processing requirements for a large spatio-temporal model such as the one here is thus many minutes per frame (although quite easily parallelized). In this section we consider two methods for speeding up processing in the specific case of tracking walking people. The resulting technique runs at about 15 seconds per frame for the full model considered above (on a 2.0 GHz Pentium 4). The first technique uses the horizontal variance of the silhouette pixel locations to determine the best state sequence, and then only matches the pictorial structure model that corresponds to the best state at each time. The second technique uses a bounded velocity assumption to limit the parameter range that is searched in finding the best pictorial structure parameters for a given image frame.

Our goal here is to compute a best state sequence $(Q_1^*, \dots, Q_T^*)$ without needing to estimate the integral in (2) for every model and image. Thus we consider an approximation to the observation probabilities $P(Y_t = k | Q_t = i)$. This approximation is based on using the horizontal variance of the model pixel locations rather than the full model. As has been observed in other work (e.g., [2]), most of the change in the silhouette of a walking person is in the horizontal locations of the pixels. Let $f(t)$ denote the variance in the x -locations of the silhouette pixels at time t , where the silhouette has first been scaled to a standard height in order to accommodate overall size changes. We define the normalized horizontal variance as the difference between $f(t)$ and a smoothed version of $f(t)$. Namely, $g(t) = f(t) - (f(t) * G_p(t))$, where G_p is a Gaussian of standard deviation p , the period of $f(t)$. The period, p , can be determined easily using auto-correlation or the FFT. If the speed of walking changes substantially then windowing can be used to vary the smoothing.

An example of the normalized horizontal variance is shown in Figure 4 for a video sequence of a person walking

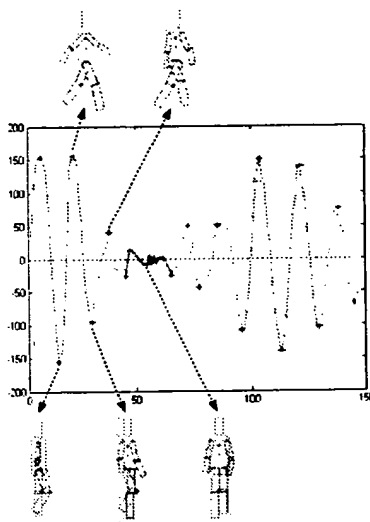


Figure 4: The normalized horizontal variance, $g(t)$, for a person walking in a circle, illustrating the combined effects of changes in viewpoint and gait cycle.

around in a circle, so the viewpoint is changing. The Figure also shows corresponding representative pictorial structure models for several of the views. Note that the primary effect of the normalization is to create a common "zero point" around which $g(t)$ oscillates in a periodic manner. The gait cycle is captured by the oscillation of the function and the viewing direction is captured by its amplitude. In order to explicitly represent the amplitude at each time we also compute the separation between the minimal and maximal values over each period, $h(t) = g_{\max}(t) - g_{\min}(t)$ where the min and max are computed for the period containing t .

For states within a gait cycle, the horizontal variance $g(t)$ distinguishes the two minimal extent states, which have negative values of $g(t)$, from the two maximal extent states, which have positive values. Analogously to the pictorial structure models, in a single time frame the two minimal extent states cannot be distinguished from one another, nor can the two maximal extent states, but these states are distinguishable based on transitions between states over time.

For changes in viewpoint, the amplitude $h(t)$ distinguishes the two side views (colored magenta in the Figure), which have high amplitude, from the four 45 degree views (cyan), which have moderate amplitude, from the front and back views (black), which have low amplitude. However, even with the associated state transitions, $h(t)$ is not enough to distinguish different 45 degree views. Recall the view transition diagram in Figure 3. The transitions from a side view (or from a front/back view) are both to 45 degree views which have essentially the same amplitude. However, while the overall amplitude is the same, these two views differ in terms of whether or not the 45 degree viewing direction

aligns both the forward and backward arms with the torso (and vice versa for the legs as they are in opposing positions to the arms) while the viewpoint change is happening. Thus we also compute the amplitude of the normalized horizontal variance separately for the upper half of the silhouette (corresponding to the arms) and the lower half (corresponding to the legs) to capture these differences. Call these amplitudes h^u and h^l , respectively.

We use a normal distribution to model these parameters for a given state i . Thus there are a total of eight parameters to this model, the mean and variance for each of the normalized horizontal variance g , its amplitude, h , and the amplitudes for the upper and lower halves h^u and h^l . Let v_i denote these eight parameters for state i . We approximate the observation probabilities using $P(Y_t = k|v_i)$. Given the best state sequence $Q^* = (Q_1^*, \dots, Q_T^*)$ found using this approximation, we then compute the spatial parameters Θ_t for each time frame by matching the pictorial structure model corresponding to state Q_t^* to the image at time t by sampling from the distribution in (1). As discussed above, the pictorial structure model is interpolated between representative views, so as to provide a stronger prior on the locations of parts and thus increase the localization accuracy for occluded parts.

In the final step of computing the spatial parameters Θ_t , we use a bounded velocity assumption to further speed up matching by eliminating large portions of the parameter space from consideration when sampling the posterior distribution of model parameters in equation (1). The location of each part is characterized by a set of parameters. Bounding the velocity implies bounds on the degree of change in the parameters from one time frame to the next. This makes it possible to restrict the domain over which the posterior must be estimated. We use a simple hypercube, limiting each parameter to an interval around the corresponding value for the previous time-frame. Recall that estimating the posterior involves computing both the likelihood and prior over possible values of these parameters for each part, whose domains are in turn limited. This results in an order of magnitude speedup. Such windowing cannot only drastically shrink the volume of the state space that needs to be computed, it can also improve the accuracy of the matching results when there is a good prior on the part locations but certain parts are hidden from view.

3.4. Examples

In this section we present some experimental results to demonstrate the capabilities of the method in tracking a human silhouette. We consider three sequences, the first two have a fixed viewpoint and in the third a person walks in a circular path so that the viewpoint changes extensively.

The first two sequences are from the multi-view imagery

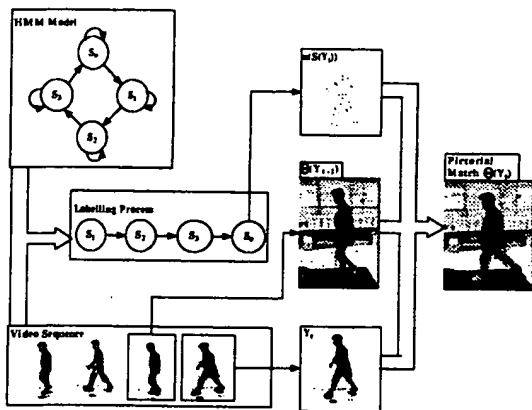


Figure 5: This graph illustrates our experimental framework. The three major steps are, (1) silhouette extraction, (2) state labeling using approximation, (3) windowed-sampling based pictorial structure matching.

in the CMU HID dataset. Each sequence is an 11 second clip of a person walking on a treadmill. One sequence is taken from a side-view and the other from a front-side 45 degree view. The third sequence, from Michael Black, is of a person walking in a circle and poses substantial challenges in terms of self occlusion and viewpoint change, demonstrating the power of having integrated spatial and temporal models (we refer to this as the PARC sequence).

Our experimental framework can be summarized by the illustration in Figure 5. There are three major steps in the whole process. The first step is silhouette extraction using background subtraction because of the stationary camera. The remaining steps of processing are all covered in the previous sections. The second major processing step involves calculating the horizontal variance measures and Viterbi estimation of the best state sequence given the walking model and the horizontal variance. The third major step is estimation of the model parameters for the pictorial structure model corresponding to the best state at each time t (including the use of interpolated models between representative views). These model parameters are then used to overlay the model on the original images. The calculations are all done using a size-normalized coordinate frame and then mapped back into image coordinates.

Figure 6 shows some of the results for the side-view walking sequence. These frames illustrate that our approach can handle self-occlusion and the ambiguity in the identity of the parts in individual frames. Note that the distinctions between the left (colored red) and right (colored blue) limbs are correctly tracked over time, including through occlusions. The matching is completely automatic, except that for the first frame we specify whether the left arm and right leg are forward, or vice versa, as this cannot be distinguished from the image data. This is done by setting the initial state



Figure 6: Results for the side view of the treadmill sequence.



Figure 7: Results for the 45-degree view of the treadmill sequence.

probabilities of the HMM. Figure 7 shows some results for the 45-degree viewpoint sequence, again correctly tracking the parts over time and through occlusions. These two sequences illustrate the capabilities of the single-view four-state HMM gait model, using pictorial structures for the states.

Figure 8 shows the results of our method for the PARC sequence, where a person walks in a circle. Note the substantial changes in viewpoint and self occlusions. This result demonstrates that the full model, incorporating both viewpoint and gait changes, can successfully track the left and right side limbs over a sequence that combines many successive changes in viewpoint simultaneous with the self-occlusions due to the gait cycle. This is a challenging sequence and we are not aware of other methods that can successfully track the parts over time as demonstrated here.

4. Conclusions

We have presented a modeling framework that integrates pictorial structure spatial models with hidden Markov temporal models, and have shown an efficient inference procedure for finding a maximum a posteriori (MAP) probability state sequence for such a model. Our method also produces a sequence of configuration parameters corresponding to a high posterior probability match of the model at each image frame. We then described an approximation that uses the horizontal variation of pixel locations to more efficiently determine a MAP state sequence. The results illustrate the ability of this combined spatio-temporal modeling technique to correctly track the position and identity of a person's limbs through self occlusions and changes in viewing direction.

References

- [1] C. Bregler, Learning and Recognizing Human Dynamics in Video Sequences. *CVPR*, San Juan, Puerto Rico, June 1997
- [2] R.T. Collins, R. Gross and J. Shi, Silhouette-Based Human Identification from Body Shape and Gait, International Conference on Face and Gesture, pp. 351-356, 2002.
- [3] A. Elgammal, V. Shet, Y. Yacoob and L.S. Davis, Learning Dynamics for Exemplar-Based Gesture Recognition. *CVPR*, Madison, Wisconsin, June 2003.
- [4] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial Structures for Object Recognition, *IJCV* to appear.
- [5] M.A. Fischler and R.A. Elschlager, The representation and matching of pictorial structures. *TC*, 22(1):67-92, January 1973.
- [6] D. M. Gavrilu. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, Academic Press, vol. 73, nr. 1, pp. 82-98, 1999.
- [7] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In 2nd Int. Conf. on Automatic Face and Gesture Recognition, pages 38-44, 1996.
- [8] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. *NIPS*, 2000.
- [9] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* Vol. 77(2), pp. 257-286, 1989.
- [10] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. *CVPR*, Madison, Wisconsin, June 2003.
- [11] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. *ICCV*, Vol. II, pages 1063-1070, Nice, France, October 2003
- [12] A.D. Wilson and A.F. Bobick, Parametric Hidden Markov Models for Gesture Recognition, *IEEE Trans. PAMI*, v. 21, pp. 884-900, 1999.



Figure 8: Results for the multi-viewpoint sequence.

BEST AVAILABLE COPY

Combining Segmentation and Tracking for the Classification of Moving Objects in Video Scenes

Gregory W. Donohoe

Exploratory Systems Development Division 9133
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This paper describes a method of detecting, tracking and identifying moving objects in video scenes. The method is based on an adaptive change detector which detects and tracks moving objects and extracts silhouettes of the objects from the background so they can be classified by shape. The adaptive change detector uses estimates of image noise and contrast to dynamically adjust decision thresholds. A principal feature of this method is the synergistic interaction between the tracker, the segmenter and the classifier to eliminate uninteresting objects, to improve estimates of the background and noise, to guide threshold selection, and to influence feature selection for classification.

1. Introduction

In some applications of machine vision it is desirable to detect, track and identify moving objects [4]. An important requirement for a practical machine vision system is that it be able to adapt to changing conditions without operator adjustment. It should operate reliably in the presence of environmental "noise" such as dust, cloud shadows, and small camera motion as well as electronic noise. It should be immune to nuisance objects such as distant traffic, flashing lights, and birds. It should function with a variety of targets against a wide range of backgrounds and with targets at different ranges.

A vision system that attempts to address these requirements is under development for applications that include physical security [7], automatic on-site inspection [1], and keeping track of nuclear materials in a storage facility [2]. This system exploits changes in image sequences to distinguish moving objects from the background. The camera is stationary with respect to the background. The system recursively computes a reference image which is an estimate of the background scene and compares current images with the reference. If a significant difference is detected between the current and reference images within an image region, a change has occurred in that region.

The basic change detection method can be applied to target detection and tracking (collectively called "cueing") and to segmentation. The two applications have different requirements, however. In cueing, the goal is to examine the entire image, detect moving objects, and track those objects as they move. Segmentation is the process of extracting a silhouette of an object so that it can be classified by shape. Both detection/tracking and segmentation will be described. Then it will be shown how the two can be combined to make both more effective and how both can contribute to classification.

1.1 Detection and Tracking

A method for tracking targets in change images has been described in detail in [5]. The basic change detection scheme is applied to the whole image at low resolution. Wherever a "change" is detected in a region of the image, the region contains a potential target. All connected pixels that have changed are grouped together to form one object; this object is placed on a list of known objects. In subsequent frames, changes are once again detected, grouped into objects, and compared to known objects. If a new object matches well with a known object on the basis of length, height, area, velocity and predicted location, the new object is assumed to be a new sighting of the old. The old object description is updated with new information. When a moving target has been seen several times, it becomes "qualified"; the tracker places a bounding box on the target. Objects that do not exhibit consistency in size and motion can be dismissed as random events or noise.

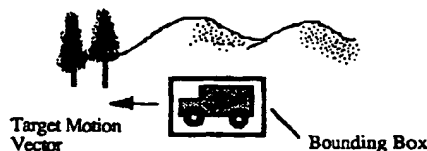


Figure 1: Using Change Detection for Target Tracking

The object tracker is characterized by low resolution, high speed, and high reliability. The latter is due largely to the fact that objects are observed many times before they are qualified.

The tracker "coasts," so that targets can escape detection for a few frames and be picked up again without losing their identity. The tracker also tries to piece together objects that have broken up so that different pieces of the object have been detected as unconnected image regions. Because of these multiple observations, the tracker is relatively immune to occasional missed detections and false alarms. The tracker is able to determine with great confidence that a target exists, and approximately where it is, but it is able to produce only the roughest target description.

When a target has been detected and tracked with sufficient confidence, the tracker can trigger or cue the segmenter.

1.2 Segmentation

When used as a segmenter, the change detector operates on a "snapshot" of the current and reference images to separate, or segment, moving objects from the background. The silhouettes of the objects can then be used for classification by shape. For certain kinds of targets, such as vehicles, silhouettes have the advantage of being relatively invariant from one instance of a target class to another. For example, two pickup trucks of the same model are likely to have very similar silhouettes, whereas they may differ considerably in other characteristics such as color.

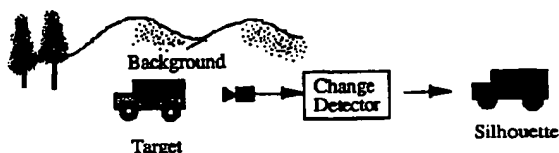


Figure 2: Change Detection for Segmentation

The segmenter is characterized by high resolution and low speed. For object classification, accurately locating the essential features of the target is important. It may be necessary to examine many pixels in overlapping regions to decide just where the object boundary lies. This requires great concentration of computing resources over a small area.

1.3 Classification

Many techniques have been developed for classifying objects from silhouettes. One that has been applied to this problem is to use central geometric moments as features [8] and to apply statistical pattern recognition methods [9].

2. The Adaptive Change Detection Algorithm

The adaptive change detector can be cast as a classical detection theory problem. Let us assume that the camera is stationary with respect to the background. In our system we have a reference image frame, r , which is a recursive estimate of the background image, and a *current* frame, c . We assume that a given pixel belongs either to a target or to the background. That is, we have either a *target* event, t , or a *background* event, b . At each pixel location (x,y) , we compare the current image $c_{x,y}$ and the reference image $r_{x,y}$. If the absolute value of the difference exceeds a threshold T , then the pixel belongs to a target; otherwise, it belongs to the background:

$$|c_{x,y} - r_{x,y}| \underset{b}{\overset{t}{\gtrless}} T \quad (1)$$

In order to make the system adaptive, we need to automatically compute the threshold T that gives the desired performance under changing conditions. Let us define an observation z such that $z = |c_{x,y} - r_{x,y}|$. Now the problem can be expressed in terms of classical detection theory:

$$\frac{p(z/t)P(t)}{p(z/b)P(b)} \underset{b}{\overset{t}{\gtrless}} \lambda \quad (2)$$

where

$p(z/t)$ is the probability density of observation z given the target event;

$P(t)$ is the discrete *a priori* probability that pixel (x,y) represents a target;

$p(z/b)$ is the probability density of observation z given the background event;

$P(b)$ is the discrete probability that the pixel contains the background.

The double inequality means that if the ratio on the left-hand side of (2) is greater than λ , we decide that the target event t has occurred; otherwise, we choose background event b . The decision threshold λ is determined by the type of decision rule. For those who subscribe to the Bayesian philosophy of decision theory, the choice of λ can be justified in terms of Bayes risk [6]. In practice, λ is often a fixed parameter chosen empirically by the user. Now the decision can be expressed in the form of Equation (1):

$$z \underset{b}{\overset{t}{\gtrless}} T \quad (3)$$

where

$$T = \lambda \frac{p(z/b)P(b)}{p(z/t)P(t)} \quad (4)$$

These relationships are illustrated in Figure 3 below for the case where $\lambda = 1$. In this case, T is chosen as the value of z where the two curves cross.

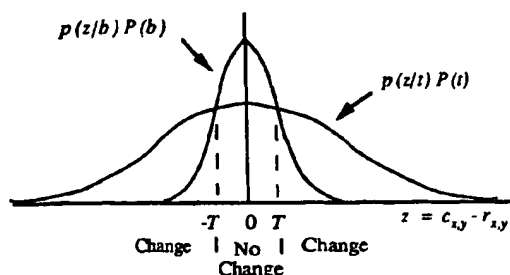


Figure 3: Change Detection Threshold for $\lambda = 1$.

2.1 Estimating the Conditional Probabilities

In order to adjust to changing conditions, the adaptive change detector must estimate the probabilities on the right-hand side of (4).

One approach to estimating $p(z/t)$ and $p(z/b)$ is to assume that the background image estimate is effectively noiseless, and that the system noise is additive. Then the uncertainty associated with the observation of the background image is equal to the system noise:

$$p(z/b) = p(n)$$

Now we must assume a distribution for targets. In many applications, a reasonable assumption is that target pixel grey levels are distributed in the same way as the background. This is because, in general, targets are made of the same kinds of materials as backgrounds and are subject to the same illumination, imaging and noise processes. (Although the targets and the background share the same statistics, in order to detect a target pixel, the given target pixel and corresponding background pixel must not be identical.)

These assumptions can lead to several parametric and non-parametric methods for determining $p(z/t)$, some of which are described in [3]. In order to conserve computing resources, it is often reasonable to assume that noise and image statistics are the same over the image. Then, a running estimate of these statistics can be obtained by sampling the reference frame and the current frame at randomly chosen locations during each image sampling interval. On the average, *if there are no targets present*:

- (1) the reference image grey level distribution corresponds to both background and target pixel distributions;
- (2) the difference between the reference image and the current image gives us the distribution of the additive noise.

In busy scenes, many changes occur due to moving objects; if it is not known that these changes are due to targets, they will be erroneously attributed to noise. This will cause the noise estimates to be artificially high. The adaptive change detection threshold is increased accordingly, and probability of detection suffers. Recall, however, that the tracker is able to tell us, with low resolution but high reliability, whether there are moving objects in the scene and approximately where they are. We can easily mask the noise and background estimates with the bounding boxes of all detected objects: if an image region contains a target, we don't use this region in updating the estimates. Thus, we can use the tracker to verify the condition that no targets are present in the part of the scene being analyzed.

2.2 Estimating the *a priori* Probabilities

We now need to make an estimate of the probability that a particular pixel belongs to the target or to background pixel in the first place. That is, explicitly or implicitly, we must make some assumptions about $P(t)$ and $P(b)$. Here the requirements for the tracker and the segmenter differ.

For target detection, the *a priori* probability that a pixel contains a target is likely to be quite small, on the order of 0.01. Since there are only two possible events, $P(b) = 1 - P(t)$. The change detection thresholds should be set accordingly.

Once a target has been found and qualified, the bounding box tells us where the target can be found. The segmenter can search aggressively within the bounding box to find the target silhouette. Recall that the tracker has kept a running estimate of the target area, that is, the number of pixels on the target. Within the bounding box, the *a priori* probability that a pixel belongs to a target is determined by

$$P(t) = \frac{\text{Area of Target}}{\text{Area of Bounding Box}}$$

$P(t)$ is likely to be quite large, around 0.80 (see Figure 2). Thus, the appropriate threshold for the segmenter is considerably lower than for detection and tracking. The information provided by the tracker allows us to automatically set this threshold.

The cost of making a mistake in the segmenter is higher than in the tracker — the segmenter gets only one look at the target, and we want an accurate estimate of the boundaries — so the user-selectable threshold parameter λ must be different as well.

3. The Tracker as an Aid to Classification

The tracker can be used to aid classification in several ways. The first is in a screening role. The tracker provides information about the gross size and behavior of targets that can be used to eliminate certain candidates from consideration. For example, in one security application, nuisance alarms are caused by birds flying by the camera. Distant birds are easily eliminated by their size. Birds near the camera appear large, but their apparent speed is too great for them to be mistaken for humans. Thus, the classifier can discard them on the basis of their velocity.

The tracker also provides supplemental information that can be useful for classifying. In one application, the objects of interest are vehicles seen from the roadside. Each of these objects has only two stable positions: it faces either left or right. In most cases, the vehicles travel forward. If we know which way a vehicle is heading, we can guess which end is the front. The tracker gives us this information with high reliability. Thus, we can map left facing and right-facing vehicles into the same feature space. If the features are geometric moments, this is accomplished by simply reversing the sign of the odd moments. This has two advantages: (1) it reduces the classification error by increasing the separation between classes, and (2) it reduces the size of the feature library, since we don't need to include samples of both left- and right-facing targets.

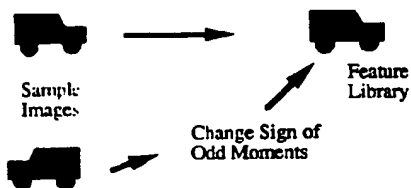


Figure 4: Using Direction of Motion to Map Left and Right-facing Vehicles into the Same Point in Feature Space

Finally, the size of the bounding box can be used to automatically scale the resolution of the images before features are extracted. This is useful because many classification schemes are sensitive to quantization effects and target resolution.

4. Conclusions

Combining the tracker with the segmenter and the classifier has been shown to have several benefits. One is that it allows us to ensure that image noise estimates are not contaminated by targets. The high confidence in the target location provided by the tracker allows us to estimate the *a priori* probability that a given pixel within the target bounding box contains a target; this helps us to automatically set decision thresholds within the bounding box. It also allows us to concentrate processing resources within the box, where there is a very high probability that a target exists. The size and motion information provided by the tracker can help us to eliminate unlikely targets. We can use the direction of motion to simplify the search for matching targets and increase the separability between target classes. Finally, we can use the size of the bounding box to automatically scale segmentation resolution.

References

- [1] Darnell, R.A. "TOST" Video Dimensional Analysis System", Sandia National Laboratories Report, January 20, 1987.
- [2] Darnell, R.A. and C.S. Sonnier, "Semi-Automated Reviewing Station for IAEA** Optical Surveillance Data," Proc. INMM† Annual Meeting, Vol. XVI, pp. 579-584, July 1987.
- [3] Donohoe, G.W., D.R. Hush and N. Ahmed, *Change Detection for Target Detection and Classification in Video Sequences*, Proc. ICASSP-88, New York, April, 1988.
- [4] Huang, T.S. Ed., *Image Sequence Analysis*, Berlin: Springer-Verlag, 1981.
- [5] Hush, D.R., and G.W. Donohoe, Object Tracking in the 2D Image Plane, *Proc. 21st Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, November, 1987.
- [6] Melsa, J.L., and D.L. Cohn, *Decision and Estimation Theory*, New York: McGraw-Hill, 1987.
- [7] Pritchard, D.A., "Application of the 2D-VMD Algorithm to VISDTA: Video Imaging System for Detection, Tracking and Assessment", Sandia National Laboratories internal memo, March 3, 1988.

* Technical On-Site Inspection

** International Atomic Energy Agency

† Institute for Nuclear Materials Management

[8] Teague, M. R., "Image Analysis Via the General Theory of Moments," *J. Opt. Soc. Am.*, Vol 70, No. 8, August 1980.

[9] Fukunaga, K., "Statistical Pattern Classification." in *Handbook of Pattern Recognition and Image Processing*, T.Y. Young and K.S. Fu, Eds., San Diego: Academic Press, 1986.

AN ELECTRONIC VISION SYSTEM FOR SORTING COTTON BOLL WORM PUPAE BY SEX

Jay Jordan and Joe Ellington
New Mexico State University
Las Cruces, NM

John McCoy
USDA/ARS
Mississippi State, MS

ABSTRACT

The cotton boll worm *Heliothis zea* and the tobacco bud worm *Heliothis virescens* are becoming increasingly resistant to pyrethroid based insecticides. Rather than using more powerful chemicals to control these pests, an environmentally safe biological control system is being developed. This system, based on the mass release of sterile male insects, requires the sorting of large quantities of boll worm pupae by sex. This paper describes the specification and design of a computer vision system developed for this task. The system presented represents a transfer of technology from the military to the agricultural sector as it employs multiple sensors, feature extraction techniques, and performance measurement criteria initially developed for military computer vision systems.

INTRODUCTION

Mass releases of sterile insects have been an effective pest management strategy for medfly, *Ceratitis capitata*, and screwworm fly, *Cochliomyia americana* infestations. Applying this technique to the cotton boll worm and tobacco bud worm is attractive, but difficult as female insects of the *Heliothis* genus (Lepidoptera: Noctuidae) require significantly more radiation for sterilization than do males. Male insects subjected to the radiation dose required to sterilize females become inactive. This means that the insects must be sorted by sex prior to sterilization. As a mass release may involve as many as 50000 insects per acre per day during a period of heavy infestation, this tedious task of insect sexing must be automated to achieve a practical biological control system.

In the larval stage, *H. zea* and *H. virescens* caterpillars are extremely active and impractical to handle with automatic machinery. Sex differentiating features are not well developed. In the adult stage, the male and female moths are slightly different colored and relatively easy to distinguish, however, they are very active and too fragile to handle. Sorting by sex must be performed during the pupal stage of development when the insect is relatively inactive and can be handled by automatic machinery. Figure 1 shows an overview of the system.

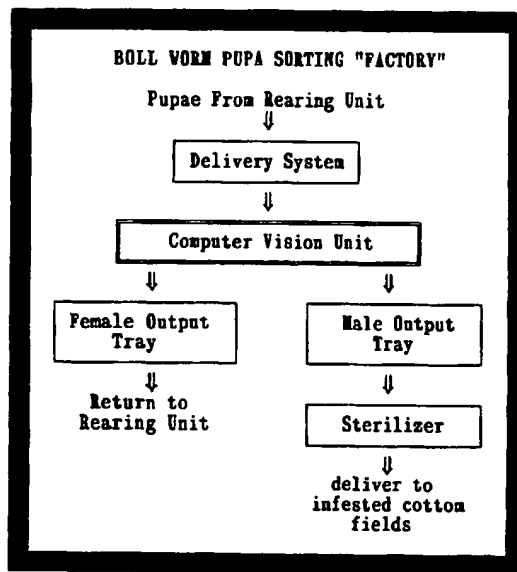


Figure 1.
Sterile Male Biological Control System Overview

COMPUTER VISION SYSTEM DESIGN

The initial approach to the design of the boll worm sorting computer vision system was obtained by observing the technique used by entomology technicians with expert level ability in sorting boll worm pupa by sex. The human experts identify the male pupa by detecting the presence of two pads midway between the last abdominal segment ring and the tip of the abdomen on the ventral side of the pupa. These features are shown in Figure 2. The female insect is identified by the presence of V-shaped sutures (grooves) midway between the last abdominal segment ring and the abdomen tip on the ventral side of the pupa and by the presence of a small notch perpendicular to and almost touching the last abdominal segment ring.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

28524 7590 06/13/2008

SIEMENS CORPORATION
INTELLECTUAL PROPERTY DEPARTMENT
170 WOOD AVENUE SOUTH
ISELIN, NJ 08830

EXAMINER

LU, TOM Y

ART UNIT

PAPER NUMBER

2624

DATE MAILED: 06/13/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

11/135,210

05/23/2005

Leonid Sigal

2004P09036US01

3651

TITLE OF INVENTION: GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1440	\$300	\$0	\$1740	09/15/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

**Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

28524 7590 06/13/2008

SIEMENS CORPORATION
INTELLECTUAL PROPERTY DEPARTMENT
170 WOOD AVENUE SOUTH
ISELIN, NJ 08830

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

11/135,210 05/23/2005 Leonid Sigal 2004P09036US01 3651

TITLE OF INVENTION: GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
-------------	--------------	---------------	---------------------	----------------------	------------------	----------

nonprovisional NO \$1440 \$300 \$0 \$1740 09/15/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
----------	----------	----------------

LU, TOM Y 2624 382-103000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____
- (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____
- 3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

- ☐ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies _____

4b. Payment of Fee(s); (Please first reapply any previously paid issue fee shown above)

- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)

- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____

Date _____

Typed or printed name _____

Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

11/135,210

05/23/2005

Leonid Sigal

2004P09036US01

3651

28524

7590

06/13/2008

EXAMINER

LU, TOM Y

ART UNIT

PAPER NUMBER

2624

DATE MAILED: 06/13/2008

SIEMENS CORPORATION
INTELLECTUAL PROPERTY DEPARTMENT
170 WOOD AVENUE SOUTH
ISELIN, NJ 08830

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 691 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 691 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability

Application No.

11/135,210

Examiner

Tom Y. Lu

Applicant(s)

SIGAL ET AL.

Art Unit

2624

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to application filed 05/23/2005.
2. ☒ The allowed claim(s) is/are 1-16.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).
- * Certified copies not received: ____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
- (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
- 1) ☐ hereto or 2) ☐ to Paper No./Mail Date ____.
- (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).**
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date See Continuation Sheet
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413),
Paper No./Mail Date ____.
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other ____.

Continuation of Attachment(s) 3. Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date: 05/23/2005; 10/24/2005.

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

The application has been amended as follows:

Claim 9:

Replace "A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for object detection, the method steps comprising:" with "A computer readable medium embodying a computer program to perform method steps for object detection, the method steps comprising:"

Information Disclosure Statement

2. The information disclosure statements (IDS) submitted on 05/23/2005 and 10/24/2005 have been considered by the examiner.

Allowable Subject Matter

3. Claims 1-16 are allowed.

The following is an examiner's statement of reasons for allowance:

Independent claims 1 and 9 both define features of providing a spatio-temporal model for an object to be detected; determining a probability that the object is in each image; and detecting the object in any image upon comparing the probabilities for each image to a threshold for detecting the object. These features in combination with other features in claims 1 and 9, which are the broadest allowable claims, are not taught or suggested by the art of record.

The close prior art, U.S. Patent No. 6,597,801 B1, Cham et al, teaches providing a kinematic model for an object to be detected (column 11, line 26), providing a video comprising a plurality of images including the object (figure 4B); measuring the object as a collection of components in each image (figure 10A); performing feature search operations for object features by comparing a total probability with a predetermined threshold (column 3, lines 64-65). However, Cham does not teach the claimed “spatio-temporal model”, which is different from the kinematic model (spatial model).

The another close prior art, U.S. Patent No. 7,200,266 B2, Ozer et al, teaches a method to detect objects in multiple video frames through object modeling and graph matching, where the output of the graph-matching is the classified object parts, the movements of the object parts are a spatio-temporal sequence of feature vectors that consists of the direction of the object part movement. Additionally, Ozer's system checks direction of the movements of the object parts for a number of frames and calculates the probabilities of the actives with the known activities by using Hidden Markov Models and choose the pattern with the highest probability as the recognized activity in these frames. However, Ozer does not teach the claimed features of determining a probability that the object is in each image; and detecting the object in any image upon comparing the probabilities for each image to a threshold for detecting the object.

Additionally, there is no suggestion or motivation to combine the Cham and Ozer references because the object detection algorithms in two references are vastly different.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

Art Unit: 2624

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tom Y. Lu whose telephone number is (571) 272-7393. The examiner can normally be reached on 8:30AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Bella can be reached on (571)-272-7778. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tom Y Lu/

Examiner, Art Unit 2624

Notice of References Cited	Application/Control No. 11/135,210		Applicant(s)/Patent Under Reexamination SIGAL ET AL.	
	Examiner Tom Y. Lu		Art Unit 2624	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,200,266	04-2007	Ozer et al.	382/173
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

<i>Index of Claims</i> 	Application/Control No. 11135210	Applicant(s)/Patent Under Reexamination SIGAL ET AL.
	Examiner Tom Y Lu	Art Unit 2624

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47										
CLAIM		DATE								
Final	Original	06/08/2008								
	1	=								
	2	=								
	3	=								
	4	=								
	5	=								
	6	=								
	7	=								
	8	=								
	9	=								
	10	=								
	11	=								
	12	=								
	13	=								
	14	=								
	15	=								
	16	=								

Search Notes 	Application/Control No. 11135210	Applicant(s)/Patent Under Reexamination SIGAL ET AL.
	Examiner Tom Y Lu	Art Unit 2624

SEARCHED			
Class	Subclass	Date	Examiner
382	103, 173,	6/7/2008	TYL

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search printout attached	6/7/2008	TYL

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
	See printout attached	6/7/2008	TYL



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 3651

SERIAL NUMBER 11/135,210	FILING or 371(c) DATE 05/23/2005 RULE	CLASS 382	GROUP ART UNIT 2624	ATTORNEY DOCKET NO. 2004P09036US01		
APPLICANTS Leonid Sigal, Providence, RI; Ying Zhu, Monmouth Junction, NJ; Dorin Comaniciu, Princeton Jct., NJ; ** CONTINUING DATA ***** This appln claims benefit of 60/574,799 05/27/2004 ** FOREIGN APPLICATIONS ***** ** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 06/20/2005						
Foreign Priority claimed ☐ Yes ☒ No 35 USC 119(a-d) conditions met ☐ Yes ☒ No Verified and Acknowledged <u>/TOM Y LU/</u> Examiner's Signature		☐ Met after Allowance TYL Initials	STATE OR COUNTRY RI	SHEETS DRAWINGS 5	TOTAL CLAIMS 16	INDEPENDENT CLAIMS 2
ADDRESS SIEMENS CORPORATION INTELLECTUAL PROPERTY DEPARTMENT 170 WOOD AVENUE SOUTH ISELIN, NJ 08830 UNITED STATES						
TITLE Graphical object models for detection and tracking						
FILING FEE RECEIVED 1130	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:			☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit		

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	9	spatio adj temporal same (video or frame) same (probability or statistic) same (object or subject) near2 (track\$3 or detect\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/05 14:51
S2	171	lu-tom-y.xa.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/05 14:56
S3	4	joint near probability same (spatio adj temporal)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/05 14:58
S4	1	S2 and joint near probability	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/05 15:00
S5	2	"6597801".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/05 21:49

S6	9	spatio same temporal same (video or frame) same (probability or statistic) same (object or subject) near2 (track\$3 or detect\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 12:13
S7	9	spatio same temporal same (video or frame) same (probability or statistic) same (object or subject) near2 (track\$3 or detect\$3 or segment\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 12:22
S8	171	lu-tom-y.xa.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 12:33
S9	1	S8 and (spatio adj temporal)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 12:33
S10	368	spatio adj temporal and (video or frame) and (probability or statistic) and (object or subject) near2 (track\$3 or detect\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 13:25

S11	0	S8 and S10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 14:06
S12	27	S10 and 382/103.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/07 14:06
S13	4	(spatio adj temporal and (video or frame) and (probability or statistic) and (object or subject) near2 (track\$3 or detect\$3)).clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/08 15:55

6/ 8/ 08 3:57:11 PM

C:\ Documents and Settings\ TLu\ My Documents\ EAST\ Workspaces\ 11135210.wsp

Express Mail Label No.: EV336791321US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Sigal et al.

Docket: 2004P09036US01 (8706-817)

Serial No.: unassigned

Date: May 23, 2005

Filed: concurrently herewith

For: **BOOSTED GRAPHICAL MODELS FOR OBJECT DETECTION AND TRACKING**

Commissioner for Patents
PO Box 1450
Alexandria VA 20231-1450

INFORMATION DISCLOSURE STATEMENT

Sir:

Pursuant to Applicant(s) duty of disclosure, the information listed in the attached form PTO-1449 is brought to the attention of the Examiner. Copies of the listed items are enclosed.

The citation of the listed items is not a representation that they constitute a complete or exhaustive listing of the relevant art or that the references are prior art. The items listed are submitted in good faith, but are not intended to substitute for the Examiner's search. It is hoped, however, that in addition to apprising the Examiner of these particular items, they will assist in identifying fields of search and in making as full and complete a search as possible.

The filing of this information disclosure statement is not an admission that the information cited herein is, or is considered to be, material to patentability as defined in 37 C.F.R. § 1.56(b).

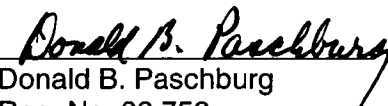
- ☒ This information disclosure statement is being filed within three (3) months of the filing date of this application.
- ☐ This information disclosure statement is being filed within three (3) months of the date of entry of the national stage as set forth in 37 C.F.R. § 1.491 in an international application.
- ☐ To the best of Applicant(s) knowledge, this information disclosure statement is being filed before the date of mailing of a first Office Action on the merits in connection with this case.
- ☐ Statement under 37 C.F.R. § 1.97(e):
I hereby state that each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than 3 months prior to the filing of this statement.
- ☐ Enclosed herewith is a petition under 37 C.F.R. § 1.97(d)(ii).
 - ☐ Enclosed by check is the petition fee of \$130.00. (37 C.F.R. § 1.17(i)(1))
 - ☐ Please charge the fee of \$130.00 for petition fee to deposit account 19-2179.
- ☐ Enclosed by check is the \$180.00 fee required by 37 C.F.R. § 1.17(p).
- ☐ Please charge the fee of \$180.00 fee required by 37 C.F.R. § 1.17(p) by enclosed herewith.

[X] Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17 at any time during the pendency of this application, or credit any overpayment of such fee(s) to Deposit Account 19-2179. Also, in the event any extensions of time for responding are required for the pending application(s), please treat this paper as a petition to extend the time as required and charge Deposit Account No. 19-2179 therefor. A COPY OF THIS SHEET IS ENCLOSED.

The claims of the application as now presented are believed to patentably distinguish over the prior art and to be in condition for allowance. Early and favorable consideration of the case is respectfully requested.

Respectfully submitted,

By:


Donald B. Paschburg
Reg. No. 33,753
Attorney for Applicant(s)

Mailing Address:

Siemens Corporation
Intellectual Property Dept, 5TH Floor
170 Wood Avenue South
Iselin, New Jersey 08830
(732) 321-3191
(732) 321-3030 (FAX)

Form PTO-1449		U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE		ATTY. DOCKET NO. 2004P09036US01 (8706-817)		SERIAL NO. unassigned	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary)				APPLICANT Sigal et al.			
				FILING DATE concurrently here with		GROUP ART UNIT	

U.S. PATENT DOCUMENTS							
EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS								
		PUBLICATION NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	ABSTRACT TRANSLATION	
							YES	NO

OTHER PRIOR ART (Including Author, Title, Date, Pertinent Pages, Etc.)		
/TYL/		M. Isard. PAMPAS: Real-valued graphical models for computer vision, CVPR, Vol. 1, pp. 613-620, 2003.
/TYL/		K. Okuma, A. Teleghani, N. de Freitas, J. Little and D. Lowe. A boosted particle filter: Multi-target detection and tracking, ECCV, 2004.
/TYL/		L. Sigal, S. Bhatia, S. Roth, M. Black and M. Isard. Tracking loose- limbed people, CVPR, 2004.
/TYL/		E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, MIT Laboratory for Information and Decision Systems Technical Report P-2551 (October 15, 2002).
/TYL/		P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, CVPR, 2001.
/TYL/		P. Viola, M. Jones and D. Snow. Detecting pedestrians using patterns of motion and appearance, ICCV, pp. 734-741, 2003.

EXAMINER /Tom Y. Lu/ (06/05/2008)	DATE CONSIDERED (06/05/2008)
---------------------------------------	----------------------------------

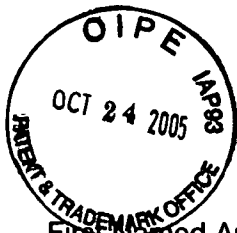
* EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

(Form PTO-1449 [6-4])

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TYL/

Meta 1002

p. 176



1FW
PATENT

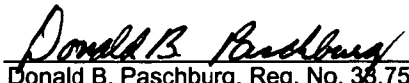
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Sigal
Serial No.: 11/135,210
Filed: May 23, 2005
Title: GRAPHICAL OBJECT MODELS FOR DETECTION
AND TRACKING

Docket No.: 2004P09036 US01
Examiner: Not yet assigned
Group Art Unit: 2621
Date: October 20, 2005

CERTIFICATE OF MAILING

I hereby certify that correspondence is being deposited on October 20, 2005 with the U.S. Postal Service with sufficient postage as first class mail on the below-indicated date in an envelope addressed to:
Mail Stop DD, Commissioner for Patents, PO Box 1450, Alexandria,
VA 22313-1450


Donald B. Paschburg, Reg. No. 33,753

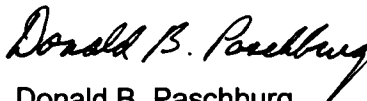
Mail Stop DD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Sir:

The documents listed on the accompanying Form PTO/SB/08a, and of which copies are enclosed, are hereby cited pursuant to 37 CFR §1.56, §1.97 and §1.98, for consideration in the examination of the above-identified application and for the purpose of having them made of record. These documents were cited in an International Search Report (copy enclosed) in a related application.

Respectfully submitted,


Donald B. Paschburg
Reg. No.: 33,753

Siemens Corporation
Intellectual Property Department
186 Wood Avenue South
Iselin, NJ 08830
(732) 321-3191

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TYL/

Substitute for form 1449A/PTO

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**

OCT 24 2005

(use as many sheets as necessary)

Complete if Known

Application Number	11/135,210
Filing Date	May 23, 2005
First Named Inventor	Leonid Sigal
Group Art Unit	2621
Examiner Name	Not yet assigned
Attorney Docket Number	2004P09036 US01

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials *	Cite No. 1	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²
/TYL/	3	Lan et al., "A unified spatio-temporal articulated model for tracking", Computer Vision and Pattern Recognition, 2004, Proceedings of the 2004 IEEE Computer Society Conference on Washington, DC, 27 June – 2 July 2004, Piscataway, NJ, Vol. 1, pp. 722-729.	
/TYL/	4	Donohoe, "Combining Segmentation and Tracking for the Classification of Moving Objects in Video Scenes", Conference Record, 22 nd Asilomar Conference on Signals, Systems and Computers (Cat. No. 88CH2835-7), Maple Press San Jose, CA, USA, Vol. 2, 31 October 1988, pp. 533-538.	

Examiner Signature	/Tom Y. Lu/ (06/05/2008)	Date Considered	(06/05/2008)
-----------------------	--------------------------	--------------------	--------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 120 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TYL/

PART B – FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Customer Number: 028524

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, NJ 08830

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571)273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAME INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/135,210	05.23.05	Leonid Sigal	2004P09036 US01	3651

TITLE OF INVENTION:

GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
Nonprovisional	NO	\$ 1440.00	\$ 300.00	\$1740.00	09.15.08
EXAMINER		ART UNIT	CLASS-SUBCLASS		
LU, TOM Y		2624	382-103000		

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363)

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB122) attached

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47, Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

1 Donald B. Paschburg

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

2 F. Chau & Associates, LLC

3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

Siemens Corporate Research, Inc.

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Princeton, NJ

Please check the appropriate assignee category or categories (will not be printed on the patent):

☐ Individual

☒ Corporation or other private group entity

☐ Government

4a. The following fee(s) are submitted:

☒ Issue Fee

☒ Publication Fee (No small entity discount permitted)

☐ Advance Order # of Copies _____

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

☐ A check in the amount of the fee(s) is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☒ The Director is hereby authorized to charge the required fee(s), or credit any overpayment, to Deposit Account Number **19-2179** (enclose an extra copy of this form)

5. Change in Entity Status (from status indicated above)

☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2)

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature Donald B. Paschburg

Date 09.02.08

Typed or printed name Donald B. Paschburg

Registration No. 33,753

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Electronic Patent Application Fee Transmittal

Application Number:	11135210			
Filing Date:	23-May-2005			
Title of Invention:	GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING			
First Named Inventor/Applicant Name:	Leonid Sigal			
Filer:	Donald Bruce Paschburg/v richman			
Attorney Docket Number:	2004P09036US01			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1440	1440
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1740

Electronic Acknowledgement Receipt

EFS ID:	3870394
Application Number:	11135210
International Application Number:	
Confirmation Number:	3651
Title of Invention:	GRAPHICAL OBJECT MODELS FOR DETECTION AND TRACKING
First Named Inventor/Applicant Name:	Leonid Sigal
Customer Number:	28524
Filer:	Donald Bruce Paschburg/v richman
Filer Authorized By:	Donald Bruce Paschburg
Attorney Docket Number:	2004P09036US01
Receipt Date:	02-SEP-2008
Filing Date:	23-MAY-2005
Time Stamp:	11:13:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$1740
RAM confirmation Number	6736
Deposit Account	192179
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	2004p09036us01-issuefee.pdf	69737	no	1
			102275845f3558e9b179a91477069d3dc0e14f32		

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing

Information:

2	Fee Worksheet (PTO-06)	fee-info.pdf	31931	no	2
			3e97191cddef88ead2efe3b104b55db4fa2d0516		

Warnings:

Information:

Total Files Size (in bytes):			101668
------------------------------	--	--	--------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/135,210	10/14/2008	7436980	2004P09036US01	3651

28524 7590 09/24/2008

SIEMENS CORPORATION
INTELLECTUAL PROPERTY DEPARTMENT
170 WOOD AVENUE SOUTH
ISELIN, NJ 08830

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 691 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Leonid Sigal, Providence, RI;
Ying Zhu, Monmouth Junction, NJ;
Dorin Comaniciu, Princeton Jct., NJ;

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
---	---

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been
filed in the U.S. District Court District of Delaware on the following

☐ Trademarks or ☒ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/24/2022	U.S. DISTRICT COURT District of Delaware
PLAINTIFF VideoLabs, Inc. and VL Collective IP LLC		DEFENDANT Meta Platforms, Inc., Instagram, Inc., WhatsApp LLC, Facebook Technologies, LLC, and Giphy, Inc.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 8,139,878 B2	3/20/2012	VideoLabs, Inc.
2 US 7,769,238 B2	8/3/2010	VideoLabs, Inc.
3 US 7,970,059 B2	6/28/2011	VideoLabs, Inc.
4 US 7,266,682 B2	9/4/2007	VL Collective IP LLC
5 US 7,436,980	10/14/2008	VL Collective IP LLC

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE
-------	-------------------	------

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy