

This Page Is Inserted by IFW Operations
and is not a 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 may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

Express Mail Label No. EV335644049US

2004P09036US

Date of Deposit: 27.May.2004

PROVISIONAL PATENT APPLICATION

NAME OF INVENTOR(S):

Leonid Sigal
8 Daniel Ct.
Woburn, MA 01801
Citizen: USA

Ying Zhu
3014 Fox Run Drive
Plainsboro, NJ 08536
Citizen: PR of China

Dorin Comaniciu
2 Stuart In. West
Princeton Jct., NJ 08550
Citizen: Romania

TITLE OF INVENTION:

Boosted Graphical Models For Object Detection And Tracking

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

Boosted Graphical Models for Object Detection and Tracking

Leonid Sigal

Department of Computer Science
Brown University
Providence, RI 02912
ls@cs.brown.edu

Ying Zhu

Siemens Corporate Research
Princeton, NJ 08540
ying.zhu@siemens.com

Dorin Comaniciu

Siemens Corporate Research
Princeton, NJ 08540
dorin.comaniciu@siemens.com

Abstract

We present a probabilistic framework for component-based automatic detection and tracking of objects in images and/or video. We represent objects as spatio-temporal two-layer graphical models, where each node corresponds to an object or component of an object at a given time, and the edges correspond to the learned spatial and temporal constraints. Object detection and tracking is then formulated as an inference over directed loopy graph, and is solved with non-parametric belief propagation that uses a variation of particle filtering. This type of an object model allows object-detection to make use of temporal consistency (over an arbitrary size temporal window), and provides mechanism for simultaneous robust tracking of the object. The two layer structure of the graphical model allows inference over the entire object as well as individual components. AdaBoost detectors are used to define the likelihood and form proposal distributions for components. Proposal distributions provide 'bottom-up' information that is incorporated into the inference process; this permits automatic object detection and aids recovery from transient tracking failures. We illustrate our method by detecting and tracking two classes of objects, cars and pedestrians, in video sequences collected using a single uncalibrated car mounted camera.

1 Introduction

We present a probabilistic framework for automatic component-based detection and tracking of objects in images and/or video. By combining the object detection with tracking in a unified framework we can achieve a more robust solution for both problems. Tracking can make use of object detection for initialization and re-initialization during transient failures or occlusions, while object detection can be made more reliable by considering the consistency of the detection over time. Component-based detection facilitates detection of

complex articulated objects, as well as helps in handling partial object occlusions or local illumination changes.

Object detection and tracking are formulated as inference in a two-layer graphical model in which object components are connected via learned spatial and temporal probabilistic constraints. Each node in a graphical model corresponds to a position and scale of the component (for the first layer) or an object as a whole (for the second layer) in an image at a given time instant. Each node also has an associated AdaBoost detector that serves as a basis for the local image likelihood and the proposal process. Directed edges between nodes represent statistical dependencies between components, and are learned using standard machine learning methods.

Since in general one cannot guarantee that these dependencies are Gaussian, and one would like to recover a solution in a continuous three-dimensional parameter space, we exploit a form of non-parametric belief propagation that uses a variation of particle filtering and can be applied over a loopy graph [19, 21]. There are at least two reasons that can lead to non-Gaussianity of component dependencies: (1) if there is indeed more than one mode for the statistical relationship between components (e.g. if we are looking for a keyboard and are using button 'Esc' as one of the components, then depending on the keyboard manufacturer there may be a number of positions that 'Esc' can be in relative to top left corner of the keyboard) (2) if one wants to reuse parts for efficiency (e.g. for a side view of a car we can model both tires using a single 'tire' component and then have a bi-modal spatial distribution for a car position, where bi-modality will get resolved by other constraints).

Our framework is a harness build on top of the standard object detection algorithm (AdaBoost) that provides facilities for component-based detection, temporal information integration and simultaneous tracking.

1.1 Related Work

The problem of describing and recognizing categories of objects (e.g. faces, people, cars) has been an active field of computer vision for a considerable time. A seemingly global agreement has been reached on the representation of object categories, with most researchers following the trend of representing objects as collections of features with distinctive appearance, spatial extent, and position [4, 5, 7, 6, 9, 10, 11]. There is however a large discrepancy on the details of how many features one must use, how these features are detected and represented. Most algorithms however rely on semi-supervised learning [4, 5, 7, 6] schemes where examples of the desired class of objects must be manually aligned, and then the learning algorithms are used to automatically select the features that best separate the images of the desired class from background image patches. More recent approaches attempt to get rid of the tedious manual alignment and learn the model automatically from the set of unlabeled and unsegmented images [9, 11]. Other algorithms [10] attempted to reduce the number of examples required to learn the classifier by incorporating 'generic' knowledge which can be obtained from previously learned models of unrelated categories. Overall object detection can be categorized into three families of approaches: part-based [4, 10, 11] (object is defined as an arrangement of smaller parts modeled either implicitly or explicitly), patch-based [24, 17] (each rectangular image region is classified as object or background), and region-based [27] (region of an image is segmented from the background and classified). In this paper we concentrate on improving the current state of the art in part/component-based approaches.

Among the earliest part-based approaches, was the work of [4, 5], where classes of objects were described using a set of 2D wavelet features selected from the overcomplete base set. The set of features was selected manually by looking at the response of features over the positive examples. The detection is then achieved by training an SVM classifier that is run exhaustively over a number of image scales. Viola and Jones [7] developed a learn-

ing algorithm based on AdaBoost that automatically selected the features from the set of overcomplete 2D Haar wavelet-like basis. A number of improvements that allowed fast evaluation of features, and fast punning of large regions of the image that did not contain object-like regions were presented. In the more recent work [6] showed that motion information can significantly improve performance by reducing the number of false positives. The motion information however is restricted to two consecutive frames.

A number of object detection [4, 8, 16] algorithms suggested that modeling complex objects by components and then combining [4, 16] or statistically fusing [8] the information is superior to the global appearance approaches in the presence of partial occlusions, out-of-plane rotation and/or local lighting variations. Component-based detection is also capable of handling highly articulated objects, for which a single appearance model is hard to learn. One example of the scenario where components prove to be quite useful, is vehicle detection. While all vehicles have almost identical parts (tires, bumper, hood, etc.) they can vary significantly in their placement due to large variability in the heights of the vehicles (see Fig. 1). Hence learning a single appearance model for all vehicles is hard, and would require the model to implicitly encode the variability in the position of components by using many similar features at various heights.

Here we present a novel approach to object detection and tracking that builds on top of, improves and extends the standard AdaBoost [7]. We exploit benefits of component-based detection and temporal integration to improve the performance of the simple single appearance object detector. We model objects using two-layer spatio-temporal graphical model, and use non-parametric belief propagation to infer the position and scale of the object and components. We should note that the use of graphical models and belief propagation is not novel for object detection. Murphy *et al* [24], used graphical models in the patch-based detection scheme, that unlike our approach did not incorporate temporal information or explicitly reasoned about the object as a whole. Coughland and Ferreira [13] used belief propagation and loopy graphical models for the deformable template matching illustrated on character recognition task. Unlike all previous approaches however we provide a automatic probabilistic framework that explicitly models objects using components, integrates temporal information, and allows for simultaneous tracking. The closest in spirit comes the work of [3] that uses AdaBoost for multitarget tracking and detection. Unlike our approach however their system does not integrate component-based object detection, and is limited to temporal propagation in only one direction (forward in time).



Figure 1: Variation in the vehicle class of objects is shown. While objects shown here have a drastically different appearance as a whole due to the varying height and type of the vehicle, their components tend to be very homogeneous and are easy to model.

2 Graphical Object Models

Following the framework of [1, 2] we model an object as a spatio-temporal directed graphical model. Each node in this graph represents an object or a component of an object at time t . The direct edges correspond to the learned spatial and temporal constraints. Each node has an associated state vector $X^T = (x, y, s)$ defining the component's position and scale within an image (where $x, y, s \in \mathbb{R}$). Finding and tracking of an object is formulated as an inference over this graphical model.

Each directed edge between components i and j has an associated potential function $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ that encodes the compatibility between pairs of node states. The potential $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ is modeled using a mixture of M_{ij} Gaussians (following [2]).

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

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. δ_{ijm} is the relative weight of an individual component and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$.

The framework presented here can be viewed as having five distinct components: (i) a graphical model, (ii) an inference algorithm that provides ability to infer a state of each node in a graph, (iii) a local evidence distribution (or image likelihood), (iv) a proposal process for some or all nodes in a graphical model, and a set of (v) spatial and/or temporal constraints corresponding to the edges in a graph. We will now discuss each one of these in turn.

2.1 Building a Graphical Model

In a single frame we choose to represent an objects using a two-layer spatial graphical model. The first layer corresponds to components of the object and is loosely connected (while maintaining connectivity, we attempt to keep the degree of each node relatively small for efficiency). The second layer corresponds to an entire appearance model and is connected to all constituent components. Examples of such models that we use for vehicle and pedestrian detection are shown in Fig. 2a and 2b respectively. We can extend this model over time to an arbitrary length spatio-temporal graphical model as shown in Fig. 2c and 2d. Unlike the model presented in [2] having a second-layer also allows us to assume conditional independence of components over time given the whole appearance, hence reducing the complexity of the overall graphical model.

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

Since we are not guaranteed convergence in the loopy-graphical models, in general one must be careful to avoid models that are inherently unstable. Fig. 2e shows a graphical model for which a unique solution may not be found. For example, nodes 1, 3, and 5 may settle on one distribution while nodes 2 and 4 settle on another. If this happens then no number of message passing iterations will result in convergence. The cliques of the graph will simply flip from one solution to the other.

2.2 Learning Spatial and Temporal Constraints

Given a set of labeled images, with each component associated with the reference point, we use standard iterative Expectation-Maximization (EM) algorithm with K-means initialization to learn $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ of the form:

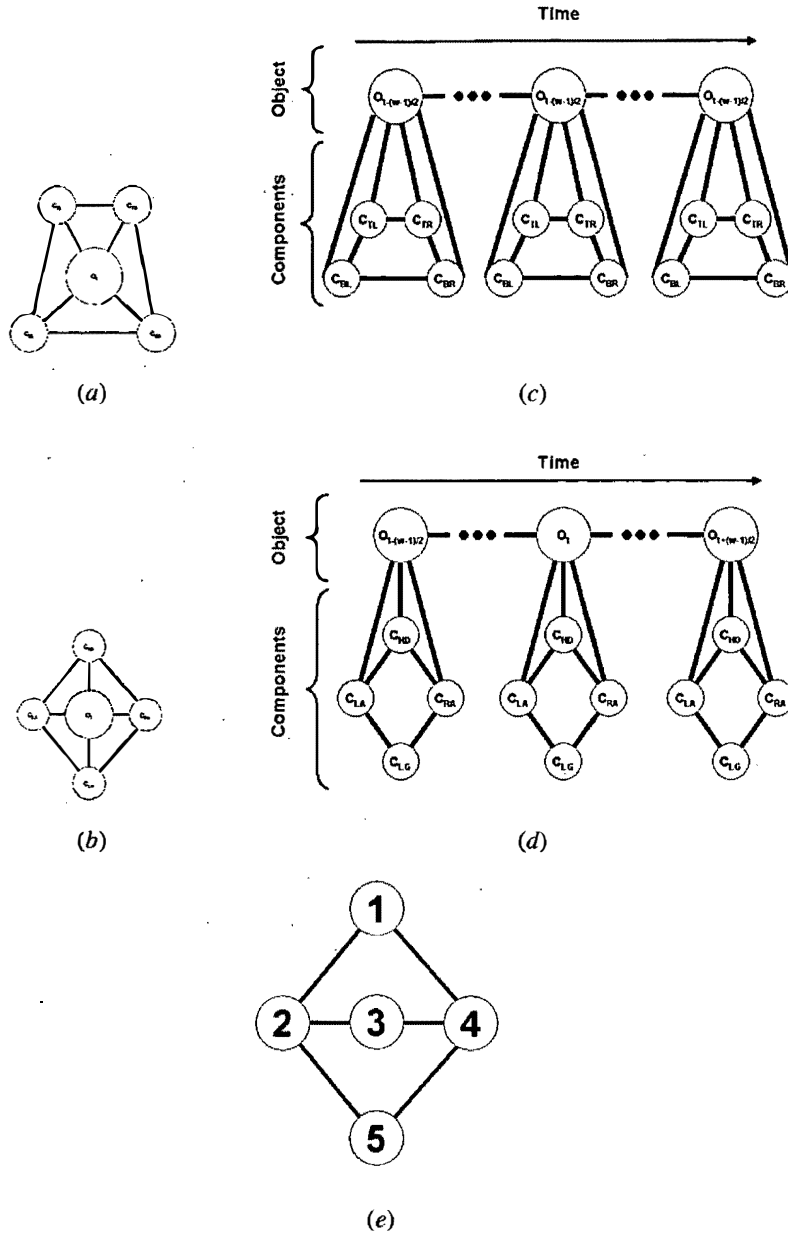


Figure 2: Spatial graphical models for the vehicle (a) and pedestrian (b), plus their spatio-temporal counterparts (c) and (d) respectively. Spatio-temporal models are obtained by replicating a spatial model along the temporal domain to a w -length window and then connecting the second-layer nodes across time.

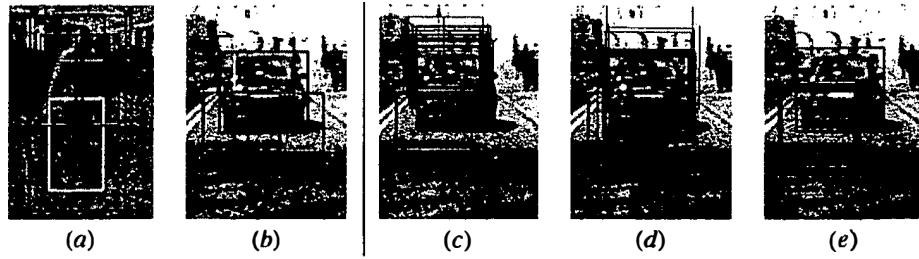


Figure 3: Components for a (a) pedestrian and (b) vehicle object models (entire appearance model is in black) and learned conditional distributions from (c) Bottom-Left (BL) to Top-Left (TL) component, (d) Bottom-Left (BL) to the whole appearance model, and (e) vice versa for a vehicle object.

$$F_{ijm}(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{X}_i + \left[\frac{\mu_{x,ijm}}{\mu_{s,ijm}}, \frac{\mu_{y,ijm}}{\mu_{s,ijm}}, \mu_{s,ijm} \right]^T \quad (2)$$

$$G_{ijm}(\mathbf{X}_i, \mathbf{X}_j) = I(3, 3) \cdot [\sigma_{x,ijm}^2, \sigma_{y,ijm}^2, \sigma_{s,ijm}^2]^T \quad (3)$$

where $\mu_{x,ijm}$, $\mu_{y,ijm}$, $\mu_{s,ijm}$ is the mean position and scale offset between the connected components i and j , and $\sigma_{x,ijm}$, $\sigma_{y,ijm}$, $\sigma_{s,ijm}$ is the standard deviation. Since our training data has only a single reference point for each component, we cannot estimate the spatial scale parameters directly, instead we assume $\mu_{s,ijm} = 0$ and $\sigma_{s,ijm} \approx 0.1$.

While this is not strictly necessary, for the ease of the formulation we assume that the constraints are stationary in time (i.e. the relative position of the component i and j is statistically the same for any given frame in our spatio-temporal window). Examples of the learned conditional distributions can be seen in Fig. 3cde.

2.3 Non-parametric BP

Inferring the state of the object and it's components in our framework is defined as estimating belief in a graphical model. We use a form of non-parametric belief propagation [19] PAMPAS to deal with this task. The approach is a generalization of particle filtering [15] which allows inference over an arbitrary graphs rather than a simple chain. In this generalization the "message" used in a standard belief propagation is approximated with a kernel density (formed by propagating a particle set through a Gaussian or mixture of Gaussians potential density), and the conditional distribution used in standard particle filtering is replaced by product of incoming messages. Most of the complexity in this algorithm lays in sampling from a product of kernel densities required for message passing and belief estimation, we use sequential multiscale Gibbs sampling and an epsilon-exact sampling [22] to alleviate 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 we integrate information from all components in the second-layer node, we can get a very reliable estimate for the overall object size.

More formally a message m_{ij} is written as

$$m_{ij}(\mathbf{X}_j) = \int \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) \phi_i(\mathbf{X}_i) \prod_{k \in A_{ij}} m_{ki}(\mathbf{X}_i) d\mathbf{X}_i, \quad (4)$$

where A_i is the set of neighbors of node i and $\phi_i(\mathbf{X}_i)$ is the local evidence (or likelihood) associated with the node i , and $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ 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 see [1, 2].

We use w -frame windowed smoothing algorithm for our tracking and object detection, where w is an odd integer ≥ 1 (see Fig. 2cd). There are two ways one can do windowed smoothing, what we call an object-detection centric way, and the tracking centric way. In an object-detection centric way we re-initialize all nodes every time we shift a window, hence the temporal integration is only applied in the window of size w . In the tracking centric way we only initialize the nodes associated with a new frame, which tends to force temporal consistency from before $t - (w - 1)/2$. While tracking-centric way tends to converge faster and produce more consistent results over time, it also less sensitive to objects entering and leaving the scene. Note that with $w = 1$, the algorithm resembles single frame component-based fusion [8].

The BP message update equation can be executed either by updating all the messages m_{ij} in a batch or by updating certain messages before others. The later, also called focused message updating [13], can significantly speed up BP given some *apriori* knowledge of the object model. For example, if a second layer node of the graph has no associated likelihood of proposal process (as is in our car model), then it would make sense to update the incoming messages to it before updating the outgoing ones. For generality and ease, we choose not to do focused message updating here.

2.4 AdaBoost Image Likelihoods

The inference algorithm outlined in the previous section requires probabilistic image likelihood model $\phi_i(\mathbf{X}_i)$. This likelihood model is the likelihood of observing the the image conditioned on the state $\mathbf{X}_i$ of the node i , and ideally should be robust to partial occlusions, the variability of image statistics across many different inputs. To that end we build our likelihood model based on the standard AdaBoost object detection.

Following [7] we train boosting based detectors [7] for each component. There are many variants of boosting [7, 6, 25, 26], which differ in the loss function they are trying to optimize, and the approach they take for choosing gradient direction. For simplicity we choose to implement the AdaBoost [7] without cascade. We expect that training a cascade would improve the performance of the system. In order to reduce the number of false positives produced by the detectors, we use bootstrap procedure that iteratively adds false positives that are collected by running the trained strong classifier over the set of background images (not containing desired object) and then re-training the detectors using the old positive and the new extended negative sets.

Given a set of labeled patterns AdaBoost procedure learns a weighted combination of base weak classifiers, $h(I) = \sum_{k=1}^K \alpha_k h_k(I)$, where I is an image pattern, and $h_k(I)$ is the weak classifier chosen for the round t of boosting, and α_k is the corresponding weight. We use a weak classifier scheme similar to the one discussed in [7]: $h_k(I) = p_k \sqrt{(|f_k(I) - \theta_k|)^{\beta_k}}$, where $f_k(I)$ is a feature of the pattern I computed by convolving I with the delta function over an extend of a spatial template; θ_k is a threshold, p_k polarity indicating the direction of inequality, and $\beta_k \in [1, 2]$ allowing for a symmetric two sided pulse classification. For the examples of the templates we use for pedestrian classification, see Fig. 2.4.



The output of the AdaBoost classifier is a confidence $h_k(I)$ that the given pattern I is of the desired class. It is customary to consider object present if $h_k(I) \geq \frac{1}{2} \sum_{k=1}^K \alpha_k$. We convert this confidence rank into a likelihood function by first normalizing the α_k 's, so that $h(I) \in [0, 1]$, and then exponentiating

$$\phi_i(I|\mathbf{X}_i) \propto \exp(h(I)/T) \quad (5)$$

where T is a temperature parameter that controls the smoothness of the likelihood function. With smaller values of T leading to peakier distribution. Consequently we can also anneal the likelihood by deriving a schedule with which T changes. We found exponential annealing schedule $T = T_i v^\kappa$, where T_i is the initial temperature, v is a fraction $\in (0, 1)$, and κ is the annealing iteration, to work well in practice.

Note that while our state is contentious, AdaBoost likelihood can only be evaluated at the discrete pixel positions. To enable sub-pixel accuracy one can use interpolation. In addition, it may be prohibitively expensive to compute the likelihood due to need for scaling imposed by the contentious scale space. In practice we discretized the scale space, and use bi-cubic interpolation to estimate the contentious likelihood function.

2.5 Proposal Process

To form a proposal distribution for a component, we run corresponding AdaBoost detector over an image at a number of scales to produce a set of detection results $\mathfrak{S}$ 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 usually large for non-specific component detectors (a few thousand positions can easily be found running a detector at every pixel for 9 consecutive scales). To reduce the dimensionality of the set we only keep the top P scoring detections, where P is on the order of a 100. This constitutes the proposal distribution, and every time a sample from the proposal is require, one of the resulting detections is uniformly sampled.

3 Experiments

3.1 Vehicle Spatio-Temporal Tracking

We have tested the system on sequences of moving vehicles collected using a mounted moving camera. Fig. 4 shows the results for the 55 frame long sequence. We use a 3-frame spatio-temporal model that is shifted in a tracking centric (i.e. only the new frame is re-initialized using the proposal when the window is shifted) way through time. The results are very encouraging. One can see a very tight distribution of samples concentrated at the correct location and scale of an image.

3.2 Pedestrian Spatio-Temporal Tracking

We have also tested the system on sequences of moving pedestrians. Fig. 5 shows the results for the single instant pedestrian detection. We use a 3-frame spatio-temporal model.

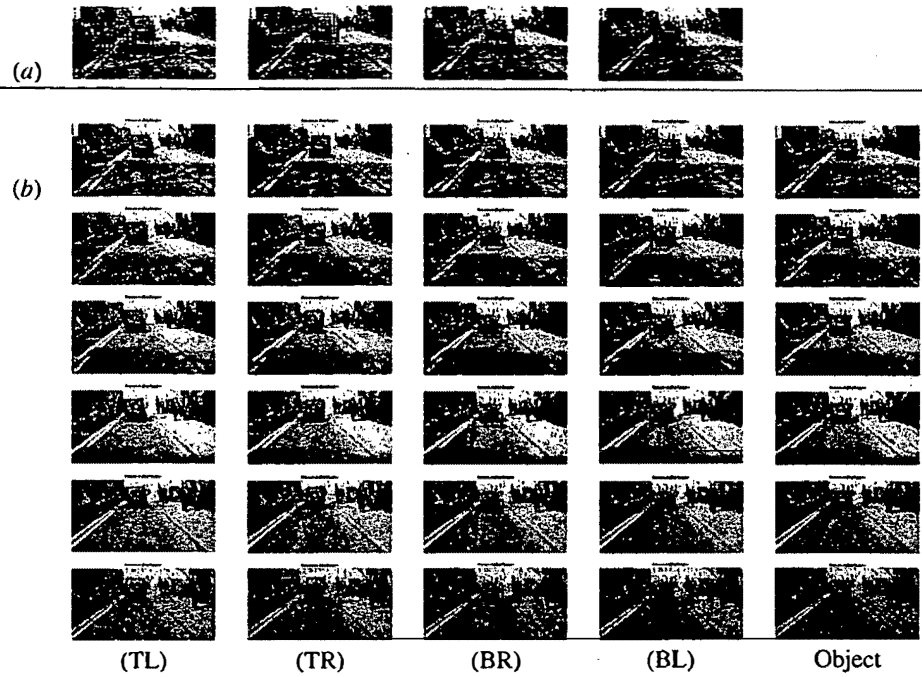


Figure 4: Vehicle object detection and tracking. (a) shows the initialization/proposal distribution, and (b) 30 samples taken from the belief for each of the four component and an object. The detection and tracking was conducted using a 3-frame smoothing window. Frames 2 through 52 are shown (top to bottom respectively) at 10 frame intervals.

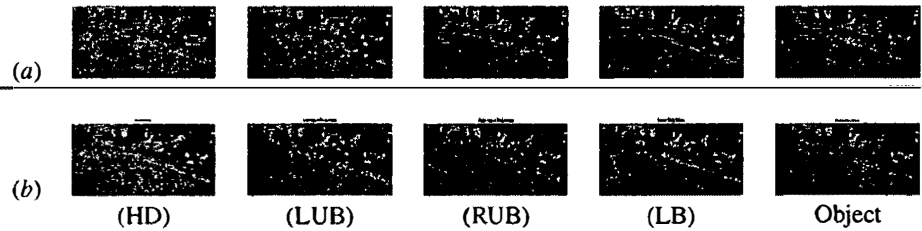


Figure 5: Pedestrian component-based spatio-temporal object detection. (a) shows the initialization/proposal distribution, and (b) the mean of 30 samples taken from the belief for each of the four component and an object. The detection and tracking was conducted using a 3-frame smoothing window.

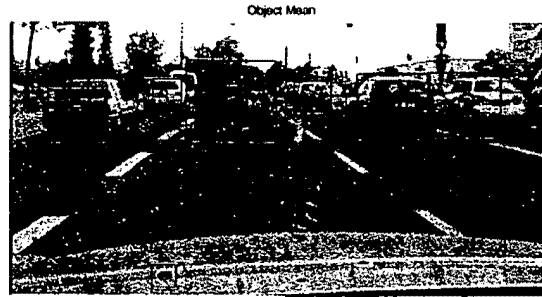


Figure 6: Multiple target detection example.

3.3 Single frame Multi-target Detection

While in general the algorithm presented here is capable of detecting multiple targets, by converging to multi-modal distributions for components and objects, in practice this tends to be very difficult. It is well known that particle filters have problems tracking multi-modal distributions [3]. This framework that further extends particle filtering, and requires message update routine to take products over particle sets, makes this problem even more apparent. We postulate that given a large number of particles this can be done, but would be prohibitively expensive. Instead we use a peak suppression scheme, where we look for the modes one at a time, and suppress the response of our likelihood function in the regions where peaks have already been found. An example of this that is produced by running a purely spatial graphical model over the image is shown in Fig. 6.

4 Conclusion

References

- [1] L. Sigal, M. Isard, B.H. Sigelman and M.J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation, *Advances in Neural Info. Proc. Sys.* 16, 2003.
- [2] L. Sigal, S. Bhatia, S. Roth, M.J. Black and M. Isard. Tracking Loose-limbed People, *CVPR*, 2004.
- [3] K. Okuma, A. Teleghani, N. de Freitas, J.J. Little and D.G. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking, *ECCV*, 2004.
- [4] A. Mohan, C. Papageorgiou and T. Poggio. Example-Based Object Detection in Images by Components, *IEEE Trans. on Pattern and Machine Intelligence*, 23(4):349–361, 2001.
- [5] C. Papageorgiou and T. Poggio. Trainable Pedestrian Detection, *ICIP*, 1999.
- [6] P. Viola, M. Jones and D. Snow. Detecting Pedestrians Using Patterns of Motion and Appearance, *ICCV*, pp. 734–741, 2003.
- [7] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, *CVPR*, 2001.
- [8] B. Xie, D. Comaniciu, V. Ramesh, M. Simon and T. Boult. Component Fusion for Face Detection in the Presence of Heteroscedastic Noise, *Annual Conf. of the German Society for Pattern Recognition (DAGM'03)*, pp. 434–441, 2003.
- [9] M. Burl, M. Weber and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry, *ECCV*, pp. 628–641, 1998.
- [10] L. Fei-Fei, R. Fergus and P. Perona. A Bayesian approach to unsupervised One-Shot learning of Object categories, *ICCV*, 2003.

- [11] R. Fergus, P. Perona and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning, *CVPR*, 2003.
- [12] H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars, *CVPR*, 2000.
- [13] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation, *ECCV* Vol. 3, pp. 453–468, 2002.
- [14] J. Deutscher, A. Blake and I. Reid. Articulated body motion capture by annealed particle filtering, *CVPR*, 2000.
- [15] A. Douce, N. de Freitas and N. Gordon. Sequential Monte Carlo methods in practice, *Statistics for Engineering and Information Sciences*, pp. 3–14, Springer Verlag, 2001.
- [16] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures, *CVPR*, Vol. 2, pp. 66–73, 2000.
- [17] P. Felzenszwalb and D. Huttenlocher. Efficient Belief Propagation for Early Vision, *CVPR*, 2004.
- [18] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, 22(1):67–92, 1973.
- [19] M. Isard. PAMPAS: Real-valued graphical models for computer vision, *CVPR*, Vol. 1, pp. 613–620, 2003.
- [20] M. Jordan, T. Sejnowski and T. Poggio. Graphical models: Foundations of neural computation, *MIT Press*, 2001.
- [21] E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, *CVPR*, Vol. 1, pp. 605–612, 2003; (see also MIT AI Lab Memo 2002-020).
- [22] A.T. Ihler, E.B. Sudderth, W.T. Freeman and A.S. Willsky. Efficient Multiscale Sampling from Products of Gaussian Mixtures, *Advances in Neural Info. Proc. Sys. 16*, 2003.
- [23] J. Yedidia, W. Freeman and Y. Weiss. Generalized belief propagation, *Advances in Neural Info. Proc. Sys. 13*, pp. 689–695, 2000.
- [24] K. Murphy, A. Torralba and W.T. Freeman. Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes, *Advances in Neural Info. Proc. Sys. 16*, 2003.
- [25] J. Friedman, T. Hastie and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–374, 2000.
- [26] R. Schapire. The boosting approach to machine learning: An overview, *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.
- [27] S. Yu, R. Gross, and J. Shi. Object segmentation by graph partitioning Concurrent object recognition and segmentation by graph partitioning, *Advances in Neural Info. Proc. Sys. 15*, pp. 1407–1414, 2003.