

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224359311>

# Background modeling from a free-moving camera by Multi-Layer Homography Algorithm

**Conference Paper** in *Proceedings / ICIP ... International Conference on Image Processing* · November 2008

DOI: 10.1109/ICIP.2008.4712069 · Source: IEEE Xplore

CITATIONS

60

READS

314

5 authors, including:



**Linmi Tao**

Tsinghua University

106 PUBLICATIONS 893 CITATIONS

[SEE PROFILE](#)



**Naveed Rao**

National University of Sciences and Technology

26 PUBLICATIONS 255 CITATIONS

[SEE PROFILE](#)



**Guangyou Xu**

Tsinghua University

255 PUBLICATIONS 2,881 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Brain Computer Interface [View project](#)



Content-Based Video Retrieval [View project](#)

# BACKGROUND MODELING FROM A FREE-MOVING CAMERA BY MULTI-LAYER HOMOGRAPHY ALGORITHM

Yuxin Jin, Linmi Tao, Huijun Di, Naveed I Rao, Guangyou Xu

Department of Computer Science, Tsinghua University  
jyx05@mails.tsinghua.edu.cn

## ABSTRACT

This paper proposes a novel Multi-layer Homography Algorithm for background modeling from a free-moving camera. Background is composed of many planes. Different planes satisfy with different homographies which can be found by our algorithm. Each pixel except for the moving pixel definitely belongs to some plane. Rectified by the corresponding homography, each static pixel in the shared view can find its match in the previous frame. Thus, frames can be rectified to a specific viewpoint for background modeling. Experiment shows it is effective. Our approach can be used in motion detection from a free-moving camera.

**Index Terms**— Multi-layer Homography, free-moving camera, background model

## 1. INTRODUCTION

Detecting moving objects is one main task in computer vision. Many high-level understanding is based on the motion detection. In order to achieve it, a stable background model is necessary. Many previous works [1][2][3] use background model to find the motion region. However, they assume the scene is captured by fixed camera. For moving camera, [4] presents a method for modeling in a panoramic view which is based on some literatures on image mosaic techniques for constructing panoramic views (e.g., [5]). But, they all have an assumption that the camera movement only contains rotation about its optical center which can guarantee no significant parallax.

Approach in [6] estimates the motion in video by tracking particles selected in the video. It does not construct an explicit background model. Our previous work [7] can handle the background modeling from free-moving camera (both transfer and rotation occurs) by finding pixel correspondences. But it cannot handle the problem with large depth disparity where the static pixels close to the camera cannot find correspondences. [8] solves the problem by implementing multiview geometric constraints after homography and epipolar filter. However, it suffers from some degenerate cases.

This work is supported by National Science Foundation of China under grant No 60673189 and No 60433030

Based on our previous work, this paper proposes Multi-Layer Homography Algorithm to constructing the background model for motion detection in presence of large depth disparity from a moving camera. By this algorithm, any static pixels can find correspondences regardless of their different depth. The experiment result shows the superiority of our approach.

This paper is organized as follows. Section 2 proposes the Multi-layer Homography Algorithm. Section 3 describes how to find pixel correspondences under multi-layer scheme. Section 4 introduces a method to construct a panorama background for a video sequence. Section 5 shows the experiment results.

## 2. MULTI-LAYER HOMOGRAPHY ALGORITHM

In our previous work [7] and Yuan's approach [8], Current frame is initially registered into the viewpoint of previous frame or panorama background by homography. It is applicable if the scene can be roughly viewed as one plane (such as the aerial video) or the depth disparity is relatively low compared with the distance from the camera. In two frames with strong parallax and large depth disparity (shown in Fig.1), our previous method is incapable and Yuan's approach need one more frame and may fail when degenerate case exists.

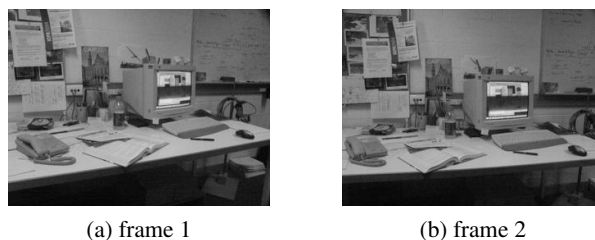


**Fig. 1.** Two frames with strong parallax and large depth disparity

Observing the open problem, it is logical to propose a new algorithm - Multi-layer Homography Algorithm which can find different planes in one scene. Each pixel in the scene definitely belongs to some plane. In two frames with parallax captured by a free-moving camera, the pixels in the same plane satisfy with one homography induced by the plane. Our

approach's goal is to compute several homography induced by different planes on which every static pixel lies. Hence, those pixels can find their accurate correspondences in another frame after transformed by proper homography.

To find homography, feature points are necessary. Harris algorithm is used to find the corners in two frames. Thereafter, KLT tracker [9] is performed to find putative correspondences of these feature points. Denote  $n$  putative matched feature points set as  $\{\mathbf{p}_i^{(1)}\} = \{(x_i^{(1)}, y_i^{(1)}, 1)\}$  for first frame and  $\{\mathbf{p}_i^{(2)}\} = \{(x_i^{(2)}, y_i^{(2)}, 1)\}$  for second frame, where  $i = 1, 2, \dots, n$  (all 2D pixels are represented in homogeneous form). Assuming there exist  $m$  layers homography  $\mathbf{H}_j, j = 1, 2, \dots, m$  induced by  $m$  planes in the scene, denote the correspondences satisfying with  $j$ th layer homography  $\mathbf{H}_j$  as  $(\mathbf{p}_{jk_j}^{(1)}, \mathbf{p}_{jk_j}^{(2)}), k_j = 1, 2, \dots, n_j$  where  $k_j$  is the number of the correspondences induced by  $j$ th homography. We have  $\mathbf{p}_{jk_j}^{(1)} = \mathbf{H}_j \mathbf{p}_{jk_j}^{(2)}$ .

Our algorithm is quite straightforward. Firstly, a subset of the putative correspondences is computed by RANSAC fitting one specific homography. Secondly, after excluding the previously calculated subset, we perform RANSAC again to find the next layer of homography for the residual feature points. After  $m$  iteration or no enough correspondences remained, algorithm terminates and  $m$  or less layers of homography can be found.

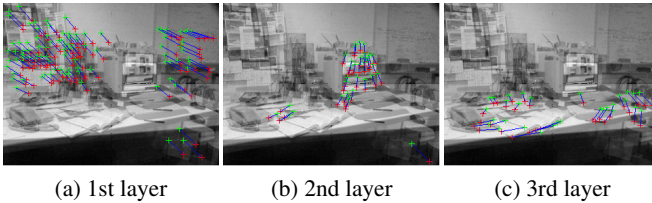


**Fig. 2.** Three layer homography

Frame 2 in Fig.1 is rectified and registered into frame 1. Fig.2 superimposes frame 2 onto frame 1 and shows three set of correspondences satisfying with different layers of homography induced by three main planes: wall, computer screen and desk. In order to build background model, a dense match for all pixels is required. A new approach to achieve dense match is introduced in the next section.

### 3. DENSE PIXEL CORRESPONDENCE

In our previous work [7], likelihood and smoothness constraints are introduced for optimizing matching problem and are formulated as energy function. The correspondences are found between the panorama background and the current frame rectified by main homography. But it only suits the situation where only one main plain exists in the scene. Under new Multi-layer Homography scheme, some new function should be introduced to decide which layer each pixel lies in.

In this section, we discuss how to rectify one frame so as to register it into another frame's viewpoint.

We let the rectified pixel in the second frame be  $\hat{\mathbf{x}}_j^{(2)} = H_j^{-1} \mathbf{x}^{(2)}$  according to  $j$ th layer. Our task is to assign layer label  $l_h \in 1, 2, \dots, m$  to each pixel in the overlap area. Here the overlap area means the same area shared with the first frame and the rectified second frame.

In the first frame, we search for the most similar pixel to the pixel  $\mathbf{x}^{(2)}$  in the neighbor of location  $\hat{\mathbf{x}}_j^{(2)}$  ( $j$ th layer). The neighbor  $D = \{(x, y) | -a \leq x \leq a, -b \leq y \leq b\}$ .

$$\Delta \mathbf{x}_j^{(2)} = \arg \min_{\Delta \mathbf{x} \in D} \{I^{(1)}(\hat{\mathbf{x}}_j^{(2)} + \Delta \mathbf{x}) - I^{(2)}(\mathbf{x}^{(2)})\} \quad (1)$$

$$j = 1, 2, \dots, m$$

$I^{(i)}(\mathbf{x})$  is the intensity of the pixel  $\mathbf{x}$  in the  $i$ th frame.  $\Delta \mathbf{x}_j^{(2)}$  is the displacement. The best layer for one pixel in the second frame let the rectified pixel coincides with its correspondence in the first frame. Intensity is used to measure the match. Then, the layer label for each overlap pixel is

$$l_h = \arg \min_j \{I^{(1)}(\hat{\mathbf{x}}_j^{(2)} + \Delta \mathbf{x}_j^{(2)}) - I^{(2)}(\mathbf{x}^{(2)})\} \quad (2)$$

The difference of intensity for one correspondence is

$$\Delta I(\mathbf{x}^{(2)}) = I^{(1)}(\hat{\mathbf{x}}_{l_h}^{(2)} + \Delta \mathbf{x}_{l_h}^{(2)}) - I^{(2)}(\mathbf{x}^{(2)}) \quad (3)$$

For each pixel outside overlap area, the layer label is also necessary. Most non-overlap pixels stay in the smooth segment shared with some overlap pixels. In [7], MST (minimal span tree) is used to represent the scene smoothness. Here, we use it to decide which segment these pixels belong to. The tree can be trimmed if the weight of edge  $(u, v)$  is higher than some threshold. The trimmed part can be regarded as one smooth segment. For those non-overlap pixels in the smooth segment, the layer label is assigned to them same as the label of majority of overlap pixels in this segment.

In this way, each overlap pixel  $\mathbf{x}^{(2)}$  in the second frame can find its correspondence  $\hat{\mathbf{x}}_{l_h}^{(2)} + \Delta \mathbf{x}_{l_h}^{(2)}$  in the first frame and each non-overlap pixel can be properly rectified. Unlike [6], our approach is for dense match.

Based on our dense correspondences, background modeling for video sequence is feasible which is to be introduced in the next section.

### 4. VIDEO BACKGROUND MODELING AND FOREGROUND SEGMENTATION

With the help of Multi-layer Homography Algorithm and dense pixel correspondence, background model can be constructed for video captured by a free-moving camera. We adopt the same background modeling approach as [2]. Each pixel's values are modeled as a mixture of Gaussians.

The first frame of the video is used for the initialization of the panorama. Then, each following frame in the video

is rectified to the viewpoint of the first frame's viewpoint by our approach. In frame  $t$ , the pixel  $\mathbf{x}^{(t)}$  is rectified to  $\mathbf{x}_{l_h}^{(t)} = \hat{\mathbf{x}}_{l_h}^{(t)} + \Delta\mathbf{x}_{l_h}^{(t)}$ . Then the MoG parameter at location  $\mathbf{x}_{l_h}^{(t)}$  in the background panorama will be updated according to the strategy in [2]. The frame may contain some pixels outside the panorama background, rectify them to proper location according to the method introduced in Section 3, and initialize the background model at this location.

With the background model, the probability for this pixel to be background is calculated as

$$P(I^{(t)}(\mathbf{x}^{(t)})) = \sum_{i=1}^K \omega_i^{(t)}(\mathbf{x}_{l_h}^{(t)}) \eta(I^{(t)}(\mathbf{x}^{(t)}), \mu_i^{(t)}(\mathbf{x}_{l_h}^{(t)}), \Sigma_i^{(t)}(\mathbf{x}_{l_h}^{(t)})) \quad (4)$$

where  $K$  is the number of Gaussian. For the pixel at location  $\mathbf{x}_{l_h}^{(t)}$  in the panorama background,  $\omega_i^{(t)}$  is the occupancy probability for the  $i$ th Gaussian,  $\mu_i^{(t)}$  and  $\Sigma_i^{(t)}$  are the mean value and covariance matrix at this location (for simplicity, superscript  $(t)$  will be omitted).  $\eta$  is the Gaussian density function

$$\eta(I, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(I-\mu)^T \Sigma^{-1} (I-\mu)} \quad (5)$$

Unlike many approach for motion detection based on background modeling, we label the pixel as background or foreground under global restriction firstly given in our previous work [7] and modified under the new scheme.

In [7], the global constraints are used not only to decide whether the pixels belong to background, but also find displacement for alignment. In this paper, the displacement for pixels is found during the process of finding correspondences in Section 3 (equation 1 and 2). More searching for the displacement is redundant in this stage and cannot yield much profit compared with its high computational cost. Thus, the pixels are only labeled as background or foreground.

The likelihood function for each pixel can be written as

$$\ell(l(\mathbf{x})|I(\mathbf{x})) = \begin{cases} P(I(\mathbf{x})), & l(\mathbf{x}) = 0 \\ 1 - P(I(\mathbf{x})), & l(\mathbf{x}) = 1 \end{cases} \quad (6)$$

where  $l(\cdot)$  is the label,  $l(\cdot) = 0$  for background,  $l(\cdot) = 1$  for foreground. Similar to [7], MST is used for smoothness constraint. Assuming it has the set of nodes  $V$  and edges  $E$ , the smoothness function can be formulated as

$$S(l(u), l(v)) = \begin{cases} \alpha_1 e^{-w(u,v)}, & l(u) \neq l(v) \\ \alpha_2 e^{-1/w(u,v)}, & l(u) = l(v) \end{cases} \quad (7)$$

where the  $w(u, v)$  is the edge weight in the MST, a small value stands for  $u$  and  $v$  are in the same smooth area while a large value means they may in different segments.  $\alpha$  is some constant value. To label each pixel is equivalent to minimize the following energy function

$$E_{min} = \min_l \left( \sum_{u \in V} \log \ell(l(u)) + \sum_{(u,v) \in E} S(l(u), l(v)) \right) \quad (8)$$

The global optimal label corresponding to the above minimal energy function can be attained by dynamic programming on a tree. Then, the foreground pixel can be segmented.

## 5. EXPERIEMENTS



(a) Rectified Frame 2 with 4 homography layers



(b) Disparity from Frame 1



(c) Rectified Frame 2 with 1 homography layer



(d) Disparity from Frame 1

**Fig. 3.** Comparison between proposed approach and previous approach in [7] on data set 1

Some parameters are adaptively or manually selected. The number of layers is adaptively decided. When the remained feature points used for finding homography by RANSAC are fewer than some quantity (10-16 is proper), the algorithm will automatically stop. The neighbor size for finding displacement is  $(-3, -3) - (+3, +3)$ . We let  $\alpha_1 = \max 1/(e^{-w(u,v)})$  and  $\alpha_2 = \max 1/(e^{-1/w(u,v)})$ .

Our Algorithm is tested on several data sets. The first data set (shown in Fig.3) comes from <http://www.csse.uwa.edu.au/pk/Research/MatlabFns/index.html>. Fig.3 compares the new algorithm with previous algorithm [7]. Without multi-layer homography, the second frame cannot be correctly rectified to the first viewpoint. As a result, [7] approach fails.

The second data set is the same as that in [7] (24 frames). The proposed approach attains similar result shown in Fig.4 as that reported in [7]. It is because that the depth disparity of the scene is relatively small compared with the distance from the camera and only minor movement of camera exists.

The third data set is from [6] containing 200 frames. The result is shown in Fig.5. The fourth data set is captured by hand-held camera roughly moving along the same direction as the cars (degenerate case in [8]). Moreover, the distance from the building is not very far.

Worth to mention, in the last several frames, the correspondences of feature points are hard to find due to the great parallax (large rotation) between the panorama and the cur-

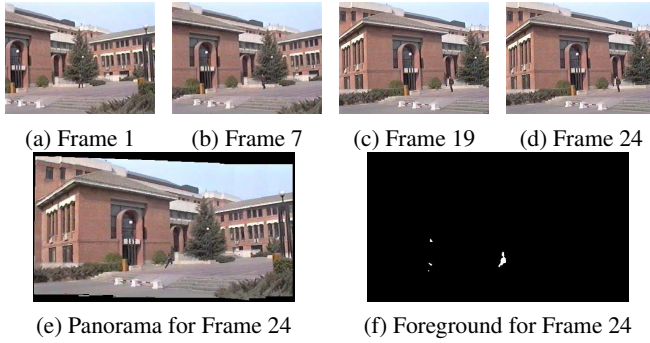


**Fig. 4.** Our result on data set 2

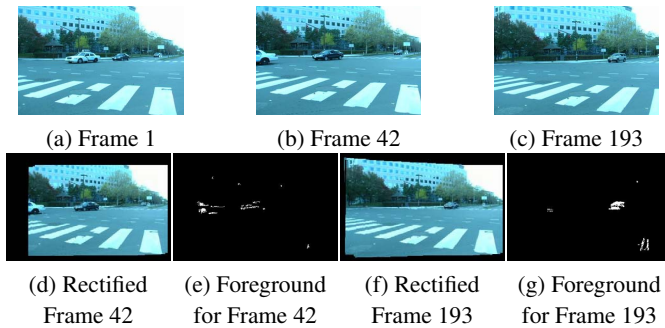


**Fig. 5.** Our result on data set 3

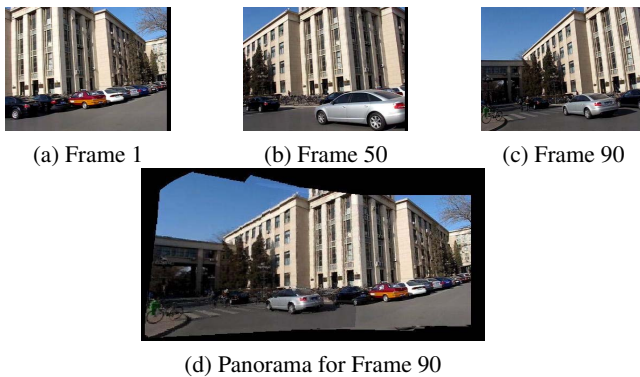


**Fig. 6.** Our result on data set 4

rent frame which will cause KLT tracker inapplicable. To solve the problem, a more robust tracker may be used such as SIFT. Additionally, misalignment still exists and cumulative error cannot be totally eliminated by our approach.

## 6. CONCLUSION

This paper proposes a novel Multi-layer Homography Algorithm to construct background model for scene captured by a free-moving camera. It is based on our previous work [7] on this challenging problem. Better performance of our approach should be ascribed to its effective algorithm to find

several homography layers induced by different plains in the scene. With this important improvement, only some minor modifications are made on the background model establishment and energy function for foreground segmentation.

We still notice some deficiency in our algorithm. Firstly, cumulative error still exists. Secondly, when it occurs large parallax between the current frame and the panorama especially caused by rotation, the putative matches of feature points cannot be reliably found. Our future work will focus on these problems.

## 7. REFERENCES

- [1] Cristani M, Bicegi M, and Murino V, "Integrated region- and pixel-based approach to background modeling," in *Proc. Workshop on Motion and Video Computing*. IEEE, 2006, pp. 3–8.
- [2] Stauffer C and Grimson W., "Learning patterns of activity using real-time tracking," *IEEE Transactions on PAMI*, vol. 22, pp. 747–757, August 2000.
- [3] Mittal A and Paragios N., "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. Computer Vision and Pattern Recognition*. IEEE, 2004, pp. 302–309.
- [4] Mittal A and Huntenlocher D., "Scene modeling for wide area surveillance and image synthesis," in *Proc. Computer Vision and Pattern Recognition*. IEEE, 2000, vol. 2.
- [5] M.Irani, P.Anandan, and S.Hsu, "Mosaic based representations of video sequences and their applications," in *Proc. Int'l Conf. Computer Vision*. IEEE, 1995, pp. 605–611.
- [6] Peter Sand and Seth Teller, "Particle video: Long-range motion estimation using point trajectories," in *Proc. Computer Vision and Pattern Recognition*. IEEE, 2006.
- [7] Rao N.I, Huijun Di, and Guangyou Xu, "Joint correspondence and background modeling based on tree dynamic programming," in *Proc. Int'l Conf. Pattern Recognition*. IEEE, 2006, vol. 2, pp. 425–428.
- [8] Chang Y., Gerard, Jinman K., and Isaac C., "Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints," *IEEE Transactions on PAMI*, vol. 29, pp. 1627–1633, September 2007.
- [9] B.D.Lucas and T.Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int'l Joint Conf. Artificial Intelligence*, 1981, pp. 674–679.