

A Background Layer Model for Object Tracking through Occlusion

Yue Zhou and Hai Tao

Department of Computer Engineering
University of California, Santa Cruz, CA 95064
{zhou,tao}@soe.ucsc.edu

Abstract

Motion layer estimation has recently emerged as a promising object tracking method. In this paper, we extend previous research on layer-based tracker by introducing the concept of background occluding layers and explicitly inferring depth ordering of foreground layers. The background occluding layers lie in front of, behind, and in between foreground layers. Each pixel in the background regions belongs to one of these layers and occludes all the foreground layers behind it. Together with the foreground ordering, the complete information necessary for reliably tracking objects through occlusion is included in our representation. An MAP estimation framework is developed to simultaneously update the motion layer parameters, the ordering parameters, and the background occluding layers. Experimental results show that under various conditions with occlusion, including situations with moving objects undergoing complex motions or having complex interactions, our tracking algorithm is able to handle many difficult tracking tasks reliably.

1 Introduction

In recent years, dynamic motion layer estimation has emerged as a promising approach for object tracking [4],[9],[5],[8]. A motion layer is a region in an image that undergoes a coherent motion. The two chief problems in motion layer based tracking algorithms are how to represent motion layers and how to estimate the parameters associated with these layers.

With the dynamic motion layer representation, tracking problem can be formulated as the *maximum a posteriori* (MAP) estimation of a Hidden Markov Model (HMM) [8]. In a typical motion layer estimation process, both foreground objects and background are modeled and they compete with each other to maximize the joint posterior probability. This is one of the main reasons behind the success of layer trackers.

In terms of layer representation, in previous work, only motion, segmentation, and appearance are considered. This object representation works well for tracking multiple objects when no occlusion presents. However, it is insufficient in accommodating occlusion caused by foreground or background object and in effectively

modeling the spatial relationship among moving objects and the background.

Previous work on motion layer analysis and motion layer based tracking employed global or local motion representations [10],[11],[12]. The object shape and appearance are often modeled as Gaussian distributions [9], Markov Random Fields (MRF) [10] or other mixture models [2]. To handle object occlusion in motion analysis and tracking, an explicit generative occlusion boundary model was proposed in [3]. To handle self-occlusion on the foreground objects and adaptively change the shapes of the foreground objects to allow the tracking of non-rigid motion, [4] proposed a combined parametric shape and motion model with depth ordering to represent the visibility of each layers. The Transformed Hidden Markov Model (THMM) algorithm [6] includes both motion and appearance representation as the parameters in a generative model and formulates the tracking problem as the learning of these parameters.

In this paper, we propose a novel scene representation with ordering information that contains complete information for inferring the foreground object and the background occlusion. In this representation, each moving object is modeled as a foreground layer. Some background objects such as trees may occlude foreground motion layers. To model the depth difference in background, we introduce background layers that lie between foreground layers. This is different from the previous methods where the background region is modeled as a single layer. In addition, the depth ordering of foreground layers is treated as a state variable to explicitly model the depth relations among foreground objects. Unlike the global shape model in [9], we also allow gradual but arbitrary changes in objects shapes, which are captured in the foreground mask.

Based on this new layer representation, we propose an estimation algorithm that estimates the motion layer parameters, the foreground ordering, and the background layers in an MAP framework. The overall formulation can be written as

$$\max_{\Lambda_t} \arg P(\Lambda_t | \Lambda_{t-1}, I_t, \dots, I_0) \quad (1)$$

where Λ_t is the state of the tracker at time t , and I_t is the image observation.

The rest of the paper is organized as follows. The details of the proposed layer representation are presented in Section 2. Section 3 describes the MAP estimation of the layer state. Section 4 describes the implementation and demonstrates the experimental results. Some discussions and conclusions can be found in Section 5.

2 Dynamic layer modeling

2.1 Depth ordering of the foreground and background layers

In our proposed approach, a dynamic scene is represented by foreground and background layers. As shown in Figure 1, foreground motion layers are ordered according to their relative depth from front to back. The front-most layer is layer 1. Some background regions, which are defined as the image areas that do not move, may lie in between foreground layers. These background regions are in front of some foreground objects and are called occluding background layers. In our model, as shown in Figure 1, there is one background layer between every two neighboring foreground layers. There is also one background layer that is behind all foreground layers and one layer that is in front of all foreground layers (layer 1). If there are L foreground layers, there are $L+1$ background layers.

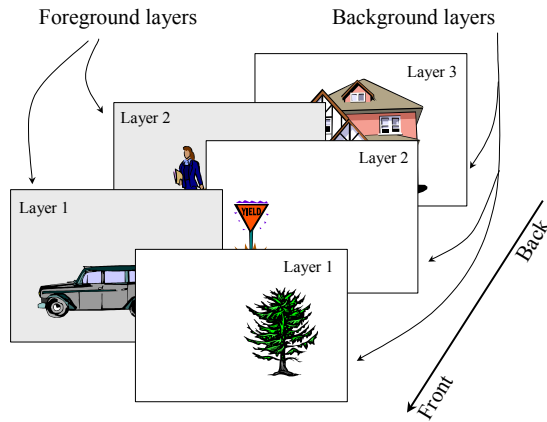


Figure 1. The ordered layer model.



Figure 2. An example of the background layers.

Each foreground layer is described by its motion, shape, and appearance. Each background layer is described by its shape and appearance. If the background also moves, all background layers share a single motion. At any time t ,

the set of all layer parameters is called the state of the tracker and is denoted by Λ_t . In later sections, we will describe in detail the models for these state variables. Figure 2 shows a real example of a video frame and the top-most occluding background layer. In this example, only the shape of the front-most background layer is shown.

Using the above layer model, from the object point of view, each object belongs to one of the foreground layers. We explicitly model and estimate the layer assignment for the objects in the scene. The depth ordering of L foreground objects at time t is denoted as $O_t = [i_1, i_2, \dots, i_L]$. The integer variable $i_j \in [1, L]$ and $i_k \neq i_l$ iff $k \neq l$. If we assume that objects do not interleave with each other, there are $L!$ possible layer assignments for L foreground objects. We further assume that the depth ordering is a random variable with a uniform distribution. This means all the permutations have the same probability and thus have the same prior probability $P(O_{t,i}) = 1/L!$, where $O_{t,i}$ is an arbitrary layer ordering configuration.

It should be noticed that the foreground layer ordering, together with the shape, appearance, and motion information of all foreground and background layers, provide the complete information for occlusion reasoning.

2.2 Motion models

We describe the background motion using a 2D affine model, and estimated this model using the so-called direct method [1]. All background layers share the same motion. Each foreground layer undergoes a 2D rigid motion, which is described using position μ , orientation ω , scaling factor s , and their temporal derivatives. A constant velocity model is used to describe the dynamics of the foreground layers. If we denote the motion parameters of a layer as $\theta = [\mu, \omega, s, \dot{\mu}, \dot{\omega}, \dot{s}]$, then the motion dynamics is written as

$$P(\theta_t | \theta_{t-1}) = N(\theta_t : \Phi \theta_{t-1}, Q) \quad (1)$$

where θ_t is the motion parameter at time t , Φ is the standard transition matrix for a constant velocity model, and the notation $N(x : \mu, R)$ denotes a multivariate Gaussian distribution with mean μ and covariance matrix R .

2.3 Shape models of the foreground and background layers

Each foreground or background layer is associated with a shape map. At each pixel location, the value of the shape map is the probability that the object in the layer presenting at that pixel location (it may not be visible though). For the foreground layer j and position x_i at time t , we denote

the value of the shape map as $\tau_{t,j}(x_i)$. For the background layer, we use the notation $\pi_{t,j}(x_i)$ to represent its shape map. One difference between the foreground shape map and the background shape map is that for the background, the probabilistic values of all shape maps at each pixel must sum up to 1. This reveals our underlying assumption that there is only one background surface for each pixel. This is a reasonable assumption because even there are more surfaces they will not be observable anyway.

2.3.1 Layer visibility

Once the shape maps are defined for all layers, for each pixel x_i , we can compute the probability that the j th foreground layer is visible. This is the probability of the joint event that background layers 1 to j are absent, foreground layer 1 to $j-1$ are absent, and j th foreground layer presents at x_i . The first probability is $1 - \sum_{l=1}^j \pi_l(x_i)$ because there is only one background surface. The second probability is $\prod_{s=1}^{j-1} [1 - \tau_s(x_i)]$, and the third probability is $\tau_j(x_i)$ (for simplicity, we ignore the subscript t). As a result, the probability of the j th foreground layer being visible at x_i is

$$P_j(x_i) = \tau_j(x_i) \cdot (1 - \sum_{l=1}^j \pi_l(x_i)) \cdot \prod_{s=1}^{j-1} [1 - \tau_s(x_i)] \quad (2)$$

Similarly, the probability of observing the j th background layer at x_i is

$$P_{B,j}(x_i) = \pi_j(x_i) \cdot \prod_{k=1}^{j-1} (1 - \tau_k(x_i)) \quad (3)$$

and the probability of observing one of the background layers is

$$P_B(x_i) = \sum_{j=1}^{L+1} \left[\pi_j(x_i) \cdot \prod_{k=1}^{j-1} (1 - \tau_k(x_i)) \right] \quad (4)$$

2.3.2 Shape dynamics

If we assume the shape of the foreground does not change dramatically, then we can use a constant value Gaussian model to describe the dynamics of the shape changes over time. More specifically,

$$P(\tau_{t,j}(x_i) | \tau_{t-1,j}) = \gamma + N(\tau_{t,j}(x_i); \tau_{t-1,j} (R(-\dot{\omega}_{t,j})(x_i - \dot{\mu}_{t,j}) / \dot{s}_{t,j}), \sigma_\tau^2) \quad (5)$$

where γ represents the uncertainty in the shape of the layer. The transformation $R(-\dot{\omega}_{t,j})(x_i - \dot{\mu}_{t,j}) / \dot{s}_{t,j}$ is used to align the shape maps.

2.4 Appearance model

The appearance of foreground layer j is defined in the local coordinate system and is denoted as $A_{t,j}$. We assume that the image observation model is a Gaussian distribution with the appearance as the mean, or

$$P(I_t(x_i) | A_{t,j}(x_i)) = N(I_t(x_i); A_{t,j}(x_i), \sigma_I^2) \quad (6)$$

where σ_I^2 is the variance of the image observation.

Like the motion and shape models, we also assume that the temporal changes of the layer appearance follow a constant value Gaussian distribution. This is formulated as

$$P(A_{t,j}(x_i) | A_{t-1,j}(x_i)) = N(A_{t,j}(x_i); A_{t-1,j}(x_i); \sigma_A^2) \quad (7)$$

where σ_A^2 is the appearance uncertainty that accounts for the appearance variations.

2.5 The MAP estimation

The tracking procedure can be considered as the maximization of the posterior probability

$$\arg \max_{\Lambda_t} P(\Lambda_t | \Lambda_{t-1}, I_t, \dots, I_0) \quad (8)$$

Using Bayes rule and the HMM model,

$$P(\Lambda_t | \Lambda_{t-1}, I_t, \dots, I_0) = P(I_t | \Lambda_t) \cdot P(\Lambda_t | \Lambda_{t-1}) \quad (9)$$

where $P(\Lambda_t | \Lambda_{t-1})$ is the state prior function, and $P(I_t | \Lambda_t)$ is the likelihood function.

Based on our models in the previous sections, the prior function is computed as

$$P(\Lambda_t | \Lambda_{t-1}) = P_{order} \cdot P_{fg_shape} \cdot P_{bg_shape} \cdot P_{motion} \cdot P_{appearance} \quad (10)$$

where

$$P_{order} = P(o_t | o_{t-1})$$

$$P_{fg_shape} = \prod_{j=1}^L \prod_{i=1}^{N_j} P(\tau_{t,j}(x_i) | \tau_{t-1,j}(x_i))$$

$$P_{bg_shape} = \prod_{j=1}^{L+1} \prod_{i=1}^{N_j} P(\pi_{t,j}(x_i) | \pi_{t-1,j}(x_i))$$

$$P_{motion} = \prod_{j=1}^L P(\theta_{t,j} | \theta_{t-1,j})$$

$$P_{appearance} = \prod_{j=1}^L \prod_{i=1}^{N_j} P(A_{t,j}(x_i) | A_{t-1,j}(x_i))$$

Here we assume the image has L foreground layers, $L+1$ background layers, and N_j pixels on the object in layer j .

To compute the likelihood function, we need to first obtain the probabilistic distribution of the front-most layer at each pixel based on foreground layer ordering, foreground and background shapes, and the appearance models. More specifically, we compute the likelihood function as

$$P(I_t | \Lambda_t) = \prod_{i=1}^N (P_{bgo}(x_i) + P_{fgo}(x_i)) \quad (11)$$

where $P_{bgo}(x_i)$ and $P_{fgo}(x_i)$ represent the likelihood of one of the background or foreground layers is visible at pixel x_i . They can be computed as

$$P_{bgo}(x_i) = P(I(x_i) | B(x_i)) \cdot P_B(x_i) \quad (12)$$

and

$$P_{fgo}(x_i) = \sum_{j=1}^L [P(I_j(x_i) | A_j(x_i)) \cdot P_j(x_i)] \quad (13)$$

$P_B(x_i)$ and $P_j(x_i)$ are defined in Eq(2-4).

3 Estimation of the object state

Solving Eq(8) is a difficult optimization problem because the state space is very large. An approximate solution can be found by first decomposing the original problem into several sub-problems (see Figure 3). Then optimization is performed to solve these sub-problems sequentially. We found that in practice this approach yields feasible solutions.

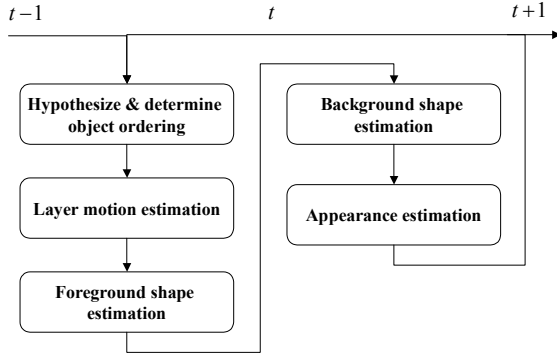


Figure 3. Estimation of the state parameters.

3.1 Foreground layer ordering hypothesis

Foreground layer ordering O_t is modeled as a uniformly distributed random variable. Because of this property, the estimation of O_t is rather simple: the algorithm goes through all the possible value of O_t , and finds the one that maximizes the posterior probability. Since all the other parameter estimation steps highly depend on the depth ordering, it is computed at the beginning of each iteration.

3.2 Motion estimation

Maximizing the posterior probability in Eq(8) w.r.t. foreground layer motion is equivalent to optimizing the function

$$\prod_{i=1}^n (P_{bgo}(x_i) + P_{fgo}(x_i)) \cdot P_{motion} \quad (14)$$

A search algorithm can be used to find the solution around the predicted position. Rotation, translation, and the scaling factor are discretized with sufficient precision for this purpose. For sequence with moving background, a direct method [1] is used to estimate the motion parameters.

3.3 Foreground shape estimation

From Eq(11-13) and Eq(2-4), it can be observed that the likelihood is a linear function of each foreground shape variable $\tau_j(x_i)$ and the prior term is a Gaussian function of $\tau_j(x_i)$. If we optimize $\tau_j(x_i)$ independently for each layer, the estimation becomes the maximization of a function in the form of

$$(ax + b)e^{-(x-x_0)^2/2\sigma^2} \quad (15)$$

where a and b are constants that can be computed using Eq(11-13) and Eq(2-4). The optimal solution is 0, 1, or the root of the quadratic equation

$$ax^2 + (b - ax_0)x - (bx_0 + a\sigma^2) = 0 \quad (16)$$

This equation is derived by taking the derivatives of the function in Eq(15) and set it to be 0.

3.4 Background shape estimation

Estimation of the background shape is similar to the estimation of the foreground shape. However, there is one additional constraint needs to be enforced: the values of all shape maps should sum up to 1. With this constraint, the global optimization becomes complicated. However, we can use the results in the previous frame or previous iteration as the starting point to perform a greedy algorithm to find the local optimal solution. We estimate each background level individually with the shape maps of other layers fixed. After all shape values for all layers are estimated, they are normalized so that their sum becomes 1.

There is another difference between the background shape estimation and the foreground shape estimation. For foreground, the object shape does not change significantly over time because of the 2D rigid model. Therefore we use the shape in the previous frame as our prior in the estimation. However, in the background shape estimation, the shape of each background layer highly depends on object motion. The occluding background shape in the same area can change quickly from time to time because of object movements. For example, a car may first pass behind a tree, turn around and then pass in front of the tree again; in the first case the tree is part of the occluding

background layer to the foreground layer of car, while in the second case the tree belongs to the background layer that does not occlude the same foreground. So in our algorithm if all the objects leave an area for a certain period of time, we actually lack visual information to infer background layer shapes. As a result, no matter what the previous background shape values are, they become obsolete and the shape of all background layers are reset to a default value.

3.5 Appearance estimation

To estimate the appearance, we need to find $A_{t,j}$ that maximizes the function

$$\prod_{i=1}^n (P_{bgo}(x_i) + P_{fgo}(x_i)) \cdot P_{appearance} \quad (17)$$

Since both the appearance observation model Eq(6) and the appearance dynamics Eq(7) are Gaussian functions, the function in Eq(17) becomes a Gaussian mixture. The closed-form solution to this optimization problem is difficult to find. However, appearance is a discrete function and we know the solution should be between the current observation and the previous estimate. For each pixel, we can search for the appearance value in this range to find the solution.

4 Implementation and experimental results

4.1 Initialization and deletion of objects

In addition to the tracking algorithm discussed in the previous sections, there are several other issues regarding the initialization and deletion of the foreground and background layers need to be addressed. In our implementation, change image is computed to determine whether a moving object presents in the scene. A new object is initialized if a change blob is detected far away from any existing objects. In this case, we assume the center of the object is located at the center of the change blob. The value of shape map at each pixel is proportional to the intensity of the change image. The appearance is set to be the original image intensity values. An additional background layer is inserted. The new layer has the same shape map as its neighboring background layer. A normalization step is then applied to make sure these background shape maps sum up to 1 at each pixel.

An object is deleted if it moves out of the image boundaries or it is occluded for a very long period of time. Then the foreground layer of this object is removed from the data structure and two background layers next to it merge into one layer with the shape mask value equal to the sum of the original two shape maps.

4.2 Synthetic videos

We have tested the proposed algorithm using synthetic and real video clips. (Video clips of the results are

available in the supplementary file). In Figure 4 and Figure 5 show the tracking results of two synthetic videos with moving objects. The videos include difficult conditions including shadows, reflections, and transparent objects (e.g. the waterfall), and out-of-plane object rotation. Our tracking algorithm locked on the moving objects successfully through occlusion in both sequences. The estimated state variables in three key frames of the second video are shown in Figure 6. It can be observed the background shape maps in row 3 accurately describe the shape of the occluding tree.

4.3 Vehicle tracking through occlusion

We implemented a tracking system based on the proposed algorithm for handling object occlusions. In Figure 7, the tracking result on a video clip with a car occluded by background is demonstrated. In this example, background objects such as trees, light poles, and the rising ground occlude the car. The proposed tracking algorithm estimates the layer parameters correctly through the sequence. The tracker found one foreground layer and two background layers. The estimated state variables in three key frames are demonstrated in Figure 8. The background shape maps are for the front-most layer.



Figure 6. Layer state variables in three frames of the video in Figure 5 (Row 1 are the original images, Row 2 are the foreground shapes, Row 3 are the background shapes, and Row 4 are the foreground appearances).

4.4 Human tracking

Although our model of layer shape is 2D rigid, our tracker is able to track moving people by adjusting the system parameters and focusing torso area, which is relatively rigid compared to the other parts of human body. Figure 9 shows the tracking result of two persons passing across each other. The algorithm tracked both persons successfully through the occlusion.

Figure 10 demonstrates the tracking results on a video clip in which a walking person is occluded by background objects. Because the occluding background area is large, there is a long period of full occlusion. Since the algorithm estimates the background occluding layers, it knows which part of the foreground is occluded. As a result, the tracker is aware of the occlusion and will not update the object appearance. The tracker is able to regain correct values of layer state soon after the object moves out of the occluding background, as observed in the last frame.



Figure 8. Layer state variables in three frames of the video in Figure 7 (Row 1 are the original images, Row 2 are the foreground shapes, Row 3 are the background shapes, Row 4 are the foreground appearances).

5 Conclusions

A novel motion layer based representation and the associated estimation algorithm have been proposed in this paper. This new approach extends the traditional layer model by introducing the background layers and layer ordering. The experimental results demonstrate the power of this representation in handling the difficult occlusion problem in tracking.

One advantage of the proposed representation is that it models all possible interaction between foreground and background objects. Not only the occlusion caused by the foreground layer is modeled, but also modeled is the occlusion caused by the background layers.

Some future research topics for improving the proposed algorithm include the development of more flexible shape

and motion models that can handle articulated and nonrigid motions and the investigation of efficient optimization algorithms for finding the optimal ordering of the foreground layers.

References

- [1] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, Hierarchical model-based motion estimation, in *Proc. of 2nd European Conference on Computer Vision*, pp. 237-252, 1992.
- [2] M.J. Black, D.J. Fleet., and Y. Yacoob A framework for modeling appearance change in image sequences. *IEEE International Conference on Computer Vision*, Mumbai, India, January 1998, pp. 660-667.
- [3] Black, M. J. and Fleet, D. J., Probabilistic detection and tracking of motion boundaries, *Int. J. of Computer Vision*, 38(3):231-245, July 2000.
- [4] Allan D. Jepson, David J. Fleet, Michael J. Black A Layered Motion representation with occlusion and compact spatial support. *ECCV(1) 2002*: 692-706.
- [5] Allan D. Jepson, David J. Fleet and Thomas F. El-Maraghi, Robust online appearance models for visual tracking *IEEE Conference on Computer Vision and and Pattern Recognition, Kauai*, 2001, Vol. 1, pp. 415-422.
- [6] N. Jovic, N. Petrovic, B. Frey, and T. S. Huang, Transformed hidden Markov models: estimating mixture models of images and inferring spatial transformations in video sequences, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.(II) 26-33, 2000.
- [7] N. Jovic and B.J. Frey Learning flexible sprites in video layers. In *Computer Vision and Pattern Recognition*, pp. (I) 199-206, 2001.
- [8] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2): 257-285, 1989.
- [9] H. Tao, H. Sawhney and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations", *IEEE Transactions On Pattern Analysis And Machine Intelligence*. Jan. 2002.
- [10] N. Vasconcelos and A. Lippman, Emprical Bayesian EM-based motion segmentation, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 527-532, 1997.
- [11] J. Y. A. Wang and Edward H. Adelson, Layered representation for motion analysis, in *Proc. of IEEE conference on Computer Vision and Pattern Recognition*, pp. 361-366, 1993.
- [12] Y. Weiss and E. H. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models, in *Proc. of IEEE conference on Computer Vision and Pattern Recognition*, pp. 321-326, 1996.



Figure 4. A synthetic video sequence with a figure moving horizontally.

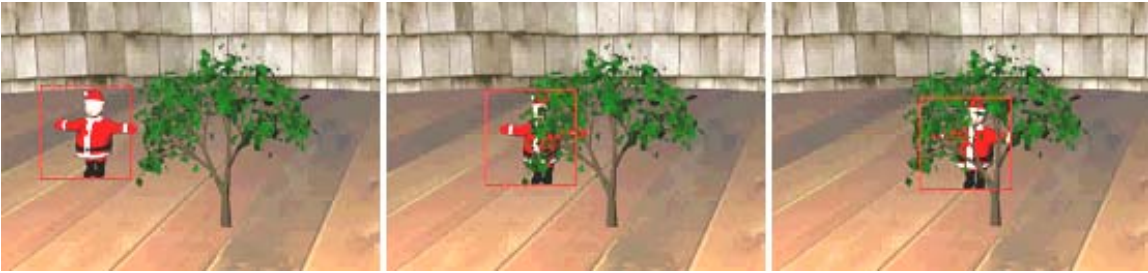


Figure 5. A moving figure moves behind a tree.



Figure 7. A moving car is occluded by trees and the rising ground.



Figure 9. Two people walk across each other.



Figure 10. A person walks behind trees and bushes.