Layered Active Contours for Tracking

Gallagher Pryor, Patricio Vela, Tauseef ur Rehman, and Allen Tannenbaum
Georgia Institute of Technology, Atlanta GA, USA

Abstract

This paper deals with the task of object tracking in the presence of occlusions and clutter by fitting a layered appearance model to data. Four major problems must be overcome: (1) the association of each pixel to a particular layer (layer segmentation), (2) the determination of layer support, (3) the determination of layer appearance, and (4) determination of layer location. Tao, Sawhney, and Kumar successfully proposed a generalized expectation maximization algorithm solving these problems by directly inferring masks representing layer segmentation in conjunction with a deforming elliptical shape prior defining layer support. We extend their work with the introduction of active contours: instead of directly inferring these masks, we evolve a series of curves to obtain a layer segmentation. These curves provide a natural shape prior by constraining segmentations to a family of curves local to layer supports and allow for non-rigid layer deformations through the prediction of unobserved appearance information during inference. A benefit of this extension is the ability to track through massive occlusions and clutter, as demonstrated on a series of difficult real-world video sequences.

1   Introduction

Over the past few years, layered representations have been shown to be powerful tools for motion analysis and tracking. Sometimes referred to as motion segmentation in the literature, fitting a layered representation to imagery consists of segmenting video into regions (layers) containing homogeneous motion (from a given family of motions, e.g. translation), tracking those layers throughout the video, and building an appearance model for each layer. To solve the tracking problem within this framework, these segmentations, motions, and appearances are evolved in a causal fashion to fit imagery from a video stream.

Critical to the success of these algorithms are the constraints placed upon the layer, or motion, segmentations they generate. As such, many different classes of these constraints have been posed within the literature. For instance, layered segmentations performed via clustering have been constrained to meet certain size and smoothness criteria [12, 1]. In [6], similar constraints were proposed in conjunction with a constellation of parts model. In [14] segmentations were constrained to have short, smooth boundaries via graph cuts. Other segmentation constraints have been proposed that utilize a model-based approach involving approximate inference via *generalized expectation maximization* [5, 9, 13, 15]. In [4], a parameterized polybone skeleton shape prior is employed to constrain layered segmentations. Dai, Zheng, and Li [3] utilized a shape constraint specific to the problem of pedestrian detection and tracking. Toyama and Blake proposed a shape constraint facilitated by a learned metric [10]. Tao, Sawhney, and Kumar [8, 17] proposed the use of

an elliptical, probabilistic shape constraint, the most general of the proposed model-based constraints.

Contribution. In our work, we propose a tracking algorithm which extends the framework proposed by Tao, Sawhney, and Kumar. Instead of representing layer segmentations as probabilistic masks (or explicit models as in other works) we propose the use of active contours in their place. We believe that our approach gives four major advantages:

1. The curve representation enforces a general shape prior capable of handling complex shapes, but is still suitable for general tracking. This prior is achieved by constraining layer segmentations to the set of curves local to layers' physical shapes, or support.

2. The use of curves naturally gives rise to a method of handling non-rigid layer deformations through the prediction of unknown layer appearance information beyond the current layer support.

3. The resulting tracker can handle massive occlusions over long periods of time and can track through clutter and other similar targets due to the accurate layer segmentation made possible through **1** and **2**.

4. Unlike other segmentation algorithms such as [7, 14], the method is *causal* and maintains *explicit track points* based on the physical shape of the layers, not the segmentation itself.

We present results on seven video sequences that show this extension significantly improves segmentation and tracking performance over the original tracking algorithm.

Outline. In Section 2, we describe the layered active contour model and accompanying dynamic model. In Section 3, we describe the active layered tracking algorithm. In Section 4, we present results of tracking using a baseline tracker (a version of that proposed in [8]) and the proposed tracker. Finally, in Section 5 we give conclusions followed by directions for future work.

2   The Layered Active Contour Model

We define an instance of a layered representation $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{G}, \mathbf{F}) = (\{C^i, M^i\}_{i=1}^{l-1}, \{\vec{g}^i, f^i\}_{i=0}^{l-1})$ that parameterizes the following family of imagery defined by the function F,

$$F(\vec{r}) = \begin{cases} f^k(\vec{r} - \vec{g}^k), & \vec{r} \in C_{in}^k \text{ and } \vec{r} - \vec{g}^k \in M_{in}^k \\ f^0(\vec{r} - \vec{g}^0), & \text{otherwise} \end{cases} \tag{1}$$

where:

- $l \in Z^+$ is defined as the number of layers utilized within the model, $k \in \{1, 2 ..., l - 1\}$ and,

- $F : \Omega \to R$ where $\Omega \subset R^2$.

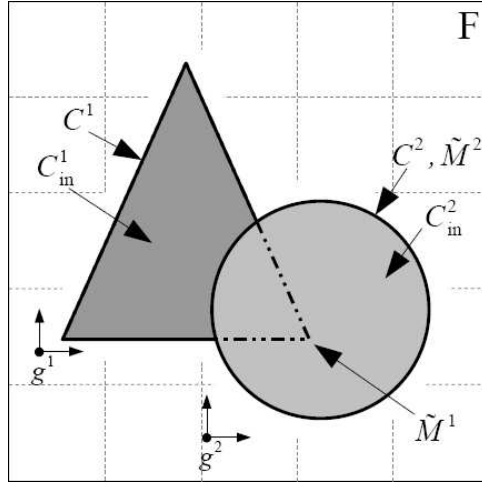The data associated with each layer is:

Figure 1: One parameterization of the layered model with two layers: a triangle f^1 occluded by a circle f^2 as indicated by the relative configuration of the curves C^1, C^2 (describing the visible region of each f) and $\tilde{M}^1, \tilde{M}^2$ (describing the support of each f).

- $\vec{g} \in \Omega$, a layer translation.

- Each C^k is a closed curve in Ω and each C_{in}^k is taken to be the set of all points enclosed by C^k. We enforce the constraint that the region enclosed by each curve is mutually exclusive of all other regions defined by their respective curves.

- Each M^k, like C^k, is a closed curve and, similarly, each M_{in}^k is taken to be the set of all points enclosed by each M^k. Unlike C, the curves M are not considered to be mutually exclusive since they each lie in different coordinate systems.

- Each $f^k : \Omega \to R$ represents the appearance of the kth layer in the model; f^0 is assumed to be the appearance of the background of the model.

The function F describes all imagery composed of l independently translated 2D signals, each occluding one another. Figure 1 illustrates one member of this family, an image of a triangle (layer one) occluded by a circle (layer two). The 2D appearances of these layers are given by f^1 and f^2 with their respective shapes defined by M^1 and M^2. These objects are translated by the offsets $\vec{g}^1$ and $\vec{g}^2$ (resulting in each $\tilde{M}$), respectively, to make up the final output F. The curves C^1 and C^2 mark the boundary of the visible regions of each appearance function in F while $\tilde{M}^1$ and $\tilde{M}^2$ give the total support or shape of each layer. Note how the curve C^1 stops short of the circle, yet $\tilde{M}^1$ continues inward to completely describe the triangle, which indicates an occlusion.

In contrast to the model of [7], the proposed model utilizes curves instead of masks to describe layer segmentation and support. Curves have been proven to be particularly suited to the problem of segmentation while their evolution has been shown to yield sensible curve deformations making them ideal for facilitating flexible layers. Thus, through the introduction of the curves **C**, we leverage a proven solution to the subproblem of segmentation within layered decomposition, and through the introduction of the curves **M**, we constrain the curves **C** to a well-studied and sensible family of shapes. This approach also leads us to develop a completely maximum likelihood solution. In contrast to

generalized expectation maximization, our method does not rely on the maintenance of distributions over time.

2.1 Dynamic Layered Model

Using the function F defined above as an observation model, we define the likelihood of an observed set of images as $P(\mathbf{I}|\{\Sigma^t\}_{t=1}^T)$, where $\mathbf{I}$ is defined as $\{I^t\}_{t=1}^T$, a set of frames in a video sequence. We make a Markov assumption on the states Σ^t and consider each I^t to be conditionally independent given Σ^t, allowing us to factor the distribution,

$$P(\mathbf{I}|\{\Sigma^t\}_{t=1}^T) = \left[\prod_{t=2}^T P(I^t|\Sigma^t)P(\Sigma^t|\Sigma^{t-1})\right] P(I^1|\Sigma^1)P(\Sigma^1), \tag{2}$$

where $P(I^t|\Sigma^t)$ is taken to be the observation model,

$$P(I^t|\Sigma^t) = \frac{1}{Z}\exp\left[-\frac{1}{2\sigma}\int_\Omega (F(\vec{r})-I(\vec{r}))^2 d\vec{r}\right], \tag{3}$$

and $P(\Sigma^t|\Sigma^{t-1})$ is taken to be the state dynamics model,

$$P(\Sigma^t|\Sigma^{t-1}) = P(\mathbf{G}^t|\mathbf{G}^{t-1}) = \prod_{i=0}^{l-1} \frac{\exp\left[-\frac{1}{2\sigma_g}(\vec{g}^{i,t}-\vec{g}^{i,t-1})^T(\vec{g}^{i,t}-\vec{g}^{i,t-1})\right]}{2\pi\sigma_g}, \tag{4}$$

To ensure that the posterior distribution $P(g^{i,t})$ remains Gaussian, we assume that $P(\vec{g}^{i,1}) = \delta(\vec{r}-\vec{g}^{i,1})$. We also assume that each f^i and M^i are fixed, independent of time.

3 Tracking via the Layered Model

To perform tracking via inference in the given probabilistic model, one must select an initial parameterization of F, or model state Σ^1, and evolve that state so as to maximize its likelihood given a series of observations $\mathbf{I}$. To do so, we treat this task as a filtering problem, assuming a causal series of observations, and adjusting each parameter set $\{\vec{g}^{i,t}, C^{i,t}\}_{i=0}^{l-1}$ in turn while recursively updating each $\{f\}_{i=0}^{l-1}$ accordingly with the new imagery.

3.1 Initialization

For this work, we assume a given initial segmentation as sufficient information to seed the algorithm. An initial segmentation consists of $l-1$ disjoint curves representing the initial visible regions of each layer to be tracked and an accompanying input image I^1. Each curve defines the state information $\mathbf{C}$ and $\mathbf{M}$ and the image information enclosed by each curve defines each $\mathbf{F}$. The state information $\mathbf{G}$ is defined to be zero, initially, essentially fixing a series of independent coordinate systems for each layer.

3.2 Tracking

Upon the arrival of each new observation I^t, we update the layered approximation to maximize the log likelihood of the current observation given the previous state. In doing

so, we also solve for the track point of each layer. Using the definitions we have presented up until now, this process reduces to the minimization of the following energy with respect to $\mathbf{G}, \mathbf{C}$, and $\mathbf{F}$,

$$E(\mathbf{G}, \mathbf{C}, \mathbf{F}) = \frac{1}{2\sigma} \int_\Omega (F(\vec{r}) - I^t(\vec{r}))^2 d\vec{r} + \sum_{i=0}^{l-1} \frac{1}{2\sigma_g} (\vec{g}^{i,t} - \vec{g}^{i,t-1})^T (\vec{g}^{i,t} - \vec{g}^{i,t-1}), \quad (5)$$

where the first term of this energy forces the approximation F to match the current observation I^t while the second term enforces a penalty on large track signal jumps from frame to frame weighted by the model parameter σ_g.

In the derivation that follows, the updates on each parameter set $\mathbf{G}, \mathbf{C}, \mathbf{F}$ of F are derived on a layer by layer basis to minimize this energy. In the cases of $\mathbf{G}$ and $\mathbf{F}$, a closed-form minimization is performed, but in the case of $\mathbf{C}$ and $\mathbf{M}$, which are curves, we apply gradient descent.

3.2.1 Solving for $\vec{g}^{i,t}$

Due to the fact that we make no assumptions about the functions f (smoothness, etc), the landscape of E is possibly complex with many local minima with respect to each $\vec{g}^{i,t}$. Additionally, depending on the value of the externally defined σ_g, the minimizer $\vec{g}^{i,t}$ will always lie within a set distance of $\vec{g}^{i,t-1} \in \Omega$, defining a region Ω'. Thus, we minimize E with respect to $\vec{g}^{i,t}$ through direct search over Ω',

$$\vec{g}^{i,t} = \underset{\vec{g}^{i,t}}{\operatorname{argmin}} \left[\int_\Omega (F(\vec{r}, \tau) - I(\vec{r}, \tau))^2 d\vec{r} + \frac{1}{2\sigma_g} (\vec{g}^{i,t} - \vec{g}^{i,t-1})^T (\vec{g}^{i,t} - \vec{g}^{i,t-1}) \right] \quad (6)$$

3.2.2 Solving for $C^{i,t}$

To solve for $C^{i,t}$ we perform gradient descent starting from the translated curve $\tilde{M}^i$ as an initial guess. In doing so, we constrain the curve $C^{i,t}$ to lie in the local neighborhood of $\tilde{M}^i$, enforcing the constraint that the shape of layer i's visible region is within a set distance of the underlying support of that layer. The size of this neighborhood is determined by the number of gradient descent iterations we perform on $C^{i,t}$, n. Additionally, by evolving $C^{i,t}$ according to a local flow, we simulate the event of an occlusion or shape change of a layer: starting from the initial layer support M^i, $C^{i,t}$ must move pixel by pixel either inwards or outwards along its normals as exactly would be seen during an occlusion or shape change of the layer. The mirroring of this physical process in our inference step, we believe, justifies our technique of providing a shape constraint within our model.

To perform gradient descent on $C^{i,t}$, we find an energy-minimizing flow on $C^{i,t}$ by taking the first variation of the energy E with respect to $C^{i,t}$. For the sake of exposition, we begin with the simple case when $l = 2$ where only one curve is present within the model. In this case, the energy E simplifies down to a competition between two quadratic terms:

$$E(\mathbf{C}) = \int_{C_{in}^{1,t}} (f^1(\vec{r} - \vec{g}^{1,t}) - I^t(\vec{r}))^2 d\vec{r} + \int_{C_{out}^{1,t}} (f^0(\vec{r} - \vec{g}^{0,t}) - I(\vec{r}))^2 d\vec{r} \quad (7)$$

Applying Green's theorem and taking the first variation of this simplified energy with respect to C^1 gives the following flow for the curve,

$$\frac{\partial C^{1,t}(\tau)}{\partial \tau} = \vec{N}[(f^1(\vec{r} + \vec{g}^{1,t}) - I^t(\vec{r}))^2 - (f^0(\vec{r} + \vec{g}^{0,t})) - I^t(\vec{r}))^2], \quad (8)$$

where τ is the typical artificial time parameter used in conjunction with gradient descent and $\vec{N}$ is the normal of the curve $C^{1,t}$. The flow (8) is related to that proposed by Chan and Vese [2], but is unique with its more specific appearance model. Heuristically, this flow expands $C^{1,t}$ along its normal wherever I^t is more similar to layer one than layer zero and vice versa. Note that no regularizing terms are present in this flow: we do not expect the visible region of an object to have any special properties, though it makes sense to assume so for the layer support M^i.

In the general case where $l > 2$, the energy E simplifies into a competition between $l-1$ curves:

$$E(\mathbf{C}) = \sum_{i=1}^{l-1} \int_{C_{in}^{i,t}} (f^i(\vec{r} - \vec{g}^{i,t}) - I^t(\vec{r}))^2 d\vec{r} + \int_{\bigcap_{i=1}^{l-1} C_{out}^{i,t}} (f^0(\vec{r} - \vec{g}^{0,t}) - I^t(\vec{r}))^2 d\vec{r}, \quad (9)$$

where $\bigcap_{i=1}^{l-1} C_{out}^{i,t}$, defines the region(s) in Ω exterior to all curves in the model. To determine the minimizing curves $\mathbf{C}$, the variation of each curve must be taken and gradient descent carried out while enforcing the constraint that all $\mathbf{C}$ must be mutually disjoint. Several researchers have studied this problem and many solutions exist to satisfy this constraint including the addition of an additional energy term with Lagrange multiplier [16] or utilizing a modified level set representation [11]. However, these solutions result in the same type of flow as in the simplified, two layer case where all curves move along their normals, competing against one another according to a quadratic matching term with I^t.

3.2.3 Solving for f^i

To solve for an updated version of each f^i, we take the derivative of the energy, set to zero, and solve for f^i. Note that according to our dynamical model, f^i is assumed to be independent of time. Thus, the error minimizing value for f^i is computed to be a time average of all observed data within the bounds of each $\{C^{i,k}\}_{k=1}^t$,

$$f^i(\vec{r}) = \frac{\sum_{k=1}^t I^k(\vec{r} - \vec{g}^{i,k}) H^{i,k}(\vec{r} - \vec{g}^{i,k})}{\sum_{k=1}^t H^{i,k}(\vec{r} - \vec{g}^{i,k})}, \quad (10)$$

where we define $H^{i,k}$ to be the Heaviside function indicating the enclosed region of the curve $C^{i,k}$,

$$H^{i,k}(\vec{r}) = \begin{cases} 1, & \vec{r} \in C_{in}^{i,k} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

3.3 Flexible Layers via Appearance Prediction

Due to the construction of the layered observation model where layers occlude one another and have flexible support, it may be impossible to observe values for the appearance functions $\mathbf{F}$ in the given data. For instance, though each observation I^t has support over Ω, the layer appearance functions $\mathbf{F}$ initially contain information only on the supports defined by $\mathbf{M}$ which are typically a strict subset of Ω. As such, any value of f^i beyond this support has yet to be observed and is considered unknown information. Information in the background may also be unknown due to occluding foreground layers.

Unfortunately, while solving for $C^{i,t}$, an estimate of this unknown information is necessary for inference as it is possible for $C^{i,t}$ to grow outward beyond M^i (in the case of
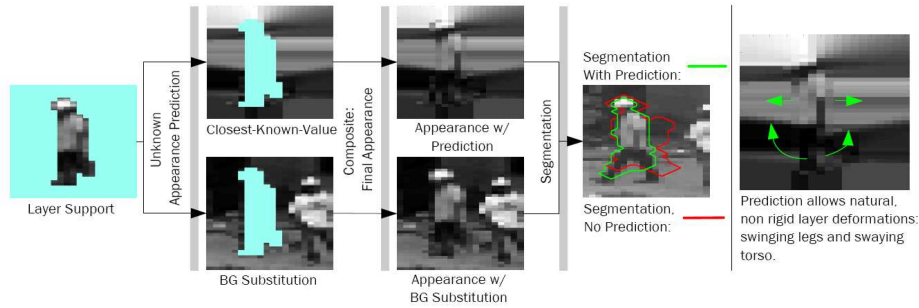
Figure 2: Flexible layers are facilitated by closest-known-value prediction of unknown appearance that allow natural non-rigid layer deformations including the swinging of the worker's legs and torso movement. This leads to improved layer segmentation over using no prediction.

layer shape change) or away from occluded background regions (in the case where a layer translates away from its initial position). We handle this problem while at the same time implicitly imposing a shape constraint on the model's appearances and supports by adhering to the closest-known-value principle: we define each unknown value $f^i(\vec{r})$ to be equal to the closest known, or observed, value f^i to $\vec{r}$,

$$f^{i\prime}(\vec{r}) = \begin{cases} f^i(\vec{r}), & \sum_{k=1}^{t} H^{i,k}(\vec{r}) > 0 \\ f^i(\mathrm{argmin}_{\vec{x} \in M^i} \|\vec{x} - \vec{r}\|), & \text{otherwise} \end{cases} \qquad (12)$$

This technique is often used in the active contour community to facilitate what are known as velocity or data extensions whenever information is required off of an interface to evolve a higher-dimensional embedding function. We use this technique, here, to enforce two assumptions: **(1)** Initially unobserved information due to occlusion is expected to be equal to its closest known neighbor and is therefore geometrically consistent and **(2)** when layers change shape, we expect edge appearance information to be attached to the edges of the shape. Through this prediction technique, many common shape and appearance changes can be handled (see Figure 2).

4 Results

4.1 Experimental Setup

For comparison purposes, we implemented a version of the tracker proposed by Tao, Sawhney, and Kumar as the baseline tracker. We then extended this implementation so as to include the active contour model we have presented, leading to a change regarding *only* the layer segmentation calculation and unknown appearance information. All other inference steps and model parameters remained fixed. Because our focus is on tracking, we used a consistent hand segmentation for initialization of both algorithms, though both can be initialized automatically with no drawbacks [8].

All experiments were carried out by running both the trackers on seven 128x128 video sequences. On average, each video sequence consisted of 300 frames or 10 seconds of
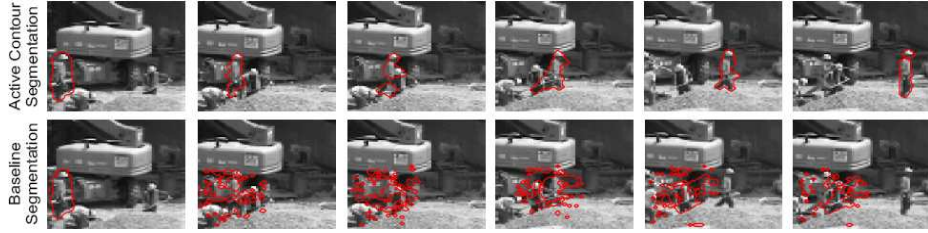
Figure 3: Tracking and segmentation results on video 5 from Table 1. The red line marks the layer segmentation curve (C^1) in the first row and the 50% probability isocontour of the layer segmentation in the second row. The active contour tracker maintains an accurate layer segmentation and track signal in the presence of clutter and similar moving targets.
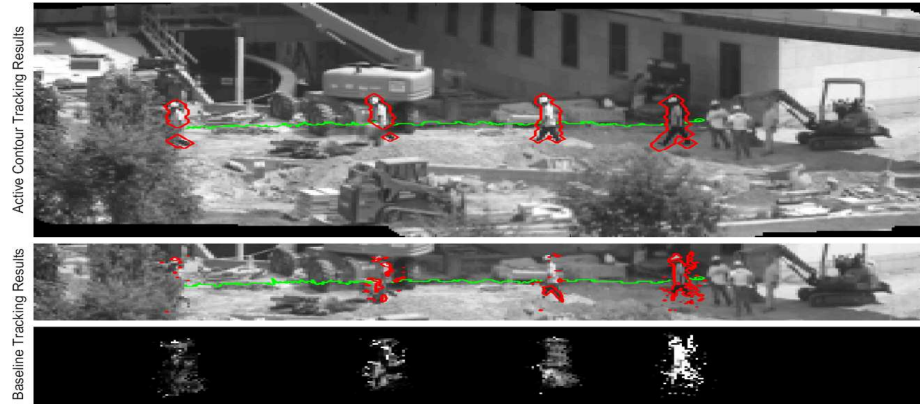

Figure 4: Example of tracking a construction worker with a left-panning camera and changing lighting conditions. Here, the final appearance, or panorama, of the background layer is overlaid with layer one's visible region indicated by the curve C^1 (red) from frames 12, 109, 228, and 387 and track signal (green). The bottom two rows give the baseline tracker's track signal, the red contour indicating the 50% probability isocontour of the layer segmentation and the last row indicating the likelihood of each pixel belonging to the foreground layer. The active contour tracker maintains a correct layer segmentation and an accurate track signal.

imagery. To generate ground truth track signals for comparison, a hand-picked series of "correct" track points were chosen on each frame for each layer on each video. Performance results were generated by comparing the tracker's output track signals for each layer, $q_{x,y}^t$, against the hand-picked track points, or ground truth signal $r_{x,y}^t$.

4.2 Analysis

A quantitative comparison between the layered active contour tracker and the baseline tracker is given in Table 1 using the performance metrics described above on a single foreground layer's track signal. Unlike the baseline tracker, the active contour tracker maintained track through all seven sequences. In the cases both trackers maintained track, the active contour tracked with less average error.
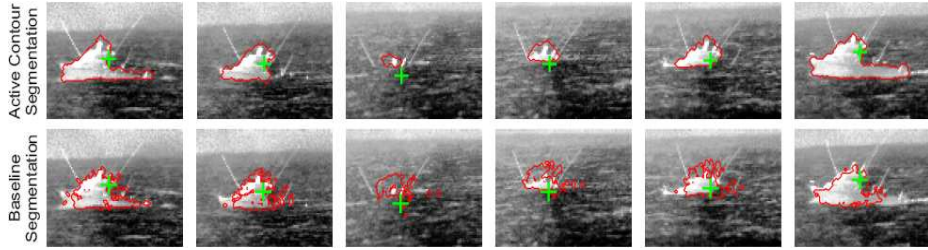
Figure 5: Tracking and segmentation results of tracking a boat in the open ocean. As the boat sails perpendicular to the line of sight, it is almost totally obscured by a passing swell of water for approximately 100 frames. The red line marks the layer segmentation curve (C^1) in the second row and the 50% probability isocontour of the layer segmentation in the third row. The green cross marks the track point that, for both trackers, stays on target. The active contour maintains a correct layer segmentation even through the massive occlusion.

Tracker Performance:
Layered Active Contour vs Baseline Tracker

Videos							
Active Contour **%**	100%	100%	100%	100%	100%	100%	100%
ε	1.25	0.68	0.81	1.66	1.97	0.76	1.03
Baseline **%**	78%	100%	100%	100%	49%	97%	100%
ε	**LT**	0.71	0.85	3.97	**LT**	0.9	1.23

Table 1: Percent of time on track (**%**) and mean track error (ε) of the layered active contour model versus the baseline layered tracker. **LT** Indicates total track loss.

The most clear advantage the active contour had was on video 5. Here, a track subject moves amongst a great deal of clutter and dust, with similar targets in proximity to the tracking subject (Figure 3). The more accurate active contour segmentation (as opposed to a mask) allowed the tracker to avoid learning an incorrect layer appearance and thus handle the ambiguous case of crossing workers. In each of the other videos, the active contour maintains an accurate segmentation and track signal (Figures 4 and 5).

5 Conclusions

In this paper, we have developed a tracker based upon a layered active contour model as an extenstion to [8]. In place of representing layer segmentation as a series of masks, we utilized curves that are evolved to fit data. We showed that as a result of this extension, a general shape prior is introduced into the model and a natural way of handling non-rigid layer deformations is introduced through the prediction of unknown appearance information. Finally, we presented results on several difficult video sequences demonstrating the benefits of this addition to the model.

References

[1] M. Allen, M. K. Titsias, and C. K. I. Williams. Fast learning of sprites using invariant features. In *Proc. British Machine Vision Conference*, 2005.

[2] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, pages 266–277, February 2001.

[3] C. Dai, Y. Zheng, and X. Li. Layered representation for pedestrian detection and tracking in infrared imagery. In *Proc. IEEE CVPR*, volume 3, page 13, June 2005.

[4] A. D. Jepson, D. J. Fleet, and M.J. Black. A layered motion representation with occlusion and compact spatial support. In *Proc ECCV*, pages 692–706, 2002.

[5] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proc. IEEE CVPR*, volume 1, pages 199–206, Maui, HI., 2001.

[6] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentation of video. In *Proc. IEEE ICCV*, volume 1, pages 33–40, 2005.

[7] Nikos Paragios and Rachid Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Comput. Vis. Image Underst.*, 97(3):259–282, 2005.

[8] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with Bayesian estimation of dynamic layer representations. *IEEE Trans. PAMI*, 24:75–89, 2002.

[9] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Trans. PAMI*, 23:297–303, March 2001.

[10] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. Computer Vision*, pages 9–19, 2002.

[11] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. Technical Report 01-25, University of California, Los Angeles, September 2001.

[12] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. IEEE CVPR*, pages 361–366, New York, June 1993.

[13] C. K. I. Williams and M. K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16:1039–1062, 2004.

[14] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Trans. PAMI*, 27:1644–1659, October 2005.

[15] H. Yalcin, M.J. Black, and R. Fablet. The dense estimation of motion and appearance in layers. In *Proc. IEEE CVPR*, page 165, June 2004.

[16] H.-K. Zhao, T. Chan, M. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, August 1996.

[17] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *Proc. IEEE ICCV*, volume 2, pages 1079–1085, October 2003.