# ADAPTIVE FRAGMENTS-BASED TRACKING OF NON-RIGID OBJECTS USING LEVEL SETS

*Prakash Chockalingam and Nalin Pradeep*

Electrical and Computer Engineering Department, Clemson University

## ABSTRACT

We present an approach to visual tracking based on dividing a target into multiple regions, or fragments. The target is represented by a Gaussian mixture model in a joint feature-spatial space, with each ellipsoid corresponding to a different fragment. The fragments are automatically adapted to the image data, being selected by an efficient region-growing procedure and updated according to a weighted average of the past and present image statistics. Modeling of target and background are performed in a Chan-Vese manner, using the framework of level sets to preserve accurate boundaries of the target. The extracted target boundaries are used to learn the dynamic shape of the target over time, enabling tracking to continue under partial and total occlusion. Experimental results on a number of challenging sequences demonstrate the effectiveness of the technique.

## 1. INTRODUCTION

Recent interest in visual tracking has centered around on-line learning of multiple cues to adaptively select the most discriminative ones. With this focus, significant progress has been achieved by algorithms such as those of Avidan [2], and Collins et al [6]. In these approaches, tracking is formulated as a classification problem in which the probability of each pixel belonging to the target is computed. While the results have been impressive, several limitations remain:

- Although the tracker locks onto the most discriminative cue, it ignores important but secondary cues. This is because the distribution is modeled as unimodal. For example, the model may capture the skin of a person's face, but not the hair.
- These algorithms produce a strength image indicating the probability of each pixel belonging to the object being tracked, but they provide no mechanism for determining the shape of the object. And without a multimodal distribution, the strength image does not make this possible.
- Occlusion of the target can cause the learner to adapt to occluding surfaces, thus causing the model to drift from the target. If the occlusion is long enough, this can lead to tracking failure. An explicit representation of the contour would enable such errors to be prevented.
- Spatial information that captures the joint probability of pixels is often ignored. This leads to an impoverished tracker that is not able to take advantage of the wealth of information available in the spatial arrangement of the pixels in the target.

In this paper we present a technique that overcomes these limitations. Like Adam et al [1], we split the target into a number of fragments to preserve the spatial relationships of the pixels. Unlike their work, however, our fragments are adaptively chosen according to the image data, by clustering pixels with similar appearance, rather than using a fixed arrangement of rectangles. This adaptive fragmentation captures all the secondary cues and also ensures that each fragment captures a single mode of the distribution. We classify individual pixels, as in [2, 6], but by incorporating multiple fragments we are better able to preserve the shape of multimodal targets. The boundary is represented by a level set using a Chan-Vese [5] model that enables level set tracking to be formulated in a Bayesian manner and leads to more stable convergence of the algorithm. To address the problem of drastically moving targets with untextured regions, the recently proposed approach of [3] is employed to impose a global smoothness term in order to produce accurate sparse motion flow vectors for each fragment. The fragment models are updated automatically using the estimated contour and the image data, and the previous shapes are used to track the object through partial and total occlusion.

## 2. APPROACH

To represent the target being tracked, we use the formulation of level sets due to their numerical stability and their ability to accurately represent a generic contour [9, 4]. Let $\Gamma(s) = [\, x(s) \quad y(s) \,]^T$, $s \in [0, 1]$, be a closed curve in $\mathbb{R}^2$, and define an implicit function $\phi(x, y)$ such that the zeroth level set of $\phi$ is $\Gamma$, i.e., $\phi(x, y) = 0$ if and only if $\Gamma(s) = [x, y]^T$ for some $s \in [0, 1]$. Let $R^-$ be the region inside the curve (where $\phi > 0$) and $R^+$ the region outside the curve (where $\phi < 0$).

Our goal is to estimate the contour from a sequence of images. Let $I_t : \mathbf{x} \rightarrow \mathbb{R}^m$ be the image at time $t$ that maps a pixel $\mathbf{x} = [x \quad y]^T \in \mathbb{R}^2$ to a value, where the value is a scalar in the case of a grayscale image ($m = 1$) or a three-element vector for an RGB image ($m = 3$). The value could also be a larger vector resulting from applying a bank of texture filters to the neighborhood surrounding the pixel, or some combination of these raw and/or preprocessed quantities. We use Bayes' rule and an assumption that the measurements are independent of each other and of the dynamical process to model the probability of the contour $\Gamma$ at time $t$ given the previous contours $\Gamma_{0:t-1}$ and all the measurements $I_{0:t}$ of the causal system as

$$p(\Gamma_t | I_{0:t}, \Gamma_{0:t-1}) \propto \underbrace{p(I_t^+ | \Gamma_t)}_{\text{target}} \ \underbrace{p(I_t^- | \Gamma_t)}_{\text{background}} \ \underbrace{p(\Gamma_t | \Gamma_{0:t-1})}_{\text{shape}}, \tag{1}$$

where $I_t^+ = \{\xi_I(\mathbf{x}) : \mathbf{x} \in R^+\}$ captures the pixels inside $\Gamma_t$, $I_t^- = \{\xi_I(\mathbf{x}) : \mathbf{x} \in R^-\}$ captures the pixels outside $\Gamma_t$, and $\xi_I(\mathbf{x}) = [\mathbf{x}^T \quad I(\mathbf{x})^T]^T$ is a vector containing the pixel coordinates coupled with their image measurements.

### 2.1. Fragment modeling

Assuming conditional independence among the pixels, the joint probability of the pixels in a region is given by

$$p(I_t^\star | \Gamma_t) = \prod_{\mathbf{x} \in R^\star} p_\star(\xi_I(\mathbf{x}) | \Gamma_t), \tag{2}$$

where $\star \in \{-, +\}$. One way to represent the probability of a pixel $\xi_I(\mathbf{x})$ is to measure its signed distance to a separating hyperplane in $\mathbb{R}^n$, where $n = m + 2$, as in [2, 6], or using a single covariance matrix, as in [10]. A slightly more general approach would be to measure its Mahalanobis distance to a pair of Gaussian ellipsoids representing the target and background. None of these approaches, however, is able to capture the subtle complexities of multi-modal regions. As a result, we instead represent both the target and background appearance using a set of *fragments* in the joint feature-spatial space, where each fragment is a separate Gaussian ellipsoid. Letting $\mathbf{y} = \xi_I(\mathbf{x})$ for brevity, the likelihood of an individual pixel is then given by a Gaussian mixture model (GMM):

$$p_\star(\mathbf{y} | \Gamma_t) = \sum_{j=1}^{k_\star} \pi_j p_\star(\mathbf{y} | \Gamma_t, j), \tag{3}$$

where $\pi_j = p(j | \Gamma_t)$ is the probability that the pixel was drawn from the $j$th fragment, $k_\star$ is the number of fragments in the target or background, $\sum_{j=1}^{k_\star} \pi_j = 1$, and

$$p_\star(\mathbf{y} | \Gamma_t, j) = \eta \exp\left\{ -\frac{1}{2} (\mathbf{y} - \mu_j^\star)^T \left(\Sigma_j^\star\right)^{-1} (\mathbf{y} - \mu_j^\star) \right\}, \tag{4}$$

where $\mu_j^\star \in \mathbb{R}^n$ is the mean and $\Sigma_j^\star$ the $n \times n$ covariance matrix of the $j$th fragment in the target or background model (depending upon $\star$), and $\eta$ is the Gaussian normalization constant.

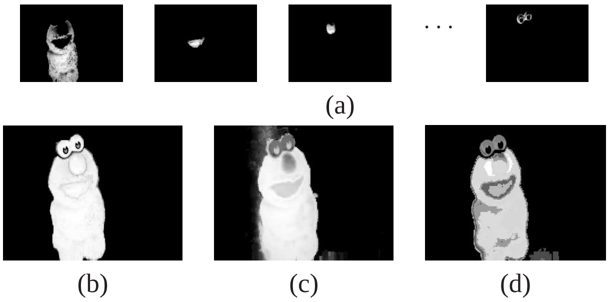


**Fig. 1**. (a) Probabilities determined by individual fragments are combined to compute (b) our strength image. For comparison, the strength image computed using (c) a single Gaussian [10] and (d) a linear separation over a linear combination of multiple color spaces [6] are also shown. Our fragment-based GMM representation more effectively represents the multi-colored target.

### 2.2. Computing the strength image

We follow the recent approach of formulating the object tracking problem as one of binary classification between target and background pixels [2]. In this approach, a strength image is produced indicating the probability of each pixel belonging to the target being tracked. The strength image is computed using the log ratio of the probabilities:

$$S(\mathbf{x}) = \log\left(\frac{p_+(\mathbf{x})}{p_-(\mathbf{x})}\right) = \Psi^-(\mathbf{x}) - \Psi^+(\mathbf{x}), \tag{5}$$

where $\Psi^\star(\mathbf{x}) = -\log p_\star(\mathbf{x})$. Positive values in the strength image indicate pixels that are more likely to belong to the target than to the background, and vice versa for negative values. An example strength image is shown in Figure 1, illustrating the improvement compared with [10] and [6]. The strength image is used to update the implicit function, which enables the level set machinery to enforce smoothness on the resulting object shape.

### 2.3. Segmentation

Our fragment-based representation of the target is similar to that of Adam et al [1] but with two significant differences. First, we use fragments to model the background as well as the target, and secondly, our fragments are automatically determined and adapted by the image data rather than being fixed and hard-coded. The challenge is to compute the model parameters $\mu_1^+, \ldots, \mu_{k_+}^+, \Sigma_1^+, \ldots, \Sigma_{k_+}^+, \mu_1^-, \ldots, \mu_{k_-}^-, \Sigma_1^-, \ldots, \Sigma_{k_-}^-$ from the current contour $\Gamma_t$. This is essentially a problem of segmentation. We tried the graph-based algorithm of [8] but found it to unacceptably merge regions with distinct colors. We also experimented with mean-shift segmentation [7], but it was not only too slow for a tracking application but it also tended to oversegment the image.

Instead, we implemented a region-growing algorithm. Initially a pixel in the image is selected at random, and a single fragment is created to hold the pixel. Neighboring pixels are added to the segment if they are within $\tau$ standard deviations of the Gaussian model of the fragment, with an appropriate relaxing of the threshold for small regions that do not yet have enough pixels for their model to be reliable. The mean $\mu_j^\star$ and covariance $\Sigma_j^\star$ are updated efficiently using a running accumulation of first- and second-order statistics. Once the fragment has finished growing, a new pixel is selected at random, and the procedure is repeated for a new fragment. This process continues until all pixels have been added to a fragment, at which point small fragments are discarded and the remaining fragments are labeled as target or background depending upon whether the majority of pixels are within or without a manually drawn initial contour $\Gamma_0$, respectively. Any fragment for which the pixels are roughly evenly distributed is split along $\Gamma_0$ to form two fragments, one labeled foreground and the other labeled background. Finally, we choose $\pi_j$ based on the size of the fragments.

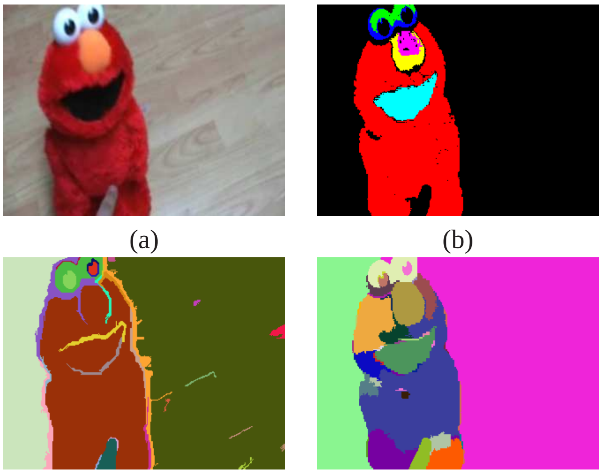


**Fig. 2**. (a) Image of Elmo. (b) Foreground regions. Output of competing algorithms (c) Graph-based segmentation [8] accidentally merges regions with distinct colors and (d) Mean-shift segmentation [7], even with a large scale parameter, oversegments the image.

## 2.4. Level set formulation

Maximizing the probability of (1) is equivalent to minimizing the following energy functional over the level set function [5]:

$$E(\phi) = \int_{R^+} \Psi^+(\mathbf{x})d\mathbf{x} + \int_{R^-} \Psi^-(\mathbf{x})d\mathbf{x} + \mu\ell(\Gamma), \quad (6)$$

where $\mu$ is a scalar that weights the relative importance of the shape term, which is assumed for the moment to consist only in measuring $\ell(\Gamma)$, the length of the curve. At this point we introduce the regularized Heaviside function $H(z) = \frac{1}{1+e^{-z}}$ as a differentiable threshold operator to rewrite the above as

$$E(\phi) = \int_\Omega H(\phi)\Psi^+(\mathbf{x}) + (1-H(\phi))\Psi^-(\mathbf{x}) + \mu|\nabla H(\phi)|d\mathbf{x}, \quad (7)$$

where $\ell(\Gamma) = \int_\Omega |\nabla H(\phi)|d\mathbf{x}$, and $\Omega = R^+ \cup R^-$ is the image domain. With $E = \int_\Omega F(x, y, \phi, \phi_x, \phi_y)d\mathbf{x}$, the associated Euler-Lagrange equation is given by

$$\begin{aligned} 0 &= \frac{\partial F}{\partial \phi} - \frac{\partial}{\partial x}\left[\frac{\partial E}{\partial \phi_x}\right] - \frac{\partial}{\partial y}\left[\frac{\partial E}{\partial \phi_y}\right] \\ &= h(\phi)\left(\Psi^+(\mathbf{x}) - \Psi^-(\mathbf{x}) - \mu\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right), \end{aligned}$$

where $\phi_x = \partial\phi/\partial x$, $\phi_y = \partial\phi/\partial y$, $h(\phi) = \partial H/\partial\phi$, and $\nabla\phi = [\,\phi_x \quad \phi_y\,]^T$ is the gradient of $\phi$. To avoid the difficulty of solving this PDE explicitly for $\phi$, we instead take the value on the left-hand side as an indication of the error, and apply gradient descent iterations [5] with

$$\phi^{(k+1)} = \phi^{(k)} + |\nabla\phi|\left(\Psi^-(\mathbf{x}) - \Psi^+(\mathbf{x}) + \mu\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right), \quad (8)$$

where $k$ is the iteration number, div is the divergence operator, and we have used the approximation $h(\phi) \approx |\nabla\phi|$, which is accurate as long as the level set function is smooth away from the boundary. The sign in the equation comes from the convention that $\phi > 0$ inside the boundary.

Note that unlike the traditional level set formulation, ours is not based upon intensity edges. Rather, we have adopted the Chan-Vese approach [5] of modeling the foreground and background regions explicitly. This approach results in a large basin of attraction, so that the iterations above will converge to the target from a wide variety of initial curves, without being significantly distracted by local noise in the data. Since the curve evolution is not required to be monotonic, the initial curve may be inside the target, outside the target, or some combination of the two.

## 2.5. Fragment motion

While the minimization above is not extremely sensitive to the initial contour, nevertheless it is beneficial for the coordinate systems of the target and the model fragments to be approximately aligned. Such alignment increases the accuracy of the strength image, due to the use of spatial information in the joint spatial-feature vectors. As a result we seek to recover, *prior* to computing the strength image, approximate motion vectors between the previous and current image frame for each fragment: $\mathbf{u}_i^\star = (u_i^\star, v_i^\star), i = 1, \ldots, k^\star$.

To find the motion vectors, we utilize the recent joint feature tracking approach [3] to track feature points. Once the feature points have been tracked, the motion vector of each fragment is computed by averaging the motions of the features within the fragment. Note that there is little risk to this averaging, since outliers are avoided by the smoothness term incorporated by the joint Lucas-Kanade approach, which enables features to be tracked even in untextured areas, as shown in [3]. Feature selection is determined by those image locations for which $\max(e_{\min}, \eta e_{\max})$, where $e_{\min}$ and $e_{\max}$ are the two eigenvalues of the $2 \times 2$ gradient covariance matrix, and $\eta < 1$ is a scaling factor.

## 3. EXPERIMENTAL RESULTS

The algorithm was tested on a number of challenging sequences captured by a moving camera viewing complex scenery. Most of the sequences presented here are chosen so that the tracker can be evaluated for objects undergoing significant scale changes, extreme shape deformation, and unpredictable motion. Some of these sequences were obtained from Internet sources, with high compression, to demonstrate the performance of the algorithm even in poor quality videos. Figure 3 shows the results of the algorithm on a sequence of a Tickle Me Elmo doll and a monkey sequence. In the monkey sequence, as the monkey swings around the tree, it undergoes a drastic shape change in just a few image frames. Yet the algorithm is able to remain locked onto the target, as well as compute an accurate outline of the animal. For comparison, we have also presented the output of FragTrack [1], which, even with its search range set to the maximum allowable value, loses the target around frame 150 and never recovers.

Additional results involving occlusion are displayed in Figure 4. In our approach, the shape of the object contour is learned over time by retaining the output of the tracker in each image frame. To detect occlusion, the rate of decrease in the object size is determined over the previous few frames. Once the object is determined to be occluded, a search is performed in the learned database to find the contour that most closely matches the one just prior to the occlusion using a Hausdorff distance. Then as long as the target is not visible, the subsequent sequence of contours occurring after the match is used to hallucinate the contour. Once the target reappears, tracking resumes. This approach prevents tracker failure during complete occlusion and predicts accurate contours when the motion is periodic. The first row in the figure shows a sequence where the person is completely occluded by a tree. Our approach predicts both the shape and the location of the object and displays the contour accordingly. The second row shows a more complex scenario where a girl, moving quickly in a circular path (a complete revolution occurs in just 35 frames), is occluded frequently by a boy. The third row shows the results of tracking multiple fish in a tank. The fish are multicolored and swim in front of a complex, textured, multicolored background. The final row shows one of the applications of tracking objects with contours - object recognition. Here we use the shape information obtained from the object contour to recognize the objects being tracked by matching them with a database which consists of shape information of different objects.

## 4. CONCLUSION

We have presented a tracking algorithm based upon modeling the foreground and background regions with a mixture of Gaussians. The GMMs are used to compute a strength image indicating the probability of any given pixel belonging to the foreground. This strength image is embedded into a level set tracking framework in which the target location is estimated by updating a level set function. Extensive experimental results show that the resulting algorithm is able to compute accurate boundaries of multi-colored objects undergoing drastic shape changes, unpredictable motions, and complete occlusion on complex backgrounds. Future work will involve utilizing the extracted shapes to learn more robust priors, and automating the initialization.

## 5. REFERENCES

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 1, 2, 4, 5

[2] S. Avidan. Ensemble tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1, 2

[3] S. T. Birchfield and S. J. Pundlik. Joint tracking of features and edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008. 1, 3, 4

[4] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In *Proceedings of the European Conference on Computer Vision*, pages 471–483, May 2006. 1

[5] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001. 1, 3

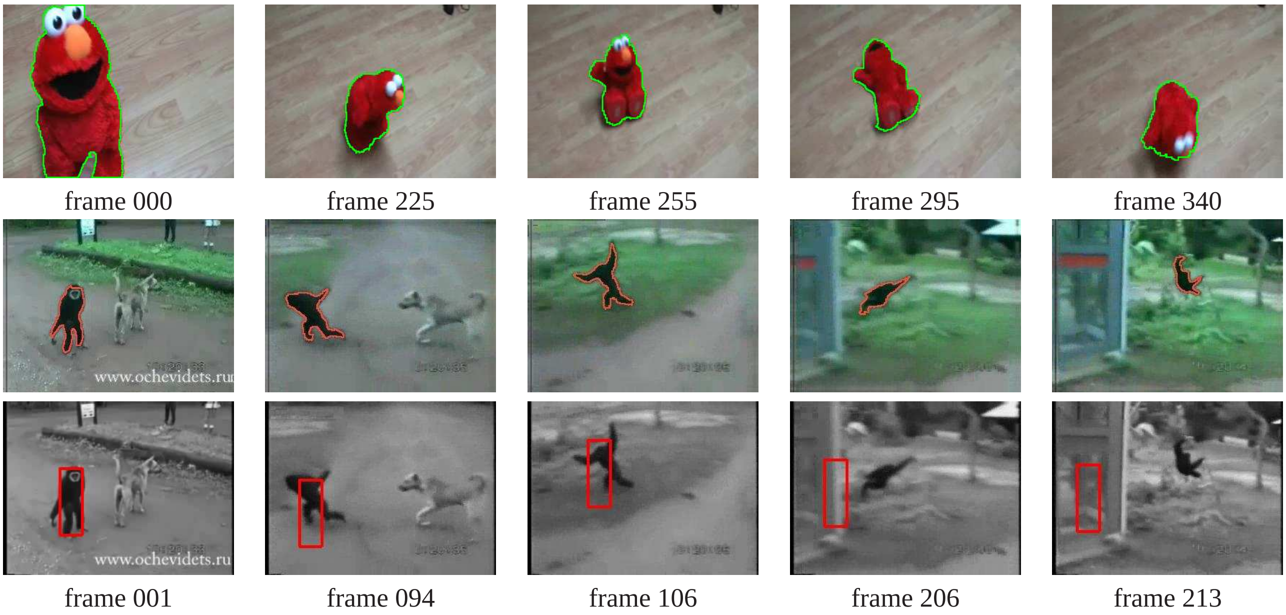
[6] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631 – 1643, Oct. 2005. 1, 2

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. 2, 3

[8] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 2, 3

[9] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, Nov. 1988. 1

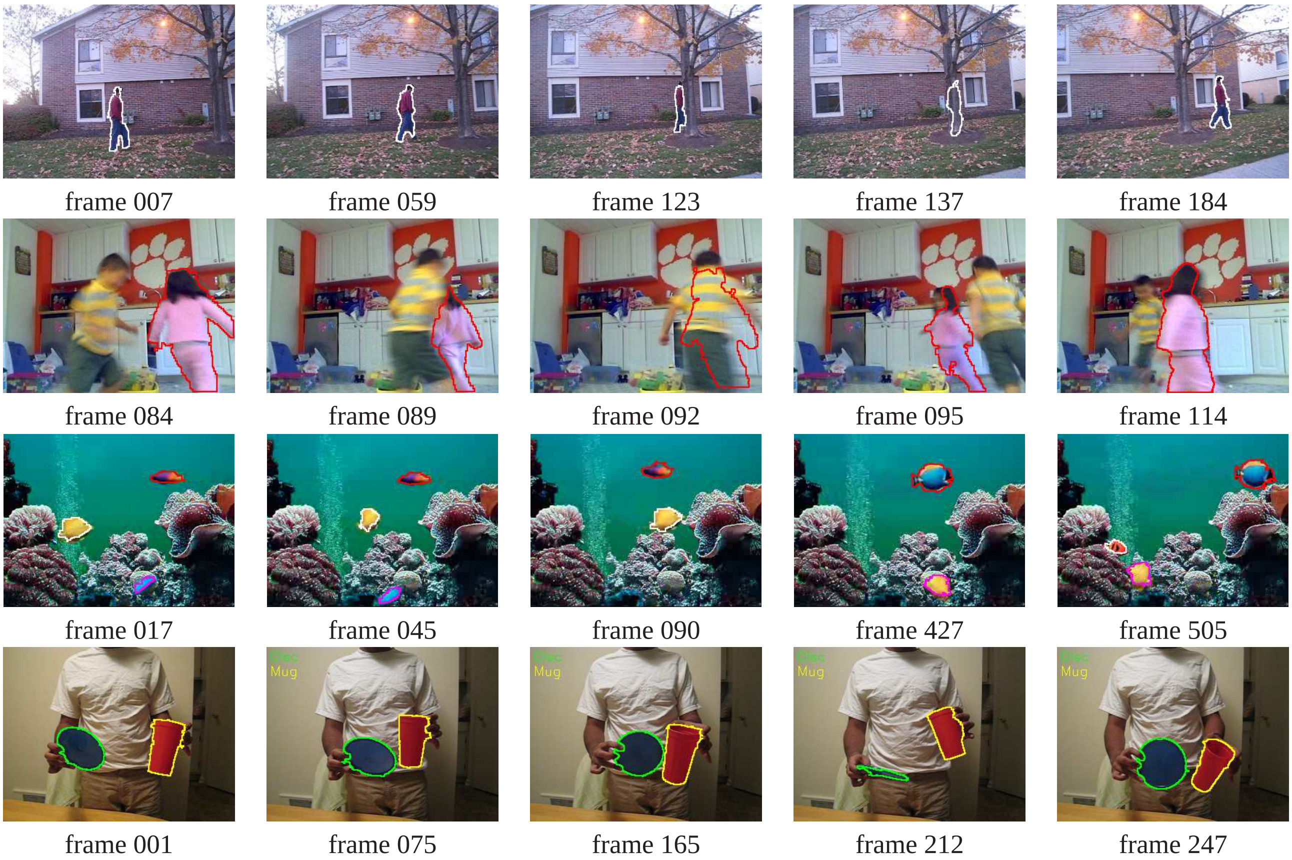
[10] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 728–735, June 2006. 2



frame 000 frame 225 frame 255 frame 295 frame 340

frame 001 frame 094 frame 106 frame 206 frame 213

**Fig. 3**. Results of our algorithm on Elmo sequence (top) and Monkey sequence (middle). Results of FragTrack [1] (bottom) on the Monkey sequence.



frame 007 frame 059 frame 123 frame 137 frame 184

frame 084 frame 089 frame 092 frame 095 frame 114

frame 017 frame 045 frame 090 frame 427 frame 505

frame 001 frame 075 frame 165 frame 212 frame 247

**Fig. 4**. Results of our algorithm on two sequences in which the target is occluded, showing the hallucinated contour in frames 137 (top) and 092 (second row). The third row shows a sequence in which multiple fish swim in a tank and are all tracked by the algorithm. Note especially the camouflaged small blue fish (magenta outline) at the bottom of frames 017 and 045. The bottom row shows the objects classified based on shape information obtained from the contour.