Semi-Automatic Object Tracking in Video Sequences

Federico Lecumberry[*] and Alvaro Pardo[†]

III Workshop de Computación Gráfica, Imágenes y Visualización
Keywords: Procesamiento de Señales, Segmentación de video

Abstract

In this paper we present a method for semi-automatic object tracking in video sequences using multiple features and a method for probabilistic relaxation to improve the tracking results producing smooth and accurate tracked borders. Starting from a given initial position of the object in the first frame the proposed method automatically tracks the object in the sequence modeling the a posteriori probabilities of a set of features such as: color, position and motion, depth, etc.

1 Introduction

In the past, several authors proposed to solve the problem of object tracking and segmentation using color, texture or motion information alone [9]. It can be shown that no single visual feature can be enough to successfully solve the problem in the wide variety of real world scenes. For example, the color of the object can be similar to part of the background and spatial information is needed to overcome this ambiguity. In addition, it is well known that not all features perform uniformly in every situation. For instance, motion and texture features are computed using neighborhood operations that tend to be unreliable at object boundaries. On the other hand, motion alone can discriminate between regions of similar color undergoing different motion.

At the light of these observations, the combination of features emerged as a promising framework. For example, we cite the combination of: color and spatial information [3, 5, 13], color, spatial and motion information [8, 2], color and depth [6], etc. The use of multiple features not only gives us more information but more specifically, it provides us with complementary information. Khan and Shah [8] proposed to join optical flow, color and spatial information into a single feature vector and then build a Gaussian Mixture Model (GMM). To classify the pixels they compute a weighted sum of individual feature likelihoods. The problem with this approach is that motion is very noisy and plays a minor role in the final segmentation while adding noisy estimations. In the present work we separate position, color and motion information. We apply parametric models for motion estimation, which are then used to update the object position in the image.

Traditionally, probabilistic modeling has been used in order to transform the segmentation problem into a classification one. The basic idea is to model the *pdf* of each feature and

[*] IIE-FING-UDELAR fefo@fing.edu.uy
[†] IIE-FING-UDELAR y DIE-UCU apardo@fing.edu.uy

then apply the maximum a posteriori (MAP) principle to classify individual pixels to the corresponding class. Let us suppose that we have two possible classes from the set $\{O, B\}$ where O stands for the object of interest and B for the background, and a set of features $\{f_1, ..., f_N\}$. Then, if we assume independence, the a posteriori probabilities $P(\chi|f_1, ..., f_N)$ with $\chi \in \{O, B\}$ can be written as:

$$\prod_{i=1}^{N} P(\chi|f_i). \tag{1}$$

That means that independent information is combined via the product of individual a posteriori probabilities for each feature. In other works [8, 11, 7], instead of applying (1) the combination is done using a weighted sum of the a posterior probabilities $P(\chi|f_i)$:

$$\sum_{i=1}^{N} \omega_i P(\chi|f_i). \tag{2}$$

This also allows us to introduce the confidence of each measure in the weighting factors ω_i. In [12] the authors studied the problem of classifier combination by averaging and multiplying. They concluded that the averaging classifier is to be preferred in the case when posterior probabilities contain errors. On the other hand, the product rule outperforms averaging when the posterior probabilities are accurately estimated. The underlying idea is that averaging reduces the estimation errors. In addition, we must consider the statistical dependence of the features. If the features are independent and computed without errors, the product rule should be used to take advantage of the independent representations. In the case of noisy estimations, the average rule should be preferred. Since the estimations from different features contain errors: mismatch between the observed color and the modeled ones, errors in the motion estimation, etc., we will use the average rule in this work.

The outline of the present paper is as follows. In section 2 we present a method for probability relaxation that will be used to smooth the posterior probabilities, in section 3 we describe our method and the features used in our tests. In section 4 we present the results and finally in section 5 we present the conclusions and future work.

2 Modified Vector Probability Diffusion

In [10] a method for the diffusion of probability vectors was introduced. Given a vector of probabilities $p(x) \in \mathcal{P} = \{p \in \mathbb{R}^m : ||p||_1 = 1, p_i \geq 0\}$, the anisotropic diffusion that minimizes the L_1 norm, $\int \sqrt{\sum_{i=1}^{m} ||\nabla p_i||^2}$, of this vector restricted to the probability simplex $\mathcal{P}$ is:

$$\frac{\partial p_i}{\partial t} = \nabla . \left(\frac{\nabla p_i}{\sqrt{\sum_{i=1}^{m} ||\nabla p_i||^2}} \right) \quad i = 1, ..., m \tag{3}$$

This evolution equation slows the diffusion at points with high $||\nabla p|| = \sqrt{\sum_{i=1}^{m} ||\nabla p_i||^2}$. Here we present a modification of (3) in order to stop the diffusion along a desired direction z while allowing the diffusion in the normal direction. Our main goal is to stop the diffusion across the object borders while allowing the diffusion along them.

Lets assume we have a vector field z with unit length and direction normal the object borders. For each component p_i of the probability vector, the direction of diffusion in (3) is

∇p_i[1]. Since we intend to stop the diffusion across the object border we modify the direction of diffusion subtracting the component across the border, i.e. the component parallel to z. Hence, we define the new diffusion direction as: $\nabla p_i - < z, \nabla p_i > z$. To correct the strength of the diffusion we must also modify the norm of the gradient to suppress the contribution of the derivative parallel to z. Taking into account the previous modifications, the corresponding Modified Vector Probability Diffusion (MVPD) equation becomes:

$$\frac{\partial p_i}{\partial t} = \nabla . \left(\frac{\nabla p_i - < \nabla p_i, z > z}{\sqrt{\sum_{i=1}^{m} \| \nabla p_i - < z, \nabla p_i > z \|^2}} \right) \quad i = 1, ..., m. \tag{4}$$

To select the object borders we set z as: $z = \left(\frac{u_x}{\sqrt{b^2 + u_x^2 + u_y^2}}, \frac{u_y}{\sqrt{b^2 + u_x^2 + u_y^2}} \right)$ where u is the luminance component of the image and b is a parameter that selects the relevant borders as points with $||\nabla u(x, y)|| \gg b$. It can be easily shown that the equation (4) comes from the minimization of the functional:

$$\int \| \nabla p_i - < z, \nabla p_i > z \| dx.$$

Furthermore, it can be shown that the evolution guarantees that the probability lives in the manifold of vectors with components adding up to one. In future work we will address the study of this problem using ideas such as the ones used in [10] to verify if the continuous equation and/or its discrete version fulfills a maximum principle and in this way if the diffusion remains in the probability simplex. In this work, we use an additional projection step of the diffused vectors to the probability simplex.

The numerical implementation of (4) can be done using standard numerical methods taking forward differences for the gradients and backward differences for the divergence operator. To select the stopping time for the evolution we ask the L_1 norm of the result to be a fraction of the L_1 norm of the initial condition. In this way, we do not need to select the stopping time.

3 The method

In this section we present the features used in this work: color, position (updated via motion estimation), and depth, and the posterior probability estimation and combination.

Color Color is an interesting feature for deformable object tracking since it is robust against different types of deformations. We represent color in the L*a*b* color space and model object and background color pdfs with a GMM:

$$p(f_c|\chi) = \sum_{i=1}^{n_c} \alpha_i^c \mathcal{N}_i(\mu_i^c, \Sigma_i^c)$$

where $\mathcal{N}(\mu_i, \Sigma_i)$ is a gaussian kernel with mean μ_i and covariance Σ_i.

Despite its robustness, color information does not contain any spatial information considering the object shape. Some authors proposed to include the (x, y) pixel positions in the feature vector [3, 5]. Including the pixel position into the feature vector produces compact clusters. However, when using GMM these clusters are elliptical and restrict the type of objects that can be modeled. Therefore, as in [13], we compute the spatial/position distribution separately using kernel methods. To obtain an estimation of the object shape we use motion estimation.

[1] This comes from the fact that the heat equation $\frac{\partial f}{\partial t} = \nabla.(k \nabla f)$ diffuses the heat f in the direction of ∇f with a conduction coefficient k

Motion Motion information is taken into account to estimate the object shape at frame $t+1$, $\hat{S}(t+1)$, given the shape at frame t, $S(t)$. To track the object shape deformation we apply an optical flow technique to obtain the affine motion of the object [14]. The optical flow $(v_x(x,y), v_y(x,y))$ is obtained as the linear least square solution of:

$$\sum_{(x,y)\in S(t)} \left[u(x-v_x(x,y), y-v_y(x,y); t+\Delta t) - u(x,y;t)\right]^2. \tag{5}$$

Taking a first order approximation of (5) and using:

$$(v_x, v_y) = (a_1 + a_2 x + a_3 y, b_1 + b_2 x + b_3 y)$$

we obtain the following system of linear equations:

$$\sum \begin{bmatrix} u_x^2 & u_x^2 x & u_x^2 y & u_x u_y & u_x u_y x & u_x u_y y \\ u_x^2 x & u_x^2 x^2 & u_x^2 xy & u_x u_y x & u_x u_y x^2 & u_x u_y xy \\ u_x^2 y & u_x^2 xy & u_x^2 y^2 & u_x u_y y & u_x u_y xy & u_x u_y y^2 \\ u_x u_y & u_x u_y x & u_x u_y y & u_y^2 & u_y^2 x & u_y^2 y \\ u_x u_y x & u_x u_y x^2 & u_x u_y xy & u_y^2 x & u_y^2 x^2 & u_y^2 xy \\ u_x u_y y & u_x u_y xy & u_x u_y y^2 & u_y^2 y & u_y^2 xy & u_y^2 y^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = -\sum \begin{bmatrix} u_x u_t \\ u_x u_t x \\ u_x u_t y \\ u_y u_t \\ u_y u_t x \\ u_y u_t y \end{bmatrix}$$

To add robustness to the estimations, in the previous summations we only consider points in $S(t)$ where the gradient can be computed with confidence. In our implementation we only consider points (x,y) such that $||\nabla u(x,y)|| > 16$.

Position The object position probability, $P(O|f_s)$, is computed convolving the position support of the object, $S(t)$, with a Gaussian kernel, and the background probability is then $P(B|f_s) = 1 - P(O|f_s)$.

Depth In the case of the flowers sequence (in figure 4) we also include the depth feature via the disparity. The disparity for the i-th frame is calculated using the algorithm proposed by Bobick and Intille [1]; considering the i-th and $(i+1)$-th frames as the left and right images of a stereo rig. Even though this sequence was not prepared for stereo processing the apparent object motion (front-parallel to the camera plane)[2] and the relative depth difference of the objects of interest, allows us to use this algorithm to estimate the disparity; which is verified with the experimental disparity estimation (see figure 4).

To estimate a probability function for each pixel given its disparity value we use a histogram model for the background and object, updating this model in each frame.

Posterior probabilities combination Once we have the posteriori probabilities we estimate the posterior probabilities as[3]:

$$\hat{P}(\chi) = \sum_{i=1}^{N} \omega_i P(\chi|f_i). \tag{6}$$

Finally, we introduce contextual information applying MVPD to the vector of posterior probabilities $(\hat{P}(O), \hat{P}(B))$, before pixel-wise MAP classification. This helps to overcome errors during the posterior probability estimation and adds coherence to the results.

[2] Which guarantee that same rows of consecutive frames are “closely” in epipolar correspondence.

[3] In our implementation we select uniform weights $\omega_i = 1/N$.

Algorithm The only input of the algorithm is the initial object position, $S(0)$. From it, we compute the posterior probabilities for each feature. In the case of color, the initial condition of the EM algorithm is obtained with the fuzzy C-means algorithm. The optimal number of components in the GMM is automatically selected using the modified EM method proposed in [4]. Then, given $S(t)$, we proceed as follows:

(1) Estimate the new object shape, $\hat{S}(t+1)$, using motion information.
(2) Compute the posterior probabilities for the selected features (shape, color, disparity, etc.).
(3) Obtain the posterior probabilities with equation (6).
(4) Diffuse the posterior probabilities with equation (4).
(5) Obtain the new object position using MAP rule.

4 Results

In the first experiment we compare the results of the proposed algorithm using MVPD against the same algorithm using VPD and the algorithm without probabilistic relaxation (WOP). To assess the quality of the results we computed the number of false positives (FP) and false negatives (FN) (figure 1-a left and right respectively) with respect to a sequence segmented by hand by an experienced user.

In figures 1-a and 2 we summarize the results for six representative frames. Note how the results with diffusion reduce the number of FN due to the regularization at the borders and a small increase of the FP. In addition, we can see that the results with MVPD reduce even more the number of FN. That means that with respect to the results expected by the expert we improve via the use of probability diffusion, particularly MVPD.

In figure 1-b we show the localization of FP (in black) and FN (in white) for MVPD and VPD. To understand the increase of FP we must consider the following facts. First, the user consistently omitted the hair close to the neck while the methods with diffusion (MVPD and VPD) included it; these points constitute the FP close to the neck. This explains the almost constant differences between the results with VPD or MVPD, and WOP. Second, the hand-segmented results have rugged borders and the results with diffusion smooth ones.

The FN, mainly concentrated at the helmet, are points with colors similar to the ones of the background, and also weak borders (gradients). That means that the hand segmented image has a global subjective decision that is not taken into account explicitly in our algorithm. Even though these characteristics of the video sequence, the algorithm successfully tracks the helmet border across several frames.

In figure 2 we can see that VPD and MVPD improve the smoothness of the object borders whit respect to WOP. MVPD smoothes even more while respecting the borders. Since these improvements can only be seen at the borders, the improvements are small in terms of FN. To remark the differences between VPD and MVPD in figure 2-b we show the diffused $\hat{P}(O)$ for each method.

To conclude, qualitative we can see that the results after MVPD are smooth and decrease the number of FN at the cost of a slightly increasing the FP while obtaining a stable segmentation across several frames, up to frame 200.

In the second example at figure 3 we present the results for the segmentation of sequence carphone. As we can see the results are stable and precise across several frames up to frame 300. In this case the method successfully tracks the object capturing the motion, deformation and zooming. In frame 178 another object (the hand) with similar features occludes the main object (head) and for this reason the algorithm included as part of the object being segmented. These kinds of problems were not modeled in the proposed method. Hence the results were as expected. Later, when the hand disappears form the scene the method segments only the head.

In the third example at figure 4 we present the results of the disparity estimation and the segmentation results with the combination of color, position and depth (disparity). In the second row we present the results using the method described above. As we can see in the results for the frames 6 and 21 part of the background is included as part of the object. This is due to the closeness of the color and position features of background and object. For the same reason in frame 38 the object includes the branches. In this case the result is correct. In the third column we present the results using only depth and color features[4]. Here the depth corrects the problems commented before but the algorithm used for the depth estimation introduces other errors that deteriorate the segmentation, see the left of the tree at frames 21 and 38. Finally, in the last row, we solve all the previously problems integrating in the proposed method color, position and depth features. Although some problems remain present, we presented the improvements achieved due the combination of several features.

Finally, in the fourth example in figure 5 we present the results for the Train sequence. In this case, we show some limitations of the proposed algorithm. Note how the object region catches some of the background. This is due to the similarity of object and background colors that are uncovered during the object motion. In figure 6 we show an example changing the weights of color and position features. In the first row, the color and position have the same weight and the chimney gets lost during the first frames because the color model cannot discriminate between object and background. In the second row we show how this can be solved using different weights assigning more wigth to the position feature.

5 Conclusions

In this work we presented an algorithm for semi-automatic object tracking in video sequences using several features and a new probabilistic relaxation method (a modified version of an existing VPD). Although the results presented can be improved with more sophisticated methods, we showed that a simple method together with MVPD is able to track objects in long sequences producing smooth and accurate borders.

We want to stress the fact that even though the proposed method is a region-based one and no constraints are imposed to the boundary of the tracked regions, the borders are smooth. The accuracy of the borders of the tracked objects depend on the power of discrimination of the selected features, and the appearance of new objects and/or background. Our algorithm does not consider the latest case. In order to overcome some of these limitations we plan to use snakes or other methods to further improve the results.

The tracked objects in the examples are almost rigid objects. For the case of non-rigid objects the motion estimation must be improved to track the deformations of the object. The solution could be to apply the same motion estimation on a region basis. For example, dividing the object in small regions and then applying a grouping principle.

[4] Remember that we always use the motion feature for the update of position.

References

[1] Aaron F. Bobick and Stephen S. Intille. Large Occlusion Stereo. *International Journal of Computer Vision*, 33(3):181–200, September 1999.

[2] Roberto Castagno, Touradj Ebrahimi, and Murat Kunt. Video Segmentation Based on Multiple Features for Interactive Multimedia Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):562–571, September 1998.

[3] Mark Everingham and Barry Thomas. Supervised Segmentation adn Tracking of Nonrigid Objects using a Mixture of Histograms Model. In *ICIP01 - International Conference on Image Processing*, pages 62–65, 2001.

[4] Mario Figueiredo and Anil Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Transaction on Pattern and Machine Intelligence*, 24(3):381–396, March 2002.

[5] Hayit Greenspan, Jacob Goldberger, and Arnaldo Meyer. Probabilistic Space-Time Video Modeling via Piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, March 2004.

[6] Michael Harville, Gaile Gordon, and John Woodfill. Foreground segmentation using adaptive mixture models in color and depth. In *IEEE Workshop on detection and recognition of events in video*, pages 3–11, 2001.

[7] Eric Hayman and Jan-Olof Eklundh. Probabilistic and Voting Approaches to Cue Integration for Figure-Ground Segmentation. In *ECCV 2002*, LNCS 2352, pages 469–486.

[8] Sohaib Khan and Mubarak Shah. Object Based Segmentation of Video using Color, Motion and Saptial Information. In *CVPR2001 - Int. Conf. Computer Vision and Pattern Recogbition*, volume 2, pages 746–751, 2001.

[9] N.Friedman and S.Rusell. Image segmentation in video sequence: A probabilistic approach. In *Conference Uncertainty in Artificial Intelligence*, number 13, 1997.

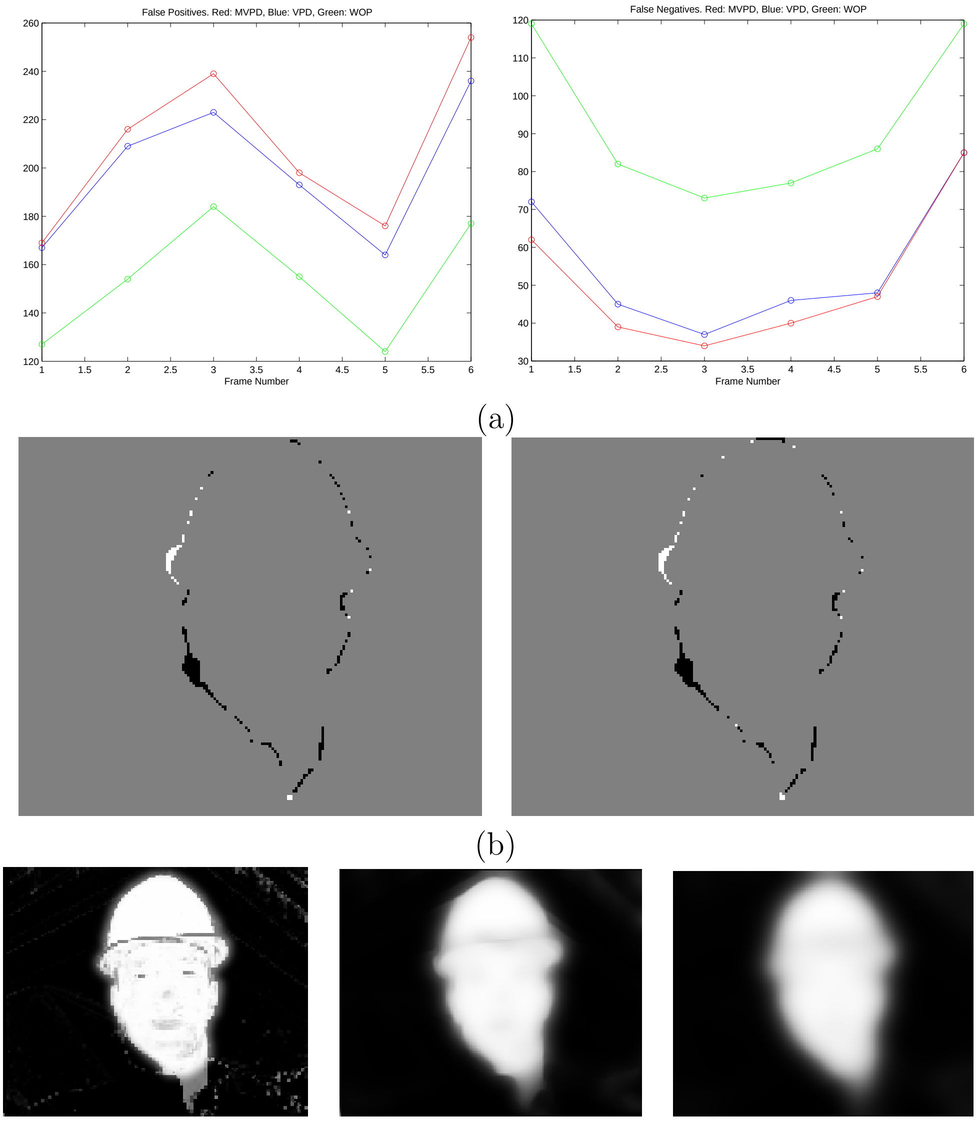
[10] Alvaro Pardo and Guillermo Sapiro. Vector Probability Diffusion. *IEEE Signal Processing Letters*, 8(4):106–109, April 2001.

[11] Martin Spengler and Bernt Schiele. Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, 14:50–58, 2003.

[12] David Tax, Martijn van Breukelen, Robert Duin, and Josef Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33:1475–1485, 2000.

[13] D. Thirde, G. Jones, and J. Flack. Spatio-Temporal Semantic Object Segmentation using Probabilistic Sub-Object Regions. In *BMVC2003 - British Machine Vision Conf.*, 2003.

[14] John Wang and Edward Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.



(a)

(b)

(c)

Figure 1: (a) False Positives and False Negatives for frames 5, 10, 15, 50, 100 and 200. (b) False Positives (in black) and False Negatives (in white) for MVPD and VPD (left and right respectively). (c) Results of diffusion with MVPD and VPD for the object probability $\hat{P}(O)$.

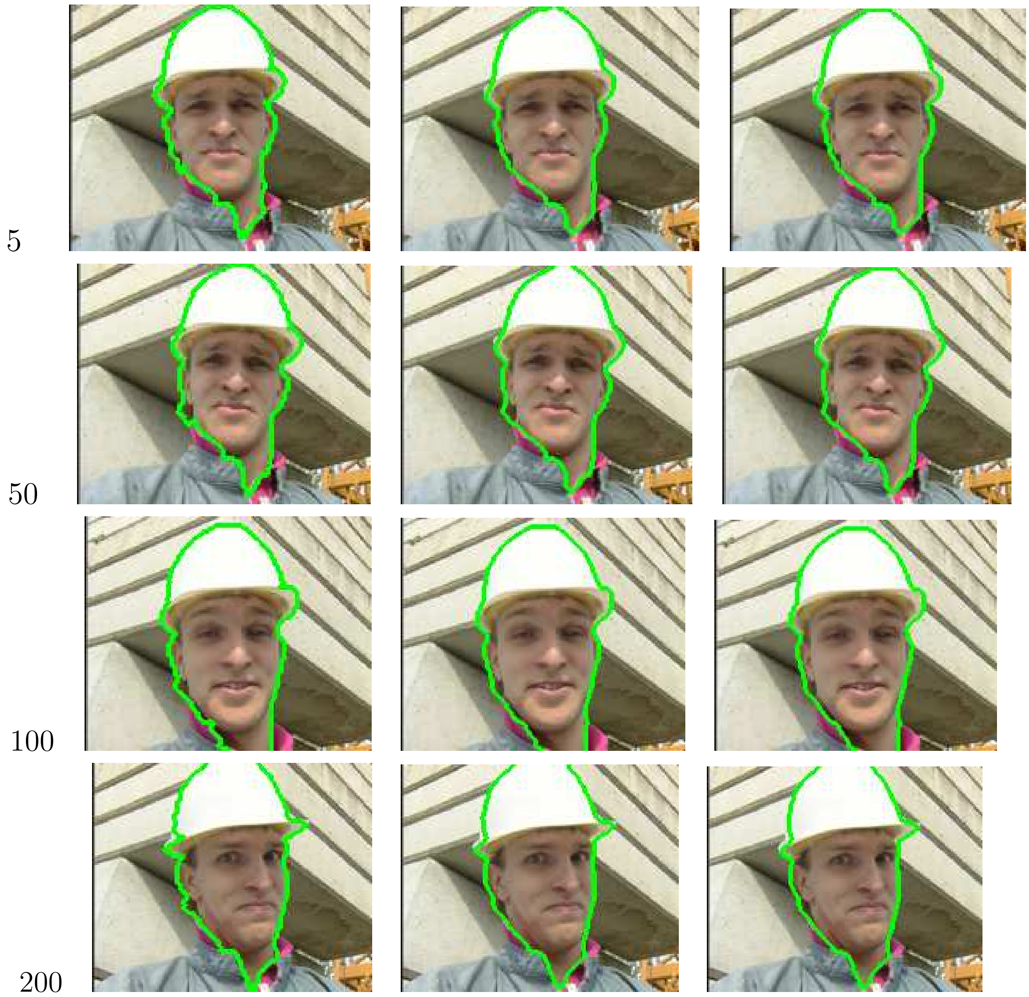


Figure 2: Left: Results of WOP. Middle: Results with VPD. Right: Results with MVPD.

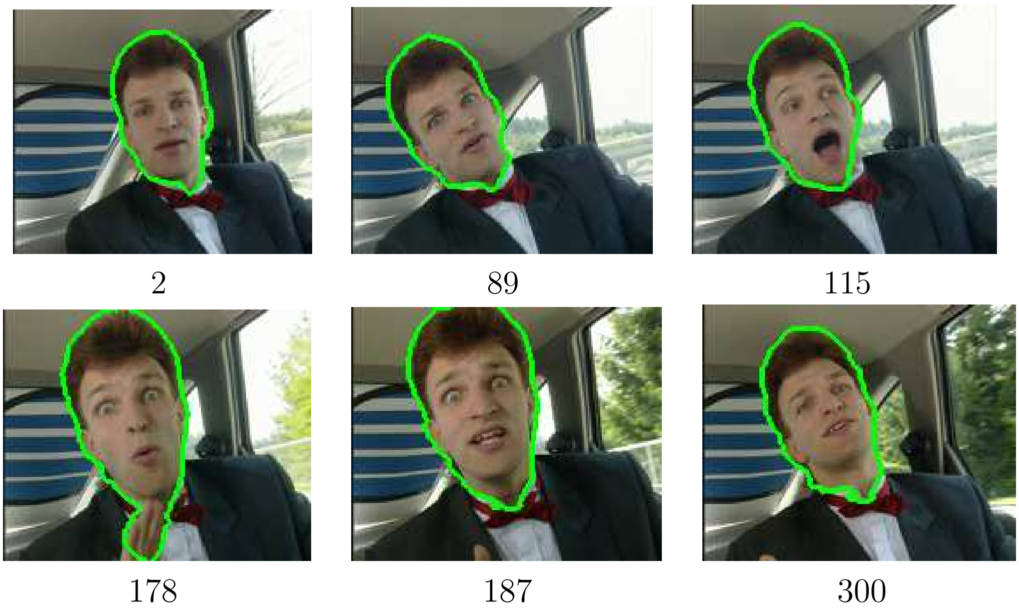


Figure 3: Results for carphone sequence with color, position and MVPD.

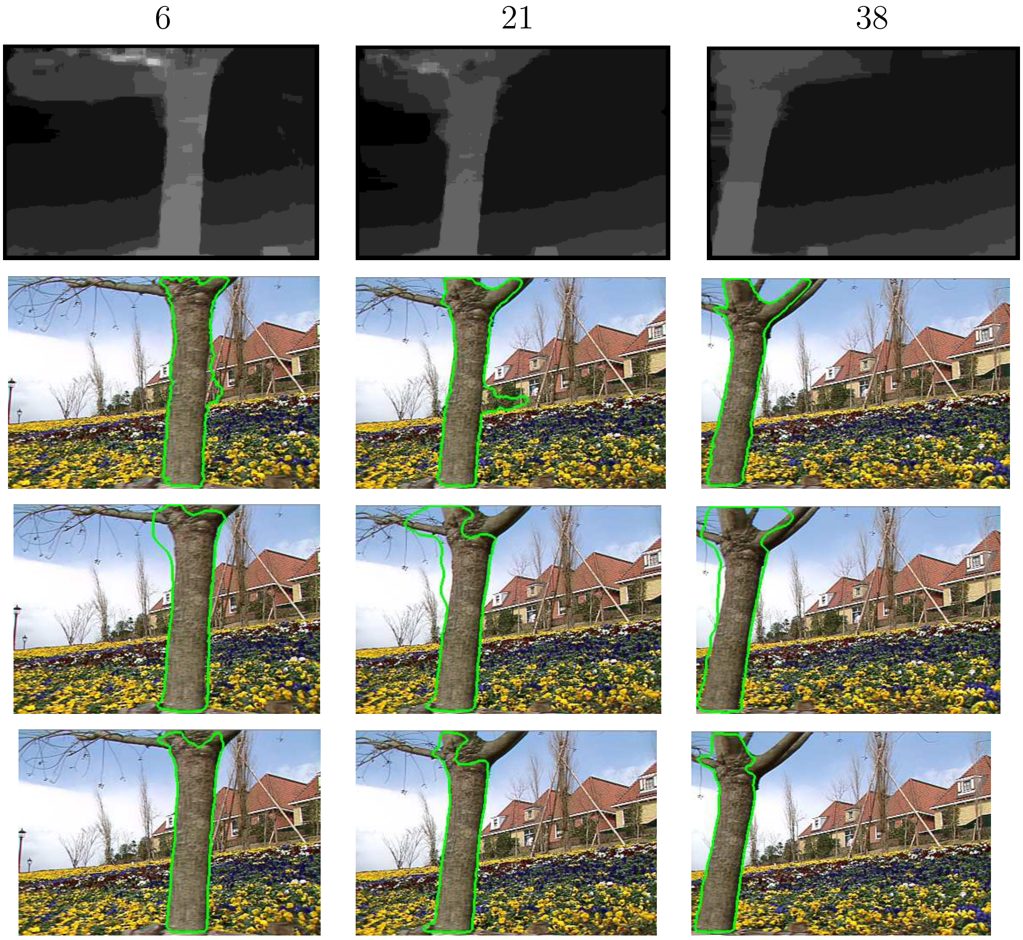


Figure 4: From top to bottom: Disparity estimation. Results using color and position with MVPD. Results using position and depth with MVPD. Results using position, depth and color with MVPD.

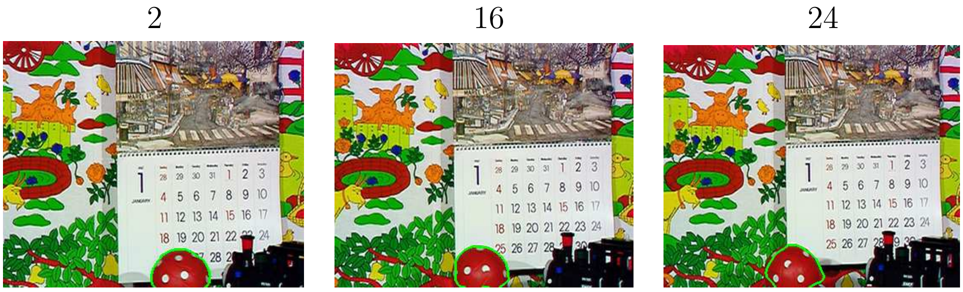


Figure 5: Note how the object region catches some of the background in frame 24, due to the similarity of object and background colors that are uncovered during the object motion.

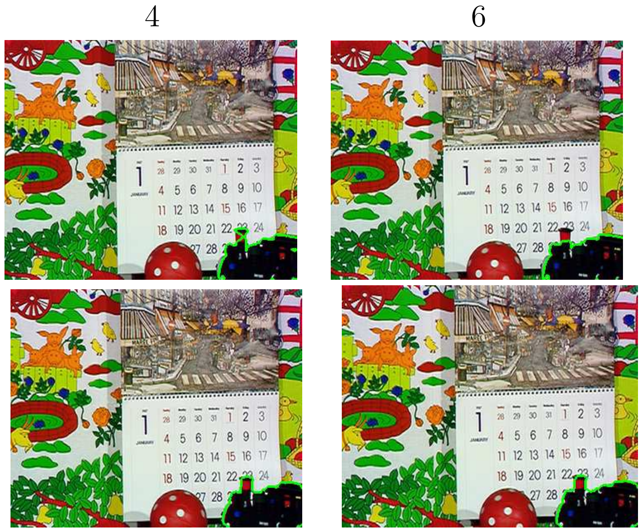


Figure 6: Note how the object region catches some of the background in frame 24, due to the similarity of object and background colors that are uncovered during the object motion.