Segmentation for Robust Tracking in the Presence of Severe Occlusion

Camillo Gentile, Octavia Camps, and Mario Sznaier

***Abstract*—Tracking an object in a sequence of images can fail due to partial occlusion or clutter. Robustness to occlusion can be increased by tracking the object as a set of "parts" such that not all of these are occluded at the same time. However, successful implementation of this idea hinges upon finding a suitable set of parts. In this paper we propose a novel segmentation, specifically designed to improve robustness against occlusion in the context of tracking. The main result shows that tracking the parts resulting from this segmentation outperforms both tracking parts obtained through traditional segmentations, and tracking the entire target. Additional results include a statistical analysis of the correlation between features of a part and tracking error, and identifying a cost function that exhibits a high degree of correlation with the tracking error.**

***Index Terms*—Active contours, robust tracking, segmentation.**

I. Introduction

Tracking a known object in a sequence of frames can fail due to occlusion or the presence of clutter. Robustness against these effects can be increased by using robust estimators [1], [8], [18], [23] that treat occlusion pixels as outliers. However, while successful for moderate occlusion, these estimators usually break down at above a 30% occlusion level [2]. This is illustrated in Fig. 1,[1] where an affine transformation combined with a robust estimator was used to track a bus in a traffic sequence. As shown there, the algorithm begins to lose track of the target in Frame 14.

This effect can be traced to the fact that robust estimators treat occluding pixels as uniformly distributed outliers, neglecting the fact that occlusion tends to be clustered in small regions. Thus, intuitively one would expect that resiliency to occlusion could be improved by dividing the object into pieces which are tracked separately, along with the entire object, to find multiple transformations. The best global transformation is then selected by voting [6]. However, homogeneous pieces are more difficult to track than regions with distinctive properties such as texture or shape.[2] Thus, standard segmentations (see for instance [12], [14], [15], [24], [25], and references therein) do not necessarily result in parts leading to good tracking performance. This is illustrated in Fig. 2 where the use of the set of homogeneous parts (an MDL-based segmentation [12] of the bus) shown in Fig. 2(a) leads to poor tracking starting in Frame 2. Motivated by this difficulty, in this paper we address the problem of how to divide the object into pieces to optimize tracking robustness to occlusion. Specifically, the contributions of the paper are

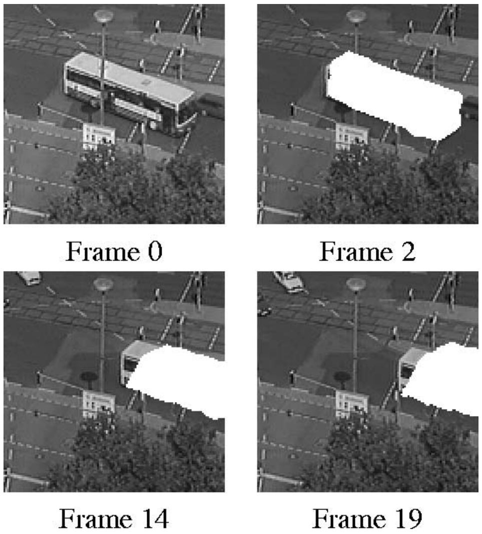

Fig. 1. Tracking using an affine transformation combined with a robust estimator. The algorithm begins to lose track of the target in Frame 14 as it moves outside the field of view.

- statistical analysis of the correlation between features of a part and tracking error;
- identifying a cost function that exhibits a higher degree of correlation with the tracking error than other indicators previously proposed;
- a segmentation algorithm specifically designed to make optimal use of the spatial information available to improve tracking robustness. This segmentation is obtained by combining this new cost function with the standard active contours ("snakes") framework [21], [25].

The paper is organized as follows. In Section II, we introduce the notation, briefly review the tracking problem and formally state the segmentation problem of interest. In Section III, we introduce a cost function that is highly correlated with the tracking error and we benchmark it against previously proposed cost functions. Section IV describes a snake architecture employed to obtain a partitioning of the object that optimizes this cost function. In Section V we report tracking results on synthetic and real images and we compare the performance of the

Manuscript received June 12, 2001; revised March 12, 2003. This work was supported in part by the NSF under Grants IRI-9712598 and IIS-0117387. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thiow Keng Tan.

The authors are with the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: camillo@gandalf.ee.psu.edu; camps@whale.ee.psu.edu; msznaier@gandalf.ee.psu.edu).

Digital Object Identifier 10.1109/TIP.2003.817232

[1] This sequence of traffic images was provided by Dr. Nagel at the Universitat Karlsruhe.

[2] This is closely related to the well known aperture problem.

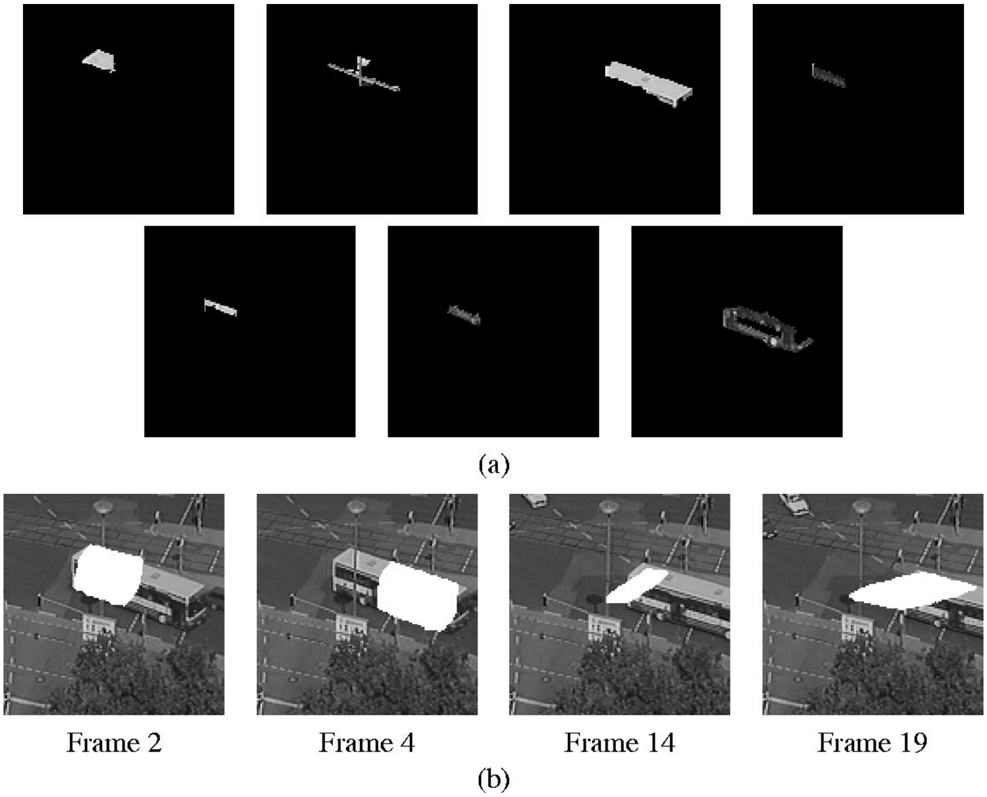

Fig. 2. Tracking using homogeneous parts. (a) Regions of the target obtained using an MDL-based segmentation algorithm. (b) The regions shown in (a) are tracked in a effort to improve resiliency to occlusion. However, regions lacking distinctive properties such as texture or shape lead to poor tracking performance.

proposed method vis-a-vis other commonly used segmentations both with synthetic and real images. Finally, in Section VI we summarize our results.

II. Preliminaries

A. Notation and Mathematical Preliminaries

Definition 1: An affine transformation $\mathcal{A}_\alpha$ is a transformation of the form $\mathcal{A}_\alpha(\mathbf{x}) = \mathbf{x}' = A\mathbf{x} + b$, where $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in R^{2\times 2}$ and $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in R^2$.

In the sequel we will denote by $\mathbf{x} = (x \quad y)^T$ the coordinates of a pixel; $I(\mathbf{x})$ the greyscale intensity at pixel $\mathbf{x}$; $I_x(\mathbf{x})$ and $I_y(\mathbf{x})$ the corresponding spatial derivatives; and by $\alpha = [a_{11}\, a_{12}\, a_{21}\, a_{22}\, b_1\, b_2] \in R^6$ the parameters of a given transformation.

B. Tracking Problem

For simplicity, in this paper we consider a prototype tracking algorithm based on estimating the transformation that maps the images of a given object between two consecutive frames in a sequence. However, our results can also be used in the context of tracking algorithms that exploit, in addition to spatial, temporal information [3], [16].

Assuming that there is little distortion, the mapping between consecutive frames can be considered to be affine [2], [4], [9]–[11]. This fact can be exploited to efficiently solve the tracking problem by recasting it into the following optimization form [2].

Problem 1 (Robust Affine Tracking): Given two image frames I^{f_1} and I^{f_2}, and a prototype object represented by a subset of pixels $P \subseteq I^{f_1}$, find an affine transformation $\mathcal{A}_\alpha : I^{f_1} \rightarrow I^{f_2}$ that minimizes the following objective function:

$$R(\mathcal{A}_\alpha) = \sum_{\mathbf{x}\in P} \rho\left\{I^{f_2}[\mathcal{A}_\alpha(\mathbf{x})] - I^{f_1}(\mathbf{x}), \theta\right\} \qquad (1)$$

where $\rho(.,\theta)$ is a robust estimator [1], [2] that rejects outliers, controlled by the tuning parameter θ. For example,

$$\rho(r,\theta) = \begin{cases} \frac{1}{2}r^2 & |r| < \theta \\ 0 & \text{otherwise} \end{cases}.$$

In principle Problem 1 can be solved by performing a gradient descent search to find a (local) minimum of (1). However, as illustrated in Fig. 1, while this approach works well for moderate occlusion, the estimator $\rho(r,\theta)$ may not prove reliable in the midst of severe occlusion.

C. Parts Reset Algorithm

Consider the experiment shown in Fig. 3, where severe occlusion is simulated by synthetically cutting off a portion of the object *Van*, pasting it to a cluttered background and adding zero-mean additive white Gaussian noise with variance 5. Fig. 3(c) shows the results of the affine transformation found using gradient descent search combined with a truncated quadratic robust estimator [1] (in the sequel we will refer to this algorithm as the Benchmark Algorithm). In order to minimize

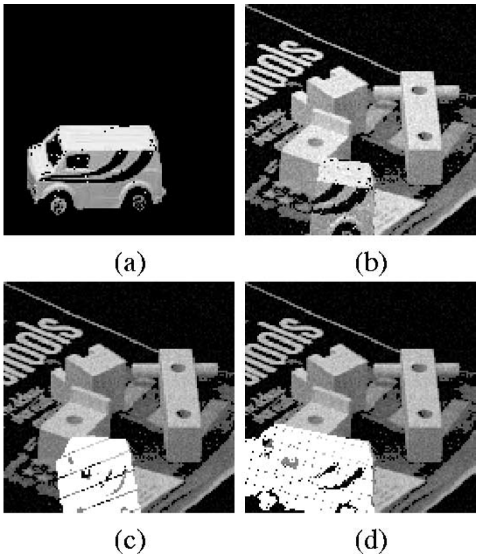

Fig. 3. PRA versus Benchmark Algorithm. (a) The object. (b) A synthetically generated scene with a portion of the object set among clutter. (c) The Benchmark Algorithm fails to identify the occluded pixels of the prototype and minimizes the error norm by compressing the object into the portion visible in the frame. (d) The Parts Reset Algorithm improves robustness to occlusion by tracking a set of parts with good tracking properties.

the error norm, the algorithm compresses the object into the portion visible in the frame, since it cannot successfully determine the occluded pixels of the prototype. Additional sources of error arise from the nonconvexity of the problem, and from the effects of cluttered background and noise on the gradient computation.

Since occlusion is a localized effect, robustness against severe occlusion can be improved by partitioning the prototype into a number N of parts and tracking the parts by estimating N candidate affine transformations $\mathcal{A}_{\alpha_i}, i = 1, \ldots, N$[6]. A single transformation $\mathcal{A}_\alpha$ may be selected from the candidate transformations by a voting scheme. Further improvement can be obtained by evaluating the performance of each transformation at intermediate stages, rather than after the steepest descent procedure has converged. Specifically, at uniform intervals (in number of iterations), all candidate transformations $\mathcal{A}_{\alpha_i}$ are applied to each part $\{P_j\}$, and the corresponding errors $R(\mathcal{A}_{\alpha_i})$ are computed using (1). The transformation associated to each part is then reset to the one yielding the lowest error and the steepest descent search is restarted. Consistent experimental evidence indicates that this approach enables unoccluded parts to set partially occluded parts on track, thus avoiding local minima of (1). This effect is illustrated in Fig. 3(d) showing the tracking results for an algorithm based upon this idea. Note in passing that, since the total number of points involved is the same for the benchmark and PRA algorithms, in both cases the total number of operations required to compute the gradients is the same. Thus, both algorithms have roughly the same computational complexity. The PRA algorithm requires slightly more overhead, due to the intermediate voting, but, on the other hand, can be easily parallelized.

III. Good Features for Tracking

While the approach described in the previous section has the potential to handle substantial occlusion, it hinges upon determining a suitable set of parts to be tracked. Possible options span a very diverse spectrum from dividing the object image by using a simple grid, to segmenting the object into homogeneous regions, to dividing the object into its "functional" parts. While the first option is the simplest partition and the latter options are intuitively appealing, they are not necessarily the best partitions for the application being considered. In this section we analyze the correlation between different features of a part and tracking performance. Based on this analysis, we propose a new segmentation, designed to optimize tracking robustness.

TABLE I
Indicators for Good Tracking Commonly Used

Description	Definition
intensity variance	$var_I = \sum_{\mathbf{x}\in P} (I(\mathbf{x}) - \eta_I)^2$
gradient	$grad = \sum_{\mathbf{x}\in P} I_x^2(\mathbf{x}) + I_y^2(\mathbf{x})$
normalized gradient	$grad_n = \frac{1}{\zeta(P)} \sum_{\mathbf{x}\in P} I_x^2(\mathbf{x}) + I_y^2(\mathbf{x})$
normalized laplacian	$lap_n = \frac{-1}{\zeta(P)} \sum_{\mathbf{x}\in P} (I_{xx}(\mathbf{x})+I_{yy})^2(\mathbf{x})$
max. abs. eigenvalue	$max_W = \max_i \lvert\lambda_i\rvert, \; Wv_i = \lambda_i v_i$
min. abs. eigenvalue	$min_W = \min_i \lvert\lambda_i\rvert, \; Wv_i = \lambda_i v_i$
norm eigenvalues	$norm_W = \sum_i \lambda_i^2, \quad Wv_i = \lambda_i v_i$
ratio eigenvalues	$rat_W = -\frac{max_W}{min_W}$

A. Performance of Several Indicators

Several ways of assessing the "goodness" (in the sense of its ability to minimize the tracking error) of a part have been proposed in the literature, based on spatial derivatives, image Laplacian or the eigenvalues of the matrix [19]

$$W = \sum_{\mathbf{x}\in P} \begin{bmatrix} x^2I_x^2 & x^2I_xI_y & xyI_x^2 & xyI_xI_y & xI_x^2 & xI_xI_y \\ x^2I_xI_y & x^2I_y^2 & xyI_xI_y & xyI_y^2 & xI_xI_y & xI_y^2 \\ xyI_x^2 & xyI_xI_y & y^2I_x^2 & y^2I_xI_y & yI_x^2 & yI_xI_y \\ xyI_xI_y & xyI_y^2 & y^2I_xI_y & y^2I_y^2 & yI_xI_y & yI_y^2 \\ xI_x^2 & xI_xI_y & yI_x^2 & yI_xI_y & I_x^2 & I_xI_y \\ xI_xI_y & xI_y^2 & yI_xI_y & yI_y^2 & I_xI_y & I_y^2 \end{bmatrix} \tag{2}$$

Table I summarizes the most commonly used indicators [7], [17], [19], [20]. Here a higher value of the criterion indicates a part that is thought to be more suitable for tracking.

To establish the performance of these features as indicators of good tracking we conducted a set of experiments to find the correlation between the indicators values and tracking error. To this effect we considered a set of parts of varying size, shape, and texture cut from real images and ran a series of tests on each part. These parts bear interesting features used for comparison, namely large regions with homogeneous texture (such as the faces of a box) as well as regions with contrast texture (such as corners and holes). Each experiment consisted of the following steps.

1) Create a prototype frame I^{f_1} by selecting a part P.

2) Create a second frame I^{f_2} by:
   a) Selecting a random background B with uniform grayscale distribution between 0 and 255.
   b) Transforming the prototype of the part P from the identity pose, $\mathcal{A}_0(\mathbf{x}) = \mathbf{x}$, to a test pose $\mathcal{A}_\alpha(\mathbf{x}) = \mathbf{x}'$ and "paste" it onto background B.
   c) Corrupting the resulting scene with zero-mean additive white Gaussian noise with variance 5.
3) Run the tracking algorithm on the frames I^{f_1} and I^{f_2} to compute an estimated pose, $\mathcal{A}_{\hat{\alpha}}$.
4) Find the corresponding ground-truth *tracking error* defined as:

$$d(\mathcal{A}_\alpha, \mathcal{A}_{\hat{\alpha}}|P, B) = \frac{1}{\zeta(P)} \sum_{\mathbf{x} \in P} ||\mathcal{A}_{\hat{\alpha}}(\mathbf{x}) - \mathcal{A}_\alpha(\mathbf{x})||_2 \quad (3)$$

where $\zeta(P)$ denotes the number of pixels of part P.

A total of 19 200 experiments were performed using 40 parts, P_i, pasted onto 10 random backgrounds B_j of size 128 × 128, and under the 48 different affine transformations:

$$A = \begin{pmatrix} w * \cos\theta & -w * \sin\theta \\ sin\theta & \cos\theta \end{pmatrix}, \; b = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

where $\theta = \pm a$, (x_0, y_0) indicate the eight directions adjacent to a pixel, $x0 = -\tau$, $y0 = -\tau$; $x0 = -\tau$, $y0 = 0$; $x0 = -\tau$, $y0 = \tau$; etc, and the values of a, w and τ are given in Table II. These values were chosen to cover problems ranging from easy to challenging.

The overall performance of a part P is obtained by summing over the 48 poses and ten backgrounds to compute the total error, $D(P)$ associated with it

$$D(P) = \sum_{j=1}^{10} \sum_{k=1}^{48} d(\mathcal{A}_{\alpha_k}, \mathcal{A}_{\hat{\alpha}_k}|P, B_j). \quad (4)$$

A potential problem when using (4) to assess the quality of part P is that a few outliers can significantly bias the cumulative performance of the part. To avoid this situation we proceeded as follows. Through the use of a Kolmogorov-Smirnov test [13] we determined that, with probability ≥ 0.75, the distribution of the experiments yielding lowest values of the error is an F distribution (the ratio of two random variables with χ^2 distribution) with parameters $v_1 = 16$ and $v_2 = 4$ (see Fig. 4). Since for this distribution, $F(d) = 0.95$ for the error value $d = 5.8$, all points above this value were considered outliers and assigned an error value of $d = 5.8$[3]. With this saturation the total error of a part ranges from 0 (perfect matching) to 2784 (poor matching).

The correlation coefficient $\sigma_{\mathbf{DJ}}$ between the 40-dimensional vectors $\mathbf{D}$ of tracking errors and $\mathbf{J}$ of indicator values is given by

$$\sigma_{\mathbf{DJ}} = \frac{\mathcal{E}(\mathbf{DJ}) - \mathcal{E}(\mathbf{D})\,\mathcal{E}(\mathbf{J})}{\sigma_{\mathbf{D}}\sigma_{\mathbf{J}}} \quad (5)$$

TABLE II
TEST POSES

level of difficulty	translation τ (pixels)	rotation a (radians)	warping factor w	number of poses
easy	3	0.00	1	8
moderate	7	0.00	1	8
difficult	3	0.15	1	16
challenging	4	0.18	0.955	16

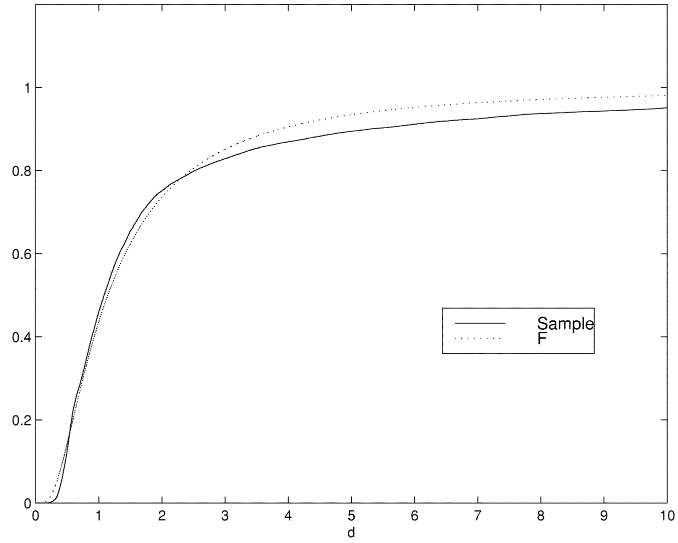

Fig. 4. Kolmogorov-Smirnov test to determine the distribution of the error.

TABLE III
CORRELATION BETWEEN INDICATORS AND THE TRACKING ERROR

indicator	**Correlation (σ_{DJ})**
var_I	-0.3187
$grad$	-0.2864
$grad_n$	-0.0079
lap_n	-0.1823
max_W	-0.3225
min_W	-0.3518
$norm_W$	-0.3049
rat_W	-0.2257
proposed	**-0.7910**

where $\mathcal{E}$ and σ indicate the expected value and standard deviation, respectively.

The correlation coefficients for the indicators defined in Table I are given in the top portion of Table III. Unfortunately, all of them have small absolute value, indicating that the performance of these indicators as predictors of good tracking properties is rather poor.

[3] Values about this threshold correspond to cases where there is a large mismatch between the actual and calculated poses. In this situation, the numerical value of the error is more a function of the background than of the disparity between poses.

B. Performance Oriented Indicator

To find an indicator more correlated with tracking performance we begin by examining the gradient with respect to the affine transformation parameters of (1) used to perform the search for the affine parameters (see (6) at the bottom of the page) where $r(\mathbf{x}) = I^f[\mathcal{A}_\alpha(\mathbf{x})] - I^p(\mathbf{x})$, $P = \bigcup_{i=1}^{N} P_i$ and $P_i \cap P_j = \emptyset$ for $i \neq j$.

Equation (6) shows that, as expected, parts that have large spatial derivatives I_x and I_y as well as large momenta xI_x, yI_y, yI_x, and xI_y result in larger gradient of the objective function in (1) and thus in a faster convergence toward the optimum set of affine parameters. Thus, one would expect that linear combination of these terms, such as

$$e = I_x^2(\mathbf{x}) + I_y^2(\mathbf{x}) + |x| I_x^2(\mathbf{x}) + |y| I_y^2(\mathbf{x}) + |y| I_x^2(\mathbf{x}) + |x| I_y^2(\mathbf{x})$$

would exhibit a large inverse correlation[4] with the tracking error. However, consistent numerical experience indicates that this is not the case. Roughly speaking, performance does not improve once each component of e exceeds a certain threshold. Thus, a better result is achieved by *saturating* the energy of each term, once it exceeds this threshold. As we show in the sequel, this leads to parts with more regular shape, where the individual terms in e tend to have comparable energy. Based on these considerations we propose to use as an indicator of good tracking properties the *energy* of a part P, defined as

$$\begin{aligned} e(P) = & \operatorname{sat}[e_x(P), e_{sat}] + \operatorname{sat}[e_y(P), e_{sat}] \\ & + \operatorname{sat}[e_{xx}(P), e'_{sat}] + \operatorname{sat}[e_{yy}(P), e'_{sat}] \\ & + \operatorname{sat}[e_{yx}(P), e'_{sat}] + \operatorname{sat}[e_{xy}(P), e'_{sat}] \end{aligned} \quad (7)$$

where

$$\begin{aligned} e_u(P) &= \sum_{\mathbf{x} \in P} \operatorname{sat}\left[I_u^2(\mathbf{x}), I_{sat}\right]; \\ e_{uv}(P) &= \sum_{\mathbf{x} \in P} |u - \eta_{uv}| \operatorname{sat}\left[I_v^2(\mathbf{x}), I_{sat}\right] \\ \eta_{uv} &= \frac{1}{e_v(P)} \sum_{\mathbf{x} \in P} u \operatorname{sat}\left[I_v^2(\mathbf{x}), I_{sat}\right]; \\ \operatorname{sat}[c, c_{sat}] &= \begin{cases} \frac{c}{c_{sat}} & \text{if } c \leq c_{sat} \\ 1 & \text{if } c > c_{sat} \end{cases} \end{aligned} \quad (8)$$

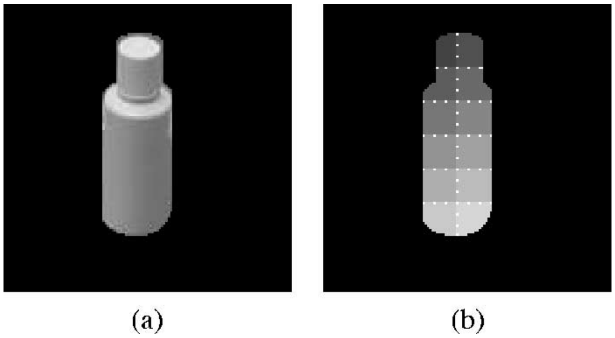

Fig. 5. Initial snake segmentation. (a) Object *Cylinder*. (b) A snake in the form of a square grid is placed on the object as an initial segmentation.

where η_{xx}, η_{yy}, η_{yx} and η_{xy} are used to center the momenta to render the energy coordinate independent.

The parameters I_{sat}, e_{sat}, and e'_{sat} are additional degrees of freedom that can be used to optimize the correlation between the tracking error E and the energy $e(P)$. For the set of 19 200 experiments described at the begining of the section, numerical optimization of σ_{DJ} led to the parameter values: $I_{sat} = 3000$, $e_{sat} = 50$, and $e'_{sat} = 700$ respectively. The corresponding indicator $e(P)$ is highly correlated with the tracking error, with correlation coefficient $\sigma_{De} = -0.7910$[5]. Note that its absolute value is substantially larger than the ones of the other entries in Table III.

IV. Object Segmentation for Tracking

In the last section we proposed to use the *energy* of a part as a predictor of its expected performance in a gradient-based tracking algorithm. The fact that the correlation between energy and expected performance is negative- larger values of the energy lead to smaller values of tracking error-suggests that tracking performance can be improved by finding a partition of the object having parts with large energy values. In this section we describe how to accomplish this by incorporating the energy (7) of a part into a deformable model or snake framework [21].

A. Snake Description

A snake is an ordered set of points $S = [s_1, s_2, \ldots, s_n]$ that can form either open or closed contours. A snake segmentation

$$\begin{aligned} \nabla_\alpha R &= \sum_{\mathbf{x} \in P} \frac{\partial \rho(r(\mathbf{x}), \theta)}{\partial r(\mathbf{x})} \nabla_\alpha r(\mathbf{x}) \\ &= \sum_{\mathbf{x} \in P} \frac{\partial \rho(r(\mathbf{x}), \theta)}{\partial r(\mathbf{x})} \left[I_x^f(\mathbf{x}')\; I_y^f(\mathbf{x}')\; xI_x^f(\mathbf{x}')\; yI_y^f(\mathbf{x}')\; yI_x^f(\mathbf{x}')\; xI_y^f(\mathbf{x}')\right]^T \\ &= \sum_{i=1}^{N} \sum_{\mathbf{x} \in P_i} \frac{\partial \rho(r(\mathbf{x}), \theta)}{\partial r(\mathbf{x})} \left[I_x^f(\mathbf{x}')\; I_y^f(\mathbf{x}')\; xI_x^f(\mathbf{x}')\; yI_y^f(\mathbf{x}')\; yI_x^f(\mathbf{x}')\; xI_y^f(\mathbf{x}')\right]^T \end{aligned} \quad (6)$$

[4] i.e., high values of e correspond to low values of the error.

[5] Computing the median rather than the average over all experiments for a part P in (4) also yields a high coefficient $\sigma_{De} = -0.7954$.

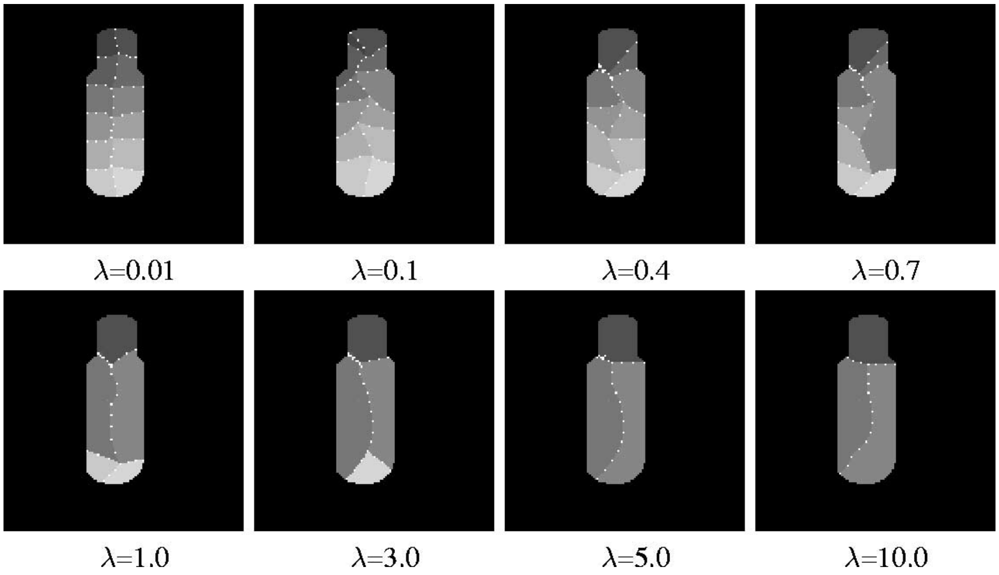

Fig. 6. Segmentation stages of *Cylinder* as λ varies.

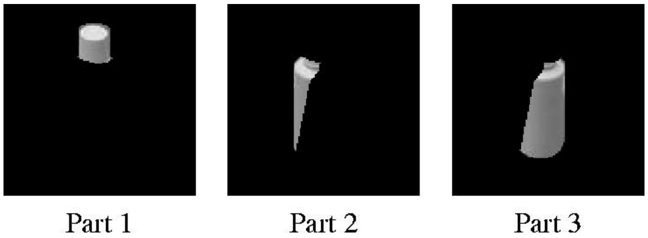

Fig. 7. *Cylinder* segmentation. The final segmentation distributes most of the texture and gradient content of the object, which is concentrated on its upper part, amongst three regions.

algorithm moves the snake on the image grid seeking to minimize an energy function

$$E = \sum_{i=1}^{n} E_{int}(s_i) + E_{ext}(s_i) \tag{9}$$

where the internal force E_{int} imposes continuity and smoothness constraints to avoid oscillations of the contours, and the external force E_{ext} attracts the snake to salient image features.

Let s be a point on the snake, U_s be the subset of points on the snake adjacent to s, and V_s be the set of parts defined by the snake that have s as a contour point. Then, the internal energy at the point s, $E_{int}(s)$, is defined as

$$\begin{aligned} E_{int}(s) = {} & \alpha \max_{t \in U_s} \left\{ (x_s - x_t)^2 + (y_s - y_t)^2 \right\} \\ & + \beta \min_{t,u \in U_s} \left\{ (x_t - 2x_s + x_u)^2 + (y_t - 2y_s + y_u)^2 \right\} \end{aligned} \tag{10}$$

where the first term ensures that points on the snake do not get too far from each other, the second term penalizes high curvature contours, and α and β control the relative influence of the corresponding terms.

As discussed in the previous section, for a tracking application, the external force at the point s, $E_{ext}(s)$, should attract the snake toward enclosing parts with high energy values. Thus, the external force is defined as

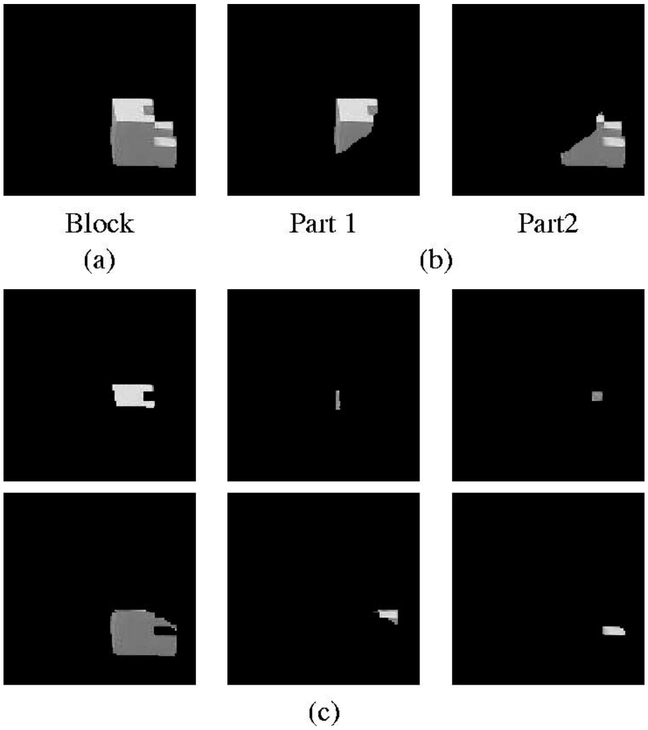

Fig. 8. *Block* Segmentation: (a) Object. (b) The proposed algorithm distributes the high gradient content region of the object between Part 1 and Part 2. The two parts have similar energy values and are localized and compact. (c) MDL-based segmentation.

$$E_{ext}(s) = -\gamma \sum_{P \in V_s} e(P) \tag{11}$$

where $e(P)$ is the trackability indicator for part P as defined in (7)[6] and the negative sign reflects the fact that the

[6] For a correct implementation of the segmentation method, the contribution of each energy term must be normalized by dividing the term by the largest value in the neighborhood where the snake point can move: $E(s)/\max_{t \in \mathcal{N}(s)} E(t)$, where $\mathcal{N}(s)$ is the set of pixels in a neighborhood of s.

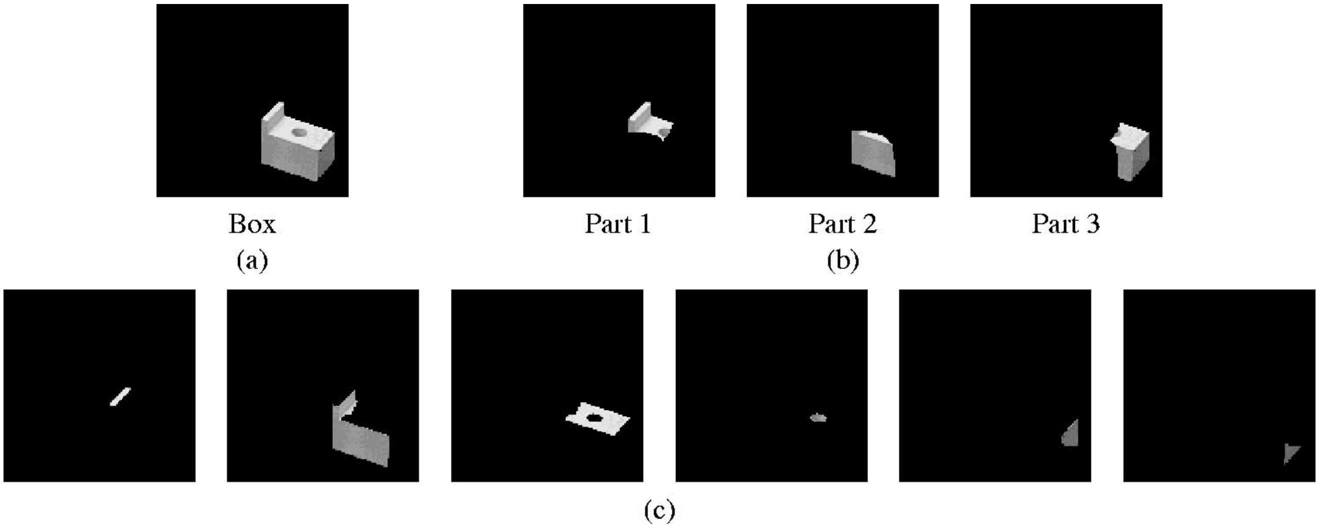

Fig. 9. *Box:* Segmentation (a) Object. (b) The proposed segmentation distributes the hole of the object between Part 1 and Part 3, with the bulk of it in Part 1, allowing the former to grow smaller than the other two parts. (c) MDL-based segmentation.

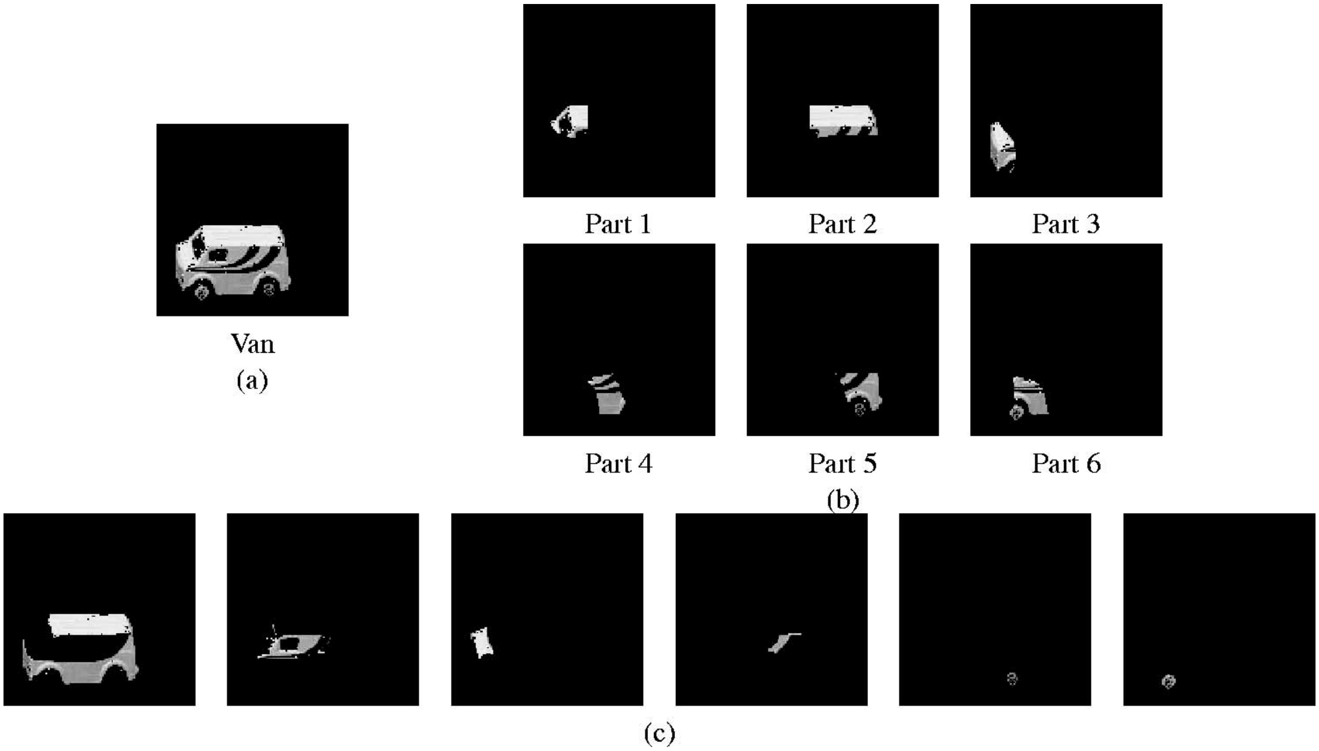

Fig. 10. *Van* segmentation: (a) Object. (b) Proposed segmentation. Even though this object is about the same size in number of pixels as the other examples, the segmentation results in six parts. This is due to the energy being concentrated in several small regions. Part 2 grows larger than the others, to achieve comparable energy. (c) MDL-based segmentation.

snake segmentation algorithm seeks to minimize the global energy.

B. Snake Initialization

A snake in the form of a square grid placed on the object performs the initial segmentation, dividing it into a number of parts. (Other grids could serve as well, triangular, etc.) Fig. 5(b) displays the initial segmentation of object *Cylinder* in Fig. 5(a) into 12 parts (each part shown with a different gray value).

Next, a minimum of the energy is found through a greedy search. The search resembles the segmentation algorithm described in [25] through region competition and merging based on snakes which guarantee closed parts, employing statistics inside the region rather than just information along the region

TABLE IV
COMPARISON BETWEEN THE TRACKING ALGORITHMS

Object	(a) Benchmark	(b) PRA with	(c) PRA with	$\frac{(a)-(c)}{2784}$ Absolute	$\frac{(a)-(c)}{(c)}$ Relative
		homo. parts	prop. parts	% change	% change
Block	2589.4	2214.6	1701.5	31.9	52.2
Log	2059.5	1886.6	1181.4	31.5	74.3
Box	2358.6	2059.0	1337.5	36.7	76.3
Cylinder	2274.1	1937.1	1669.3	21.7	36.2
Van	2286.9	2414.4	1448.6	30.1	57.9
Truck	2011.5	1675.3	740.6	45.7	171.6
Car	1901.4	1080.1	781.1	40.2	143.4
Car_2	2112.5	1942.0	1009.0	39.6	109.4
Compact	2759.3	2531.1	2199.4	20.1	25.5
Compact_2	2678.8	2356.5	1849.6	29.8	44.8
Bus	2671.9	2031.2	1177.7	53.7	126.9
Race	2358.1	1853.5	1846.7	18.4	27.7
Wagon	2578.7	1946.7	1021.2	55.9	152.5
Wagon_2	2091.2	1839.3	855.3	44.4	144.5
Car_3	2166.6	2178.5	799.8	49.1	170.9
Cadillac	2078.1	1823.1	674.8	50.4	208.0
Average	2311.0	1985.6	1268.3	37.5	101.4

boundary, and global optimization techniques based on an energy function.

C. Minimization of the Snake Energy

The experimental results shown in Section III-B indicate that a "good" segmentation for tracking should have parts with high values of "energy" as defined in (7) and hence a low value for the external force (11) term contributing to the snake energy function (9). However, simply minimizing the snake energy function (9) may lead to a segmentation composed of just a few large parts, or, in extreme cases, to a trivial solution with just one part. Clearly these solutions are undesirable in terms of robustness to occlusion. Moreover, as was the case in Section III-B, consistent experience shows that once the energy components of a part rise above a given threshold, little improvement in tracking performance is obtained by increasing them even further. Rather, performance can be improved by attempting to increase the energy components of the remaining parts above that threshold, leading to a segmentation that has more parts, with comparable energy, rather than one having a bimodal energy distribution, with a few high energy parts and several low energy ones.

Finally, note that since the snake energy function (9) is nonconvex, a minimization algorithm may get trapped in a local minimum. To take these effects into account, the external force (11) will be redefined by introducing a filtered version of the energy of the parts

$$E_{ext}(s) = -\gamma \sum_{P \in V_s} \hat{e}(P) \tag{12}$$

where

$$\begin{aligned}\hat{e}(P) =& f(e_x(P), e_{sat}, \frac{e_{sat}}{e'_{sat}}\lambda) + f(e_y(P), e_{sat}, \frac{e_{sat}}{e'_{sat}}\lambda) \\ &+ f(e_{xx}(P), e'_{sat}, \lambda) + f(e_{yy}(P), e'_{sat}, \lambda) \\ &+ f(e_{yx}(P), e'_{sat}, \lambda) + f(e_{xy}(P), e'_{sat}, \lambda)\end{aligned} \tag{13}$$

and

$$f(c, c_{sat}, \lambda) = \frac{1}{1 + \exp^{-\lambda(c - c_{sat})}}. \tag{14}$$

The parameter λ controls the shape of the filter. At the beginning of the optimization, λ is set to 0 to allow "weaker" (low trackability index) parts to grow rather than to merge with "stronger" (high trackability) ones. After the snake reaches an equilibrium point, λ is increased and the process repeated, achieving an effect similar to simulated annealing [5], that minimizes the probability of converging to a local minimum. In the limit as $\lambda \to \infty$, the resulting parts have an energy $\hat{e}$ above c_{sat} (since $f(c, c_{sat}, \infty) = 0$ for $c < c_{sat}$). This process is illustrated in Fig. 6, showing several stages of the segmentation

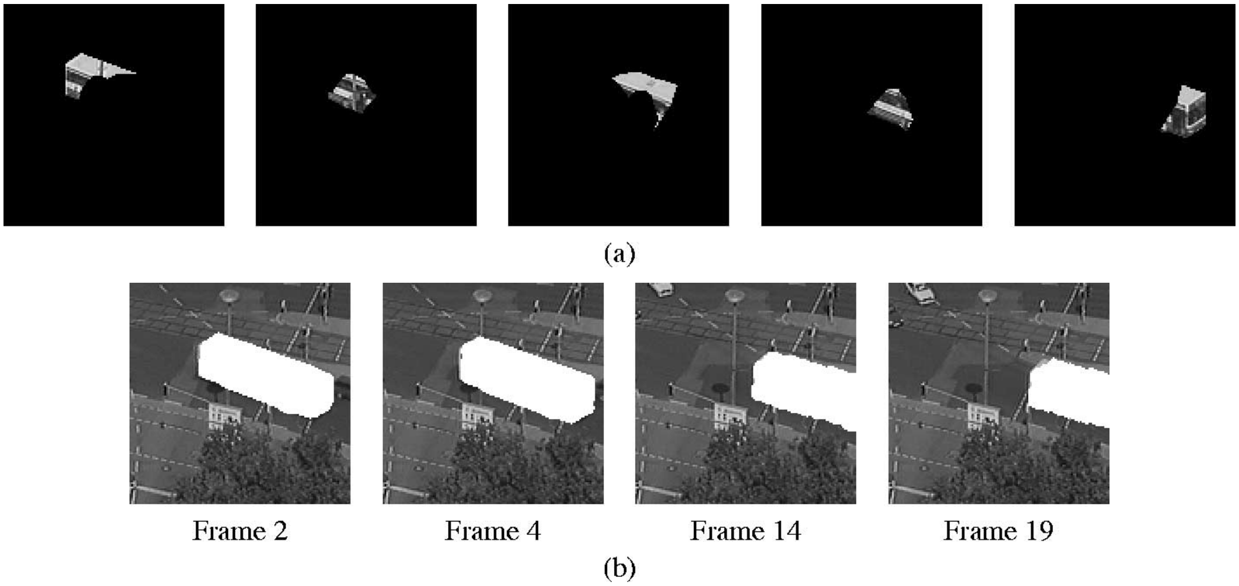

Fig. 11. (a) Parts for the bus shown in Figs. 1 using the proposed segmentation algorithm. (b) Tracking the bus using the parts shown in (a).

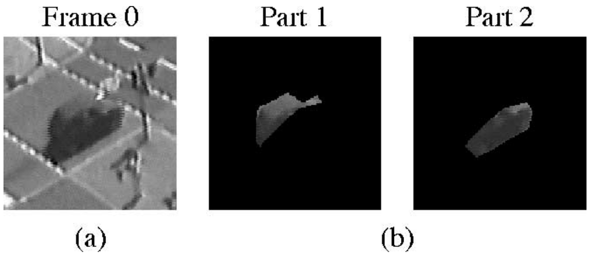

Fig. 12. *Compact* sequence: (a) Frame 0 of the sequence. (b) Parts obtained with the proposed segmentation algorithm. (c) The benchmark algorithm is not affected by the mild occlusion caused by a traffic sign but it starts losing track of the car by Frame 17 when about 30% of the car is out of the field of view. (c) Tracking with homogeneous parts leads to poor results by Frame 8 due to mild occlusion by a traffic sign. (d) Tracking the parts shown in (b) is successful throughout the entire sequence until the car is last visible in Frame 28 with about 90% of it out of the field of view.

algorithm. Here we used the values $\alpha = \beta = \gamma = 1.0$, giving equal weight to the eight energy components.

The final segmentation is shown in Fig. 7. Since most of the texture content is concentrated in the upper portion of the object, the segmentation distributes this portion amongst the three parts. Although Part 3 grows much larger than Part 2, as far as energy is concerned, they have about the same amount since the upper portion of the object is divided about evenly between them. The lack of texture and gradient content on the lower portion of the object forces the two parts which share it to be elongated. An alternative partition might have created an additional bottom part taken from Part 2 and Part 3 (for a total of four). However, this part would have virtually no energy, thus creating a bad tracking part. As we will show in Section V, an abundance of bad parts proves as detrimental to the PRA as a lack of good parts.

D. Segmentation Results

Figs. 8–10 illustrate the use of the proposed segmentation with three toy objects, selected to exhibit both large regions with homogeneous texture as well as regions with contrast texture. For comparison purposes we also include the results of an MDL-based segmentation [12].

It is worth noticing that in all cases the proposed segmentation leads to parts that are more compact than those resulting from the homogeneous one. In addition, in those cases were the energy is evenly distributed through the object, it results in fewer parts.

V. Tracking Results

In this section we present a series of experiments comparing tracking using the proposed parts against homogeneous parts obtained using a Mininum Description Length based algorithm [12]. For comparison purposes, we also include the results obtained using the Benchmark Algorithm to track the object as a whole.

These experiments were performed using six toy objects, supplemented with ten objects taken from real tracking sequences. As in Section III, for each of the 16 test objects we performed 480 tests:

1) 16 “challenging” poses of Table II;
2) five cluttered backgrounds (similar to Fig. 3(b), containing objects with similar texture to the test objects, and the scene corrupted by zero-mean additive white Gaussian noise with variance 5);
3) six translations of 50% occlusion (measured in the number of prototype pixels).

The resulting scores, ranging from zero to 2784, are shown in Table IV.

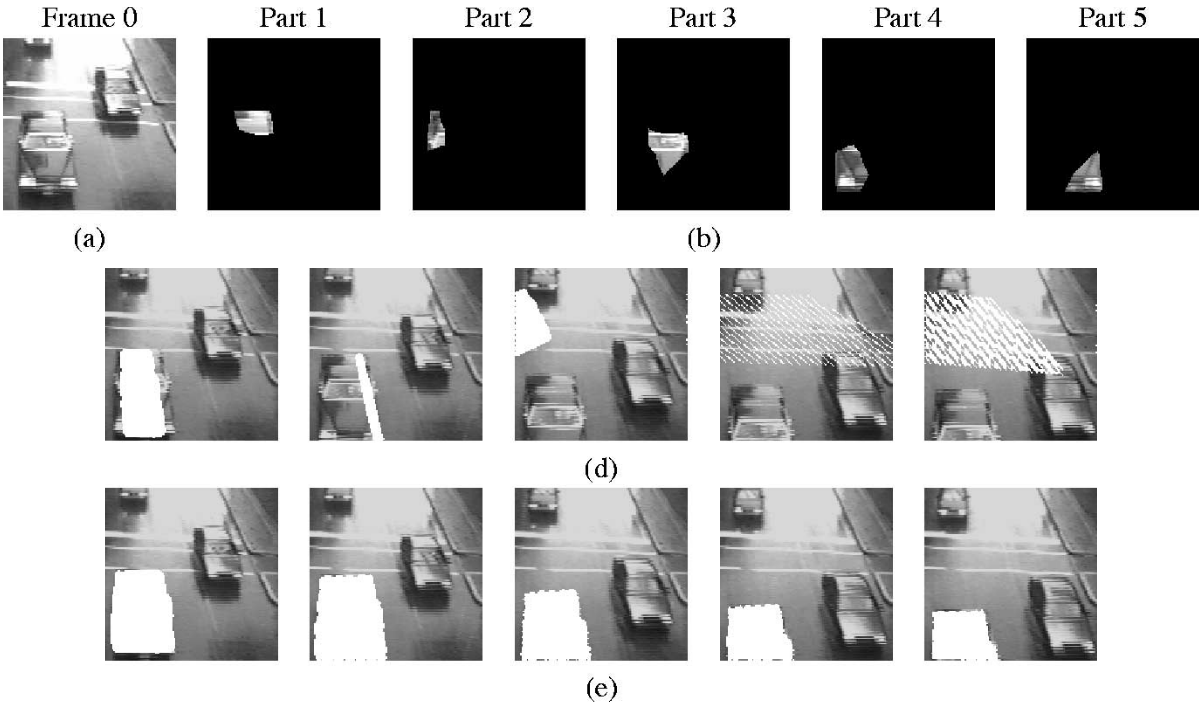

Fig. 13. *Cadillac* sequence: (a) Frame 0. (b) Parts obtained with the proposed segmentation algorithm. (c) The benchmark algorithm begins to lose track of the object in Frame 3, as the target starts moving out of the field of view. (d) Tracking with homogeneous parts loses track by Frame 2, due to a bad part. (e) Tracking the parts shown in (a) the PRA is able to track the object through Frame 10, in spite of the large perspective distortion and lighting changes.

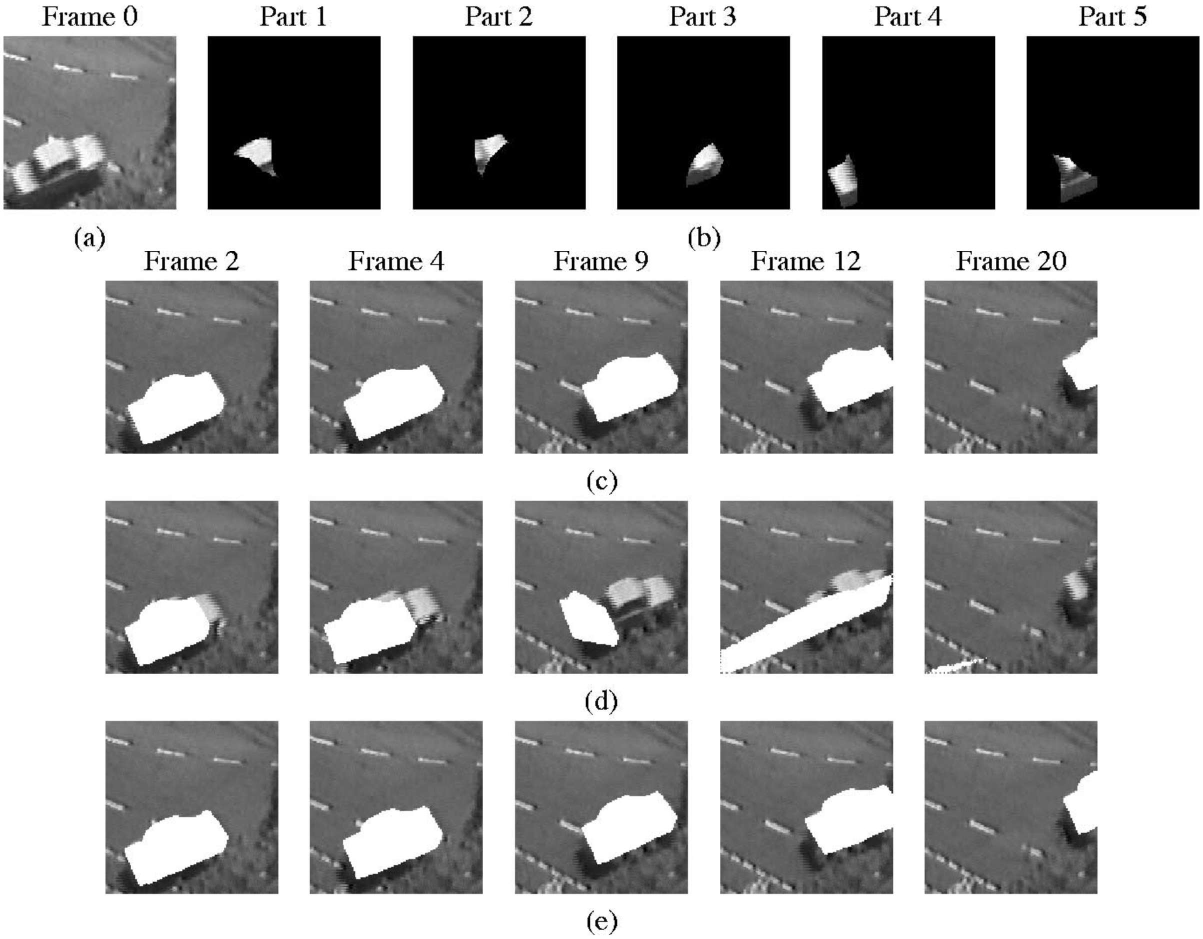

Fig. 14. *Car* sequence: (a) Frame 0. (b) Set of parts obtained with the proposed segmentation algorithm. (c) The robust estimator used by the benchmark algorithm is effective in determining occluding pixels. (d) Tracking using homogeneous parts fails after Frame 2. (e) Tracking using the segmentation shown in (b) is successful until the object is last visible in Frame 20.

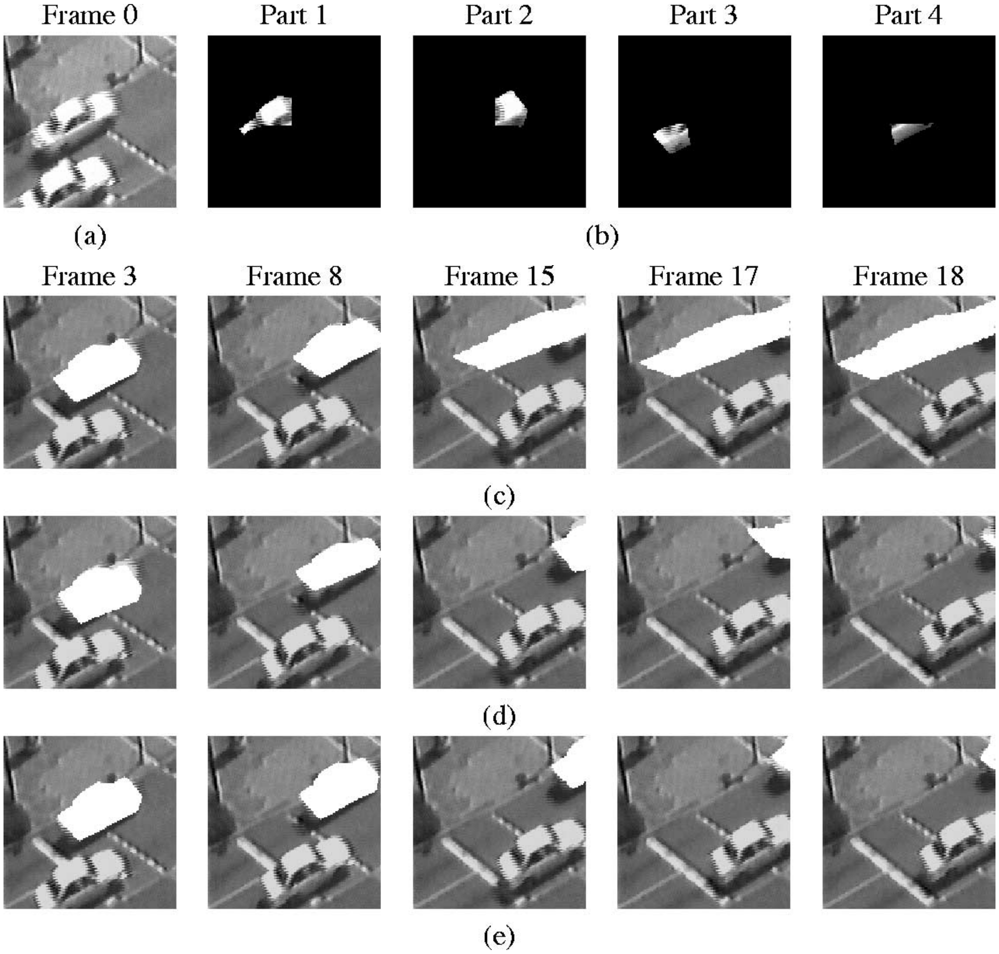

Fig. 15. *Car_2* sequence: (a) Frame 0. (b) Proposed segmentation of the target. (c) Tracking with the benchmark algorithm looses track of the target by Frame 15 at 50% occlusion. (d) Tracking using homogeneous parts starts to fail at Frame 17 due to severe occlusion. (e) Tracking using the proposed segmentation is successful throught the entire sequence until Frame 18 with 90% occlusion.

The PRA with the proposed parts (c) outperforms both the Benchmark algorithm (a) and the PRA with homogeneous parts (b) for each of the test objects.

- (c) outperforms (a) by up to 208% (*Cadillac*), and on average by 101%.
- (c) outperforms (b) by up 172% (*Car_3*), and on average by 72%.

It is worth noticing that (a) outperforms (b) in some occasions, illustrating again that using an homogenous segmentation can lead to worse results than not using parts at all.

Fig. 11(a) shows the proposed segmentation for the bus in the sequence of Figs. 1 and 11(b) shows the tracking results using this segmentation. In this case the algorithm is able to successfully track the target throughout the entire sequence until the film ends at about 60% occlusion.

Finally, Figs. 12–16 show experimental results obtained with other sequences[7]. In all cases, (c), (d), and (e) denote tracking the whole object using the Benchmark Algorithm, tracking using homogenous parts, and tracking using the proposed segmentation, respectively.

A. Compact Sequence

Using homogeneous parts leads to poor tracking after Frame 8: the mild occlusion caused by the traffic sign compresses a "bad" part into its unoccluded portion in order to minimize the error norm, and the object follows. The part continues to compress the object in the subsequent frames and eventually loses track of it completely. The traffic sign does not affect (c) and (e) at all, however by Frame 17, at about 30% occlusion, (d) has begun to lose track of the object, while (e) successfully tracks the object throughout the entire sequence until it is last visible in Frame 28 at about 90% occlusion.

B. Cadillac Sequence

(c) begins to lose track of the object in Frame 3 as occlusion sets in. (d) begins to lose track of the object in Frame 2 due to a bad part, while (e) tracks the object through Frame 10, in spite of the large perspective distortion and lighting changes from the prototype in Frame 0.

C. Car Sequence

(d) begins to track poorly in Frame 2: one of its parts corresponding to the lower side of the object (the dark strip) contains

[7] Additional experiments, omitted for space reasons, can be obtained contacting the authors.

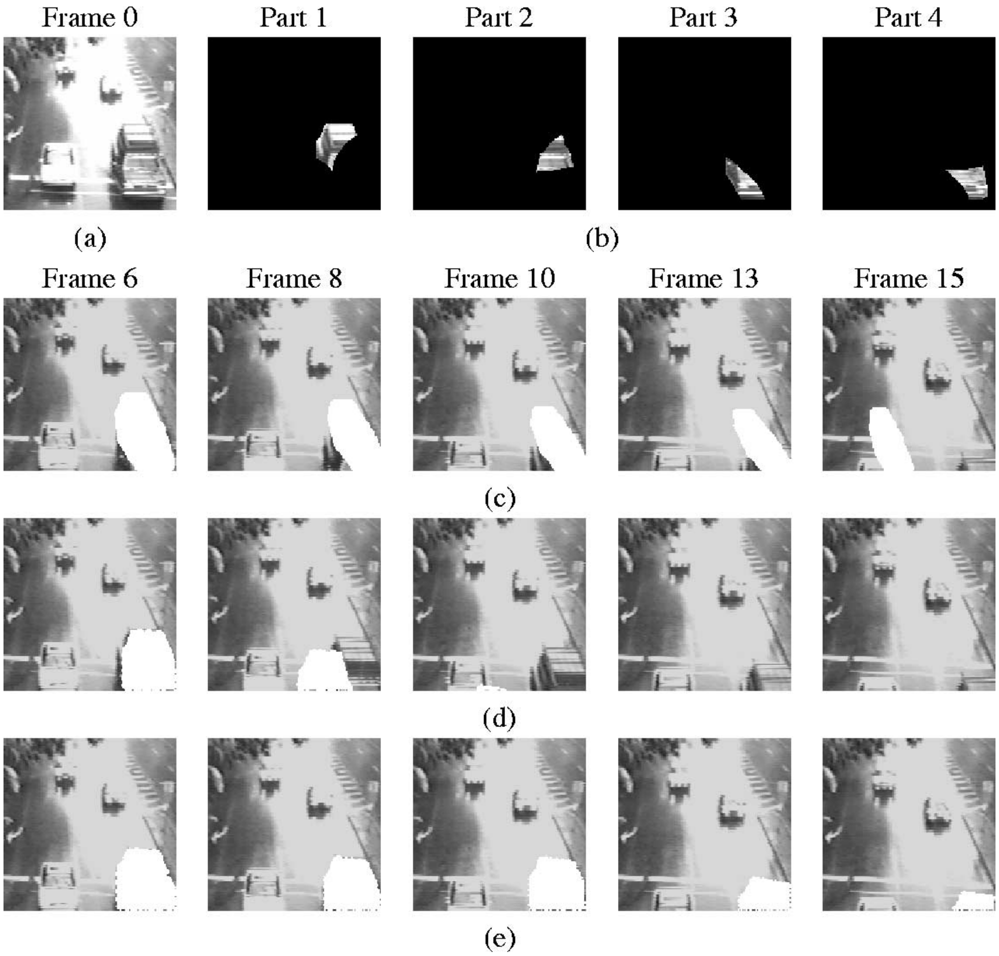

Fig. 16. *Wagon_2* sequence: sequence: (a) Frame 0. (b) Proposed segmentation of the target. (c) Tracking with the benchmark algorithm begins to fail by Frame 3 as the target leaves the field of view. (d) Tracking using homogeneous parts starts to fail at Frame 6 at about 40% occlusion. (e) Tracking using the parts shown in (b) successfully tracks the target throughout the sequence until Frame 15 at about 90% occlusion.

virtually no gradient content. As the object advances through the sequence, the part cannot provide information on which direction to move. By remaining static it finds a deceptively low error norm in a local minimum: the static, compressed strip looks the same as the tail portion of the dynamic strip. The strip lures the other parts into its transformation and continues to compress the object in the subsequent frames and eventually loses track of it completely. This demonstrates that the presence of "bad" parts negatively influences a segmentation as much as the absence of good parts. (c) and (e) however track the object throughout the entire sequence until it is last visible in Frame 20 as it disappears into the trees at about 80% occlusion. The robust estimator in (c) reliably determines the occluded pixels.

D. *Car_2 Sequence*

(c) has already lost track of the object by Frame 15 at about 50% occlusion. (d) tracks the object through Frame 17, but loses it due to severe occlusion. (e) however tracks the object throughout the entire sequence until it is last visible in Frame 18 at about 90% occlusion. In addition (e) offers a higher quality of match than (d), as evident in Frame 3 and Frame 15, and especially in Frame 8.

E. *Wagon_2 Sequence*

By Frame 6 at about 40% occlusion (d) is beginning to lose track of the object while (c) has already lost it. (d) loses the object completely in Frame 8. On the other hand, (e) tracks the object throughout the entire sequence until it is last visible in Frame 15 at about 90% occlusion, despite the large perspective distortion and lighting changes from the prototype in Frame 0.

VI. Conclusions

Many tracking algorithms used widely in the computer vision community deal with occlusion through a robust estimator. Such estimators fare well with moderate occlusion, but break down at above 30% occlusion level. To expand this range, in [6] we have proposed to track, in addition to the object, a set of parts. Intuitively, this idea exploits the fact that occlusion tends to be localized, and thus reliable tracking can be accomplished as long as a few of these parts exhibit less than 30% occlusion. However, as illustrated with several examples, successful application of this idea requires a suitable object segmentation. In this paper we have identified desirable properties (from a robust tracking standpoint) for the parts and proposed an energy function to obtain these parts by solving an optimization problem. Experimental results with both synthetic and real images show that, when used in a context of a tracking algorithm, these parts outperform those obtained using traditional segmentation methods.

VII. Acknowledgements

The authors are indebted to an anonymous reviewer for many suggestions to improve the original manuscript.

References

[1] M. J. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *Proc. IEEE ICCV*, May 1993, pp. 231–236.

[2] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. Eur. Conf. Computer Vision*, vol. 1064, Apr. 1996, pp. 329–329.

[3] A. Blake and M. Isard, "Condensation – condensation density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28,1998.

[4] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. M. Tekalp, "Motion segmentation by multistage affine classification," *IEEE Trans. Image Processing*, vol. 6, pp. 1591–1594, Nov. 1997.

[5] N. K. Bose and P. Liang, *Neural Network Fundamentals*. New York: McGraw-Hill, 1993, pp. 318–323.

[6] C. Gentile, "Robust Tracking With Parts in the Presence of Severe Occlusion," Ph.D. dissertation, Pennsylvania State Univ., University Park, 2001.

[7] J. Ferruz and A. Ollero, "Real-time feature matching in image sequences for nonstructured environments. Applications to vehicle guidance," *J. Intell. Robot. Syst.*, vol. 28, no. 1, pp. 85–123.

[8] E. Grossmann and J. Santos-Victor, "Performance evaluation of optical flow estimators: Assessment of a new affine flow method," *Robot. Auton. Syst.*, vol. 21, pp. 69–82, 1997.

[9] N. Gupta and L. Kanal, "Gradient based image motion estimation without computing gradients," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 81–101, Feb.-Mar 1997.

[10] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 10, pp. 1025–1039, 1998.

[11] D. J. Kang and I. S. Kweon, "A visual tracking algorithm by integrating rigid model and snakes," in *Proc. IEEE Conf. Intelligent Robots and Syst.*, vol. 2, Nov 1996, pp. 777–784.

[12] T. Kanungo, B. Dom, W. Niblack, and D. Steele, "A fast algorithm for MDL-based multi-band image segmentation," in *Proc. IEEE Conf. CVPR*, Jun 1994, pp. 609–616.

[13] D. E. Knuth, *The Art of Computer Programming*. Reading, MA: Addison-Wesley, 1969, vol. 2, pp. 34–54.

[14] A. Leonardis and H. Bischof, "Dealing with occlusion in the eigenspace approach," *Proc. IEEE Conf. CVPR*, pp. 453–458, June 1996.

[15] W. Y. Ma and B. S. Manjunath, "Edgeflow: A technique for boundary detection and image segmentation," *IEEE Trans. Image Processing*, vol. 9, pp. 1375–1388, Aug. 2000.

[16] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and classification of complex dynamics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 9, pp. 1016–1034, 2000.

[17] F. Pedersini, A. Sarti, and S. Tubaro, "Accurate feature detection and matching for the tracking of calibration parameters in multi-camera acquisition systems," in *Proc. IEEE Conf. Image*, vol. 2, Oct. 1998, pp. 598–602.

[18] S. Sclaroff and J. Isidoro, "Active blobs," in *Proc. IEEE ICCV*, Jan. 1998, pp. 1146–1153.

[19] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. CVPR*, June 1994, pp. 593–600.

[20] A. Singh and M. Shneier, "Grey level corner detection: A generalization and a robust real time implementation," *Comput. Vis. Graph. Image Process.*, vol. 51, no. 1, pp. 54–69, July 1990.

[21] D. Terzopoulos and K. Fleischer, "Deformable models," *Vis. Comput.*, vol. 4, pp. 306–331.

[22] C. Toklu, A. T. Erdem, M. I. Sezan, and A. M. Tekalp, "Tracking motion and intensity variations using hierarchical 2-D mesh modeling for synthetic object transfiguration," *Graph. Mod. Image Process.*, vol. 58, no. 6, pp. 553–573, Nov. 1996.

[23] F. de la Torre, S. Gang, and S. McKenna, "View-based adaptive affine tracking," *Lecture Notes in Computer Science*, vol. 1406, pp. 828–842, 1998.

[24] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *INRIA*, vol. 2273, May 1994.

[25] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/MDL for mulitband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.

Camillo Gentile received the B.S. and M.S. degrees in electrical engineering in 1996 from Drexel University, Philadelphia, PA, and the Ph.D. degree in 2001 from The Pennsylvania State University, University Park. His doctoral work focused on research on computer vision and neural networks.

Upon graduation, he joined the Wireless Communications Technologies Group of the National Institute of Standards and Technologies, Gaithersburg, MD. His current projects deal with the design of power-efficient routing algorithms and radio localization and tracking in mobile ad-hoc networks.

Octavia Camps received the B.S. degree in computer science and the B.S. degree in electrical engineering from the Universidad de la Republica, Montevideo, Uruguay, in 1981 and 1984, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1987 and 1992, respectively.

In 1991, she joined the faculty of The Pennsylvania State University, University Park, where she currently is an Associate Professor with the Departments of Electrical Engineering and Computer Science and Engineering. In 2000, she was a Visiting Faculty at the California Institute of Technology, Pasadena, and at the University of Southern California. Her current research interests includerobust computer vision, image processing, and pattern recognition.

Mario Sznaier received the Ingeniero Electronico and Ingeniero en Sistemas de Computacion degrees from the Universidad de la Republica, Uruguay, in 1983 and 1984, respectively, and the MSEE and Ph.D degrees from the University of Washington in 1986 and 1989, respectively.

From 1991 to 1993, he was an Assistant Professor of Electrical Engineering at the University of Central Florida. In 1993, he joined The Pennsylvania State University, where he currently is a Professor of electrical engineering. He has also held visiting appointments at the California Institute of Technology in 1990 and 2000. His research interest include multiobjective robust control; l1 and H-infinity, control theory, control oriented identification and active vision.