Intelligent Visual Descriptor Extraction from Video Sequences

Paraskevi Tzouveli, Georgios Andreou,
Gabriel Tsechpenakis, *Member IEEE*, Yiannis Avrithis, *Member IEEE*,
and Stefanos Kollias, *Member IEEE*

Image, Video and Multimedia Systems Lab.
School of Electrical and Computer Engineering
National Technical University of Athens
iavr@image.ntua.gr

Abstract. Extraction of visual descriptors is a crucial problem for state-of-the-art visual information analysis. In this paper, we present a knowledge-based approach for detection of visual objects in video sequences, extraction of visual descriptors and matching with pre-defined objects. The proposed approach models objects through their visual descriptors defined in MPEG7. It first extracts moving regions using an efficient active contours technique. It then computes visual descriptions of the moving regions including color, motion and shape features that are invariant to affine transformations. The extracted features are matched to a-priori knowledge about the objects' descriptions, using appropriately defined matching functions. Results are presented which illustrate the theoretical developments.

1 Introduction

An *Information Retrieval System (IRS)* consists of a database containing a number of documents, an index that associates each document to its related terms, and a matching mechanism that maps the user's query (consisting of terms), to a set of associated documents [1]. In the case of multimedia documents, the content of the document cannot be directly used by the user of the IRS in the query, since matching of multimedia content is not as simple as matching of textual terms and features of the content must be used instead. The needs for description of multimedia documents' content have been addressed by MPEG-7, the ISO standard for description of multimedia content [10]. A large number of MPEG-7 compliant multimedia descriptions are currently being produced. The standard defines three kinds of features that comprise the description, which are *Creation and Usage Information*, *Structural Information* and *Semantic Information*. The former regards mostly textual information, commonly known as *metadata*. Structural information expresses a low-level and machine-oriented kind of description, since they describe content in the form of signal segments and their properties. On the other hand, semantic information expresses a high-level, conceptual and human - oriented kind of description, since they deal with

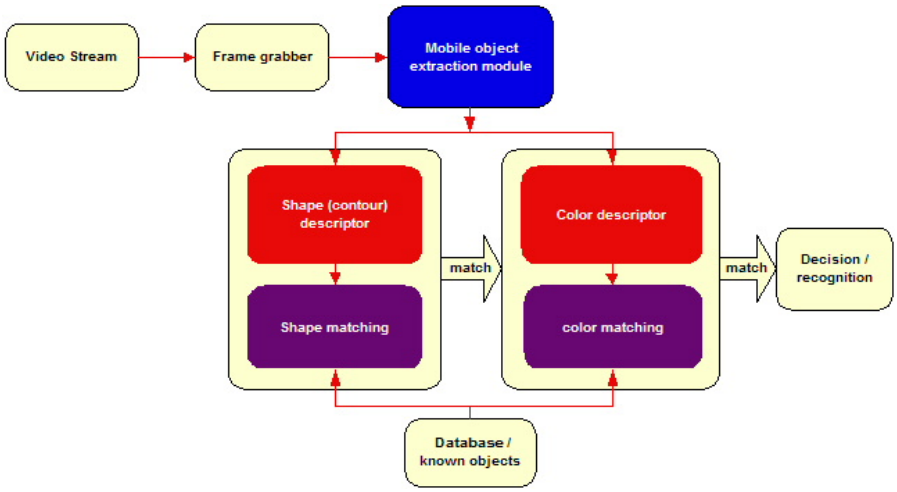


Fig. 1. The proposed integrated scheme for object recognition, using the shape and color descriptors.

semantic entities, such as objects and events.
In this paper we focus on a specific task of multimedia content description, i.e the detection and recognition of objects being present in a video stream, whose dominant characteristic is their motion. The extraction of moving objects in video streams and their description with the use of low-level feature matching, is a task that emerges in various applications in the fields of video understanding, such as content-based retrieval and semantic description of events. This work constitutes an integration of three steps for object recognition, revisiting and improving existing methods found in literature. The three steps being followed are illustrated in Fig. 1 and can be briefly described as follows. The moving objects of interest are extracted, with the use of a tracking method proposed in [12], which utilizes an active contour (modified Snake) model and the motion information obtained by a motion estimation scheme. Once the desired objects are extracted, i.e their position and contour are estimated for each frame of the sequence, color descriptors are extracted and their shape is appropriately modelled and transformed, so that it becomes affine invariant. The final step of the overall scheme is the matching of the color and shape descriptors with the respective ones of known objects, existing in a database. In the experiments presented in this paper, we use three different objects of either the same color or the same shape, to verify the performance of the proposed integrated scheme in ground-truth examples. More complicated examples of object recognition are currently being tested, with the use of a database and an efficient searching procedure in terms of complexity. Finally, for more sophisticated applications such as the semantic description of events, the motion trajectory of the desired objects is to be utilized, in order to

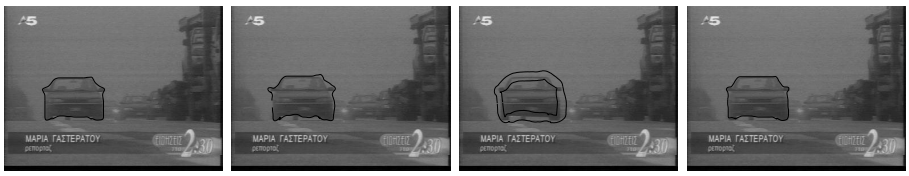
obtain further useful information about the objects' *global* motion, apart from their instant motion, provided by motion estimation schemes.

2 Moving Object Extraction

Efficient moving object extraction in real-world conditions is a challenging task for the researchers in the fields of computer vision and video processing. In modern coding standards, like MPEG-4 and MPEG-7, the term 'video objects' is used to define moving objects in a video sequence. Automatic extraction of such objects is by no means trivial, and occlusion is one of most important problems. In this paper we implement and extend the work presented in [12] for object tracking, in order to support highly textured backgrounds and partial occlusion of the moving objects.
In [12] object tracking is performed utilizing a snake model [8] and the motion information obtained in previous time instances, or motion history. Regarding the proposed snake model, its internal energy is defined in terms of the local curvature and elasticity (distances between neighboring points), whereas the external energy term is defined with the use of a modified image gradient, replacing the commonly used term $|\nabla G_\sigma * I|$ [7], which introduces noise in the snake models. More information about the definitions of the proposed energy terms can be found in [12].
Before applying the tracking model in the current frame of a sequence, as described in the following, we pre-process the image to eliminate noise, with the use of an appropriate morphological Alternating Sequential Filter (ASF) [12, 9]. The modified image gradient used for our purposes is actually a part of the Watershed transformation in image segmentation problems [9] and consists of the extraction of binary image markers through a morphological geodesic erosion reconstruction of the image gradient, and successive morphological conditional erosions of these markers, so that they constitute the only local minima of the image gradient.

2.1 Motion Estimates Extraction

The correct extraction of moving edges in terms of position and direction is important and aids the accurate estimation of an object's position from the current to the next frame. Several existing techniques are able to adequately cope with the difficult problem of optical flow recovery given that their assumptions hold. The challenge is to achieve high robustness against strong assumption violations commonly met in real sequences. We adopt the motion estimation technique proposed by Black et al. [4] as an efficient tool for overcoming these violations. They reformulate the objective function, which consists of the optical flow equation and the spatial coherence constraint, in order to include the robust statistics tools [6] in an almost straightforward way. They simply take the standard least-squares formulation of optical flow and use a robust estimator instead of the quadratic one. This approximation is then minimized using a coarse-to-fine



(a) (b) (c) (d)

Fig. 2. Tracking method in steps: (a) object contour in the previous frame, (b) snake initialization in the current frame, (c) uncertainty region, (d) object contour in the current frame.

(multiresolution) simultaneous over-relaxation technique. The proposed reformulation results in an area-based regression technique that is robust to multiple motions due to occlusion, transparency or specular reflections and compensates for over-smoothing and noise sensitivity.

2.2 Object Tracking

Given the proposed snake model presented in [12], the first step is to extract some regions (a narrow band) around the curve, which are described as *uncertainty regions* (Fig. 2). This is achieved by exploiting the motion history of the tracked contour (curve points' motion in previous time instances), estimated with the use of the motion estimation scheme proposed in subsection 2.1: the previously estimated contour (Fig. 2(a)) is deformed according to the previously estimated motion (snake initialization) (Fig. 2(b)) and the standard deviation of each point's mean motion is calculated; the uncertainty region around each point is then the region in the normal direction to the snake initialization, whose width is defined according to the corresponding standard deviation (Fig. 2(c)). The next step is to find the new position of each point of the curve, inside its corresponding uncertainty region (Fig. 2(d)): instead of following an energy minimization procedure, using a dynamic programming algorithm, we adopt a force-based approach, which reduces the computational cost but also avoids the point correspondence problem between different time instances.

According to that approach, energy terms are converted into forces and the final solution is obtained by minimizing the resultant force [12] inside the extracted uncertainty regions. The internal forces deform the snake to a shape similar to the previously estimated object contour, whereas the external term forces the snake towards the object boundaries, inside the extracted overall uncertainty region. Thus, the energy minimization is approximated by using these forces, in an iterative manner similar to the steepest descent approach [5].

The resultant force applied to each snake point is given by the weighted summation of the internal and external forces. The respective weights are automatically estimated [12], whereas their estimation accuracy is not crucial for the final results. The final object contour is obtained when one of the following criteria is satisfied: (a) if the resultant force is smaller than the one of the next iteration, or

(b) the maximum number of iterations is reached. It must be noted that the use of the proposed steepest descent approach does not ensure that the final contour corresponds to the solution of the energy minimization problem, but under the constraints we pose, even if the final contour corresponds to a local minimum, it is close to the desired solution (global minimum).
In order to separate background and object regions, especially when the background contains strong edges close to the object boundaries, as well as to cope with moving object's partial occlusion that may occur, we introduce two additional constraints that each detected edge point must obey, so that we can decide whether this edge belongs to the desired boundary; all candidate edges are indicated by the snake's external energy, and consequently by the modified image gradient.
Without loss of generality, we suppose that the background is static and possible occluding objects are also static. If $\tilde{p}_k$ is a detected candidate (possible boundary) edge pixel, and p_l and p_m are the neighboring pixels in both sides of $\tilde{p}_k$, in the normal direction to the snake initialization (Fig. 2(b)), then (a) $\tilde{p}_k$ must divide that line segment in two parts: an immiscibly moving and a immiscibly static one, that is $u(p_l) \simeq u(\tilde{p}_k)$ and $u(p_m) \simeq 0$, and (b) $\tilde{p}_k$ must be a moving point with velocity close to the mean velocity of the object region, that is $u(\tilde{p}_k) \simeq \overline{u}_{object}$; $u(\cdot)$ and $\overline{u}_{object}$ denote the instant velocity and the object mean velocity, obtained by the motion estimation scheme described in 2.1.
Thus, taking the above constraints into consideration, we overcome cases such as (a) when the maximum is found in background: it is not a moving one and does not separate two immiscible (according to the motion) parts of function g_m [12], (b) when the maximum is found inside the moving object region: although it is a moving one, it does not divide function g_m in such two parts, (c) when occlusion occurs and the maximum is on the occluding object boundary: the maximum is not moving, although it separates the uncertainty region and (d) when occlusion occurs and the maximum is in the occluding object region: neither the maximum is moving, nor it makes such a separation. In these cases, where these two constraints are not reached, we ignore the external force and the curve evolves according to its internal forces; in this way, we can obtain contours similar to the ones in the past frames. Fig. 3 illustrates the performance of the proposed method in a case of two moving objects: one getting partially occluded by a static obstacle and the other moving in front of it. The adopted motion estimation technique allows the utilization of the two rules described above, in order to separate the moving objects from the static regions (background and obstacle) of the scene.

3 Visual Descriptors

In the following, some visual descriptors, which have been introduced in the integrated scheme, are briefly revised according to the MPEG-7 framework [10]. The Dominant Color descriptor, illustrated in the experiments of this paper, is presented in more detail in subsection 3.1.

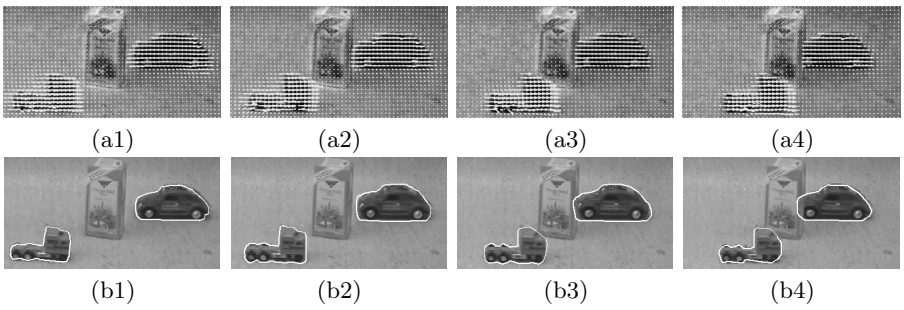


Fig. 3. (a1)-(a4) Motion estimation results and (b1)-(b4) the respective tracking results for a case of two moving objects.

Dominant Color. The dominant color descriptor specifies a set of dominant colors in any arbitrary shaped region. The extraction algorithm takes as an input a set of color values and quantizes the image color vectors based on the Generalized Lloyd Algorithm (GLA), as described in Section 3.1.
Region Contour. As contour shape descriptor an affine-invariant normalization of the extracted object contours is used, as described in Section 3.2.

3.1 Color Descriptor

The Dominant Color descriptor used in our experiments to illustrate color matching of visual objects is described in more detail below. This descriptor provides a compact description of the representative colors of an image or image region. Its main target applications are similarity retrieval in image databases and browsing of image databases based on single or several color values. The representative colors can be indexed in the $3D$ color space, which allows for efficient indexing of large databases. In its basic form, the Dominant Color descriptor consists of the number of dominant colors N_D, and for each dominant color its value is expressed as a vector of color components c_i and the percentage of pixels p_i in the image region of the corresponding cluster [10].
In order to compute this descriptor, the colors present in a given image or region are first clustered. Instead of the Generalized Lloyd Algorithm [10], the extraction procedure uses a fuzzy c-means algorithm [3] for the dominant color, to divide the set of pixel values corresponding to a given image region into clusters in the color space. The algorithm minimizes the supremum of the distance between the color pixel values and the representative color vectors using the global distortion measure J defined as

$$J = \sum_{j=1}^{N_c} \sum_{i=1}^{N_i} u_{ij}^m \|x_i - co_j\|^2, \tag{1}$$

where N_c is the number of clusters, N_j is the number of pixels of the j-th cluster, x_i is the i-th color vector, c_j is the center (representative color) of the j-th

cluster and $\mu_{i,j}$ is the degree of membership of x_i in the cluster c_j. The procedure is initialized with a predefined number of clusters N_D whose representative colors are computed as the centroid (center of mass) of each cluster. Then, the algorithm follows a sequence of centroid calculation and clustering steps until a stopping criterion (minimum distortion or maximum number of iterations) is met.

3.2 Shape Descriptor

As shape descriptor we use an affine-invariant normalization of the object contours extracted by the tracking algorithm described in Section 2. The obtained contours are first re-sampled so that they constitute of a fixed number of equidistant points, also preserving their original shape. In the following, we describe the normalization method that transforms the object contours in order to make them affine invariant, and thus appropriate for contour matching and recognition [2].

Curve Othrogonalization The proposed procedure normalizes a curve with respect to possible translation, skewing, and scaling, and affine transformation as rotation or reflection. Let $C_i = [x_i, y_i]^T$, $i = 0, 1, .., N-1$, be N curve points obtained by the tracking algorithm. A $2 \times N$ matrix notation $\mathbf{C} = [C_0, C_1, ..., C_{N-1}]$ is used to represent the points, while their horizontal and vertical coordinates are represented by $x = [x_0, x_1, ..., x_{N-1}]$ and $y = [y_0, y_1, ..., y_{N-1}]$. For each curve $\mathbf{C}$, the (p, q)-order moments

$$m_{pq}(\mathbf{C}) = \frac{1}{N} \sum_{i=0}^{N-1} (x_i^p y_j^q) \tag{2}$$

of order up to two are used for the construction of the *normalized curve* $n_a(\mathbf{C})$. A set of linear operations (translation, scaling and rotation) in the curve are computed during the orthogonalization procedure:

1. The center-of-gravity of the curve is normalized so as to coincide with the origin:
$$x_1 = x - \mu_x, \;\; y_1 = y - \mu_y \tag{3}$$
where $\mu_x = m_{10}(\mathbf{C})$, $\mu_y = m_{01}(\mathbf{C})$.
2. The curve is scaled horizontally and vertically so that its second-order moments become equal to one:
$$x_2 = \sigma_x x_1, \;\; y_2 = \sigma_y y_1 \tag{4}$$
where $\sigma_x = \frac{1}{\sqrt{m_{20}(C_1)}}$, $\sigma_y = \frac{1}{\sqrt{m_{02}(C_1)}}$,
3. The curve is rotated counterclockwise by $\theta_0 = \frac{\pi}{4}$ as follows :
$$C_3 = R_{\frac{\pi}{4}} \cdot C_2 = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} x_2 - y_2 \\ x_2 + y_2 \end{bmatrix} \tag{5}$$

4. The curve is scaled again, exactly as in (2):

$$x_4 = \tau_x x_3, \quad y_4 = \tau_y y_3 \tag{6}$$

where $\tau_x = \frac{1}{\sqrt{m_{20}(C_3)}}, \tau_y = \frac{1}{\sqrt{m_{20}(C_3)}}$

The normalized curve $n_a(\mathbf{C}) \equiv C_4$ can also be written as

$$n_\alpha(\mathbf{C}) = N(\mathbf{C})(\mathbf{C} - \mu(\mathbf{C})) = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} \tau_x & 0 \\ 0 & \tau_y \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \cdot \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix} \tag{7}$$

where $\mu(\mathbf{C}) = [m_{10}(\mathbf{C}) \;\; m_{01}(\mathbf{C})]^T$ and $N(\mathbf{C})$ denotes the 2×2 normalization matrix of $\mathbf{C}$. It can be seen in [2] that for each initial curve $\mathbf{C}$, the normalized curve $n_a(\mathbf{C})$ defined in eqs. (2)-(6) has the following properties:

$$\begin{aligned} m_{10}(n_a(\mathbf{C})) = m_{01}(n_a(\mathbf{C})) = m_{11}(n_a(\mathbf{C})) = 0, \\ m_{20}(n_a(\mathbf{C})) = m_{02}(n_a(\mathbf{C})) = 1 \end{aligned} \tag{8}$$

The term orthogonalization is justified since these conditions are equivalent to $n_a(\mathbf{C}) \cdot n_a(\mathbf{C})^T = \mathbf{I}$. Let us now consider two curves $\mathbf{C}$ and $\mathbf{C}'$ related through an affine transformation:

$$\mathbf{C}' = A \cdot \mathbf{C} + t = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{9}$$

where matrix A is assumed to be of full rank. Then, $C_1' = \mathbf{C}' + \mu(\mathbf{C}') = A(\mathbf{C} - \mu(\mathbf{C})) = A \cdot C_1$ and translation is removed. Moreover, when a normalized curve is rotated or reflected, in which case A is orthogonal, it remains normalized. It is thus shown in [2] that there exists an orthogonal 2×2 matrix Q such that:

$$n_a(\mathbf{C}') = Q \cdot n_a(\mathbf{C}) \tag{10}$$

This means that affine transformations are reduced to orthogonal ones that may contain only rotation and/or reflection, depending on whether $det(Q) = 1$ or $det(Q) = -1$. Therefore normalized curves are invariant to translation, scaling, and skew transformations. Note that normalization is performed without knowledge of the affine parameters A and t, and without one-to-one matching between curves $\mathbf{C}$ and $\mathbf{C}'$. In addition the transformation parameters $(\mu_x, \mu_y, \sigma_x, \sigma_y, \tau_x, \tau_y)$ along with $n_a(\mathbf{C})$ contain all information on the original curve $\mathbf{C}$.

Starting Point and Rotation Normalization The starting point normalization procedure is based on the Discrete Fourier Transform (DFT) of the complex vector $z = x + jy = [z_0 z_1 ... z_{N-1}]^T$ which is used here for curve representation, where $z_i = x_i + jy_i, \;\; i = 0, 1, \ldots, N-1$, denotes a single curve point. The DFT of the curve z is given by:

$$u = [u_k] = \sum_{i=0}^{N-1} z_i \cdot w^{-ki} \;\; , \;\; k = 0, \ldots, N-1 \tag{11}$$

where $w = e^{\frac{j2\pi}{N}}$, so that $w_{lN} = 1$, $l \in Z$. Employing the primary argument, or phase $a_k = Arg[u_k]$ we construct the phase vector. Consider now a second curve $z' = [z'_0 z'_1 \dots z'_{N-1}]^T$ that is circularly shifted with respect to z by m samples, where $m \in 0, 1, \dots, N-1$.

$$z' = S_m(z) = [z'_i = z_{(i+m) \mod N} | \ i = 0, 1, ...N-1] \tag{12}$$

In order to normalize the curve, a standard circular shift is defined using the first and last Fourier phases:

$$p(z) = [\frac{N}{4\pi}(\alpha_1 - \alpha_{N-1})] \ mod \ \frac{N}{2} \tag{13}$$

and the opposite shift is applied to normalize the curve:

$$n_p(z) = S_{-p(z)}(z) \tag{14}$$

It is shown in [2] that the above normalization is invariant to starting point. Rotation normalization is achieved by setting the phases of u_1 and u_{N-1} to zero, so that the became real and positive. Assume that two curves $\mathbf{C}$ and $\mathbf{C}'$ have been orthogonalized and normalized with respect to their starting point, thus satisfying eq. (8). We then uniquely decompose matrix Q as

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{15}$$

where $\theta \in [0, \pi)$, $s_x = \pm 1$ and $s_y = \pm 1$, in order to denote a one-to one relation between rotation/reflection parameters and elements of Q. Adopting the complex vector notation z, z',

$$z' = (s_x x + j s_y y) e^{j\theta} \tag{16}$$

The rotation curve z is normalized according to the average value of Fourier phases α_1 and α_{N-1}:

$$r(z) = [\frac{1}{2}(\alpha_1 + \alpha_{N-1})] \mod \pi \tag{17}$$

$$z_1 = z \cdot e^{jr(z)} \tag{18}$$

Horizontal and vertical reflection is normalized according to the third-order moments of z_1:

$$v(z_1) = v_x(z_1) + j v_y(z_1) = sgn[m_{12}(z_1)] + j \cdot sgn[m_{21}(z_1)] \tag{19}$$

$$n_r(z_1) = z_2 = v_x(z_1) x_1 + j \cdot v_y(z_1) y_1 \tag{20}$$

where $sgn[\cdot]$ denotes the signum function. It is then proved in eq. (3) that $n_r(\mathbf{C})$ is invariant to rotation and reflection transformations:

$$n_r(z') = n_r(z) \tag{21}$$

As in curve orthogonalization, the set of parameters $r(z), v_x(z), v_y(z)$ together with $n_r(z)$ contain all information about the original curve z. Combining all the above results, it is proved that the curve $n_r(n_p(n_a))$ obtained by the entire normalization procedure is invariant to any affine transformation.

4 Object Matching

Once visual descriptors have been extracted for each detected moving object, these are employed to perform matching with existing objects stored in a database with similarly computed visual descriptors. Matching functions are defined for this purpose, for each visual descriptor. In the following, the matching procedure is described for the color and shape descriptors defined in the previous section.

4.1 Color Matching

Matching of visual objects using color descriptors is based on mean color vectors and dominant colors. More specifically, for mean color vectors, we use the RGB information corresponding to the extracted moving objects of interest. The color values of the region defined by the estimated object contour are normalized in the interval $[0, 1]$, and the respective mean values $(\overline{r}, \overline{g}, \overline{b})$ are calculated. Thus, each extracted object is described by the *mean color vector* $m = [\overline{r}, \overline{g}, \overline{b}]$. The color matching criterion between two objects with mean color vectors m_i and m_j, respectively, is then,

$$D_{MC} = \|m_i - m_j\| = \sqrt{(\overline{r}_i - \overline{r}_j)^2 + (\overline{g}_i - \overline{g}_j)^2 + (\overline{b}_i - \overline{b}_j)^2} \tag{22}$$

This criterion is actually the mean square error between the two color descriptors m_i and m_j. D_{MC} is used in our implementation with adequate results, taking into account the mean color vector of the objects in one or more frames: for more accurate results, in case of external lighting changes along time, we calculate the mean value of the vector m in successive frames, and then calculate D_{MC}, according to that value.
For the dominant color descriptors, the matching function used depends on the components present in the query and target descriptors. The basic matching function D_{DC} between two objects i and j uses only the percentages and color values and is defined as follows

$$D_{DC} = \sum_{k=1}^{N_i} p_{ik}^2 + \sum_{l=1}^{N_j} p_{jl}^2 - \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} 2 a_{ik,jl} p_{ik} p_{jl}, \tag{23}$$

where p_i and p_j correspond to query and target descriptors, and $a_{ik,jl}$ is the similarity coefficient between two colors c_{ik} and c_{jl}:

$$a_{ik,jl} = \begin{cases} 1 - \frac{d_{ik,jl}}{d_{max}} & , d_{ik,jl} \leq T_d, \\ 0 & , d_{ik,jl} > T_d, \end{cases} \tag{24}$$

where $d_{ik,jl} = \|c_{ik} - c_{jl}\|$ is the Euclidean Distance between two colors c_{ik} and c_{jl}, T_d is the maximum distance between two colors considered as similar, and $d_{max} = \alpha T_d, \alpha > 1$. This distance can be modified to take into account the optional variance. One can then take a linear combination of the spatial coherency and the above distance to give a combined distance as suggested in [10].

4.2 Shape Matching

Once object contours have been normalized and are invariant to affine transforms, the most common way to measure the similarity between curves is the Euclidean distance. Another way to measure the similarity between curves s_i, s_j is the cross-correlation criterion, which is defined as

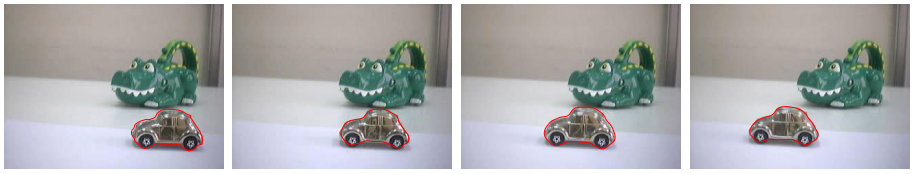
$$D_S = \rho(s_i, s_j) = \frac{\sum_{k=0}^{N-1} s_{ik} \cdot s_{jk}}{\sqrt{\sum_{k=0}^{N-1} s_{ik}^2} \cdot \sqrt{\sum_{k=0}^{N-1} s_{jk}^2}} \tag{25}$$

where s_{ik} is the k-th point of curve s_i. The cross-correlation is a normalized measure, which denotes how similar two curves are, and indicates a metric of their content similarity.

5 Experimental Results

In this section we verify the efficiency of the proposed integrated scheme shown in Fig. 1, in two video sequences representing three cases of object recognition. In the first sequence a silver car is in motion, and it is successfully extracted following the method presented in Section 2. In the second sequence two vehicles are in motion and thus tracked: a car of the same shape with the one extracted in the first sequence, but of different color (green), and a truck (different shape) of the same color with the car of this sequence. Thus, we are called upon to reach the three following conclusions: (a) the proposed scheme performs very well even when the object contours are extracted with variations from the ground-truth (actual contours), or when their shape is deformed due to the projection of their motion; to verify this assumption we use the same object in different (non-successive) frames of the same sequence, (b) the two cars of the two sequences are of the same type but they are not of the same color, and (c) in the second sequence, the two moving objects are different in terms of shape, and thus there is no need to proceed in color matching to decide whether they are *similar*. Since the integrated scheme provides efficiency in the above three cases, the authors are currently working on the construction of an object database and a low-complexity searching procedure in that database.

Fig. 4 illustrates the performance of the tracking method described in Section 2, where the moving object of interest is the silver car. The object's contour is extracted in four non-successive frames, and it is used for both the shape and the color matching procedures. In this example, the efficiency of the proposed matching algorithm is verified in two frames (Fig. 4(a),(d)) of the sequence. The contour of the car consisting of 100 sample points is illustrated in Fig. 5 for each frame. The algorithm's efficiency is based on the affine transformations, following the proposed normalization steps described in Section 3.2, as shown in Fig. 6. It can been seen that the final curves match very well, although normalization of each curve is performed without the knowledge of the other. The cross-correlation between these two curves is $\rho = 0.9995$ ($\simeq 1$), which indicates



(a) (b) (c) (d)

Fig. 4. Tracking example in four frames of a sequence.

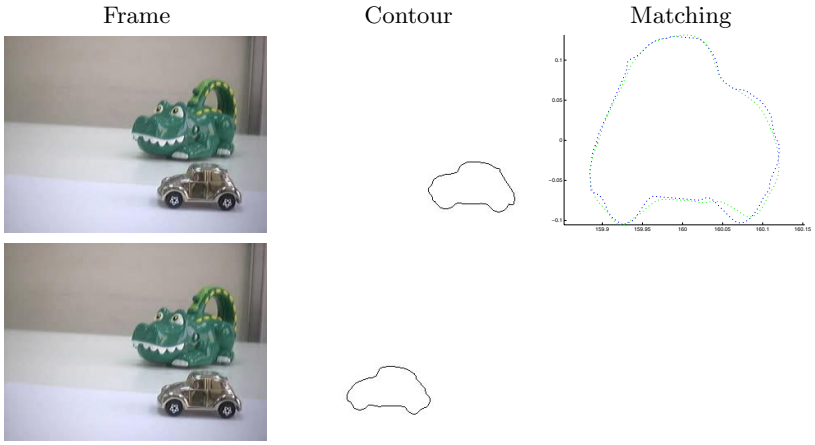
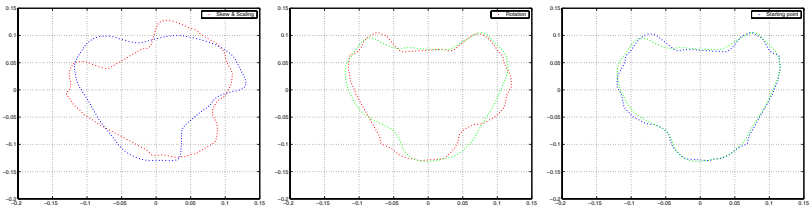


Fig. 5. Affine invariant contours obtained for the same object in two different instances



(a) (b) (c)

Fig. 6. (a) Curves after scaling normalization, (b) curves after rotation, and (c) starting point normalization.

that these two contours very similar.

In the next example, illustrated in Fig. 7, two sequences containing objects of different colors and with similar contours are presented. The respective tracking results are shown in Figs. 3 and 4. The contour transformations, proposed in Section 3.2, result in similar contours as shown in Fig. 7, which indicates that the two cars are of the same type. This is also concluded numerically, using the cross-correlation between the contours of the silver and the green car, which in this case is $\rho = 0.9988$; the value of that measure is close to 1, which indicates

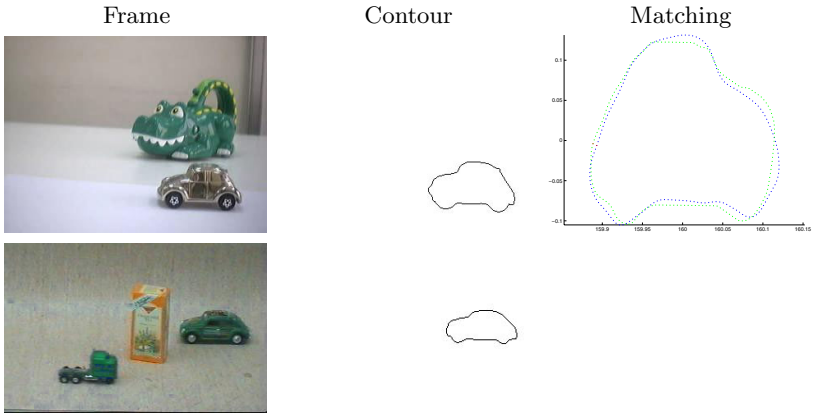


Fig. 7. Sequences which contain objects with the same shape but different color

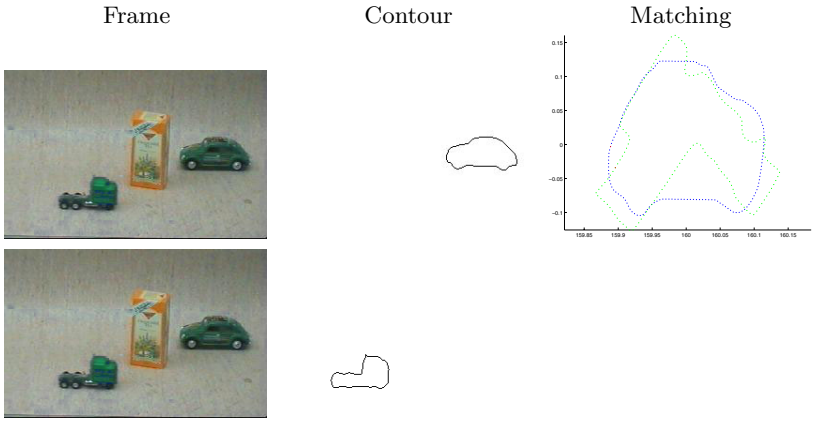
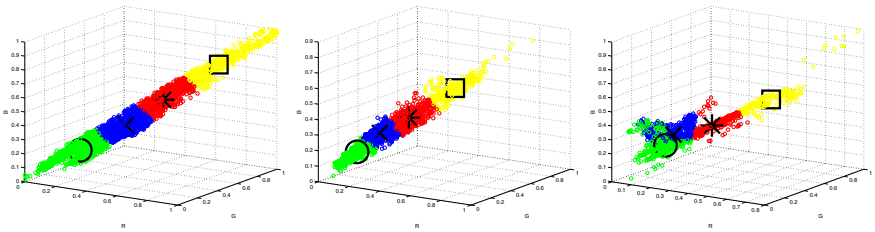


Fig. 8. Sequences with different objects of the same color.

that these two cars are of the same type.

In the final example, two objects with different shape are examined, whose dominant colors are similar, as shown in Fig. 8: green car and green truck extracted in Fig. 3). In such cases, depending on the application, we conclude either that there is no need to proceed to color matching, since the two shapes (and consequently the two objects) are quite different, or that their dominant colors are similar (if we are interested in objects of the same color). The contour normalization results, illustrated in Fig. 8, show that the two contours are quite different, whereas the cross-correlation between these two contours is $\rho = 0.6586$.

Fig. 9 illustrates the color clustering results for the three objects examined: (a) silver car, (b) green car and (c) green truck. For each object, four color clusters are estimated along with the respective centers. It must be noted that the colors shown in the $3D$ graphs do not represent the true colors corresponding to the



(a) (b) (c)

Fig. 9. Color distributions for the three objects, in the RGB space, after clustering: four color clusters for each object have been estimated, whereas the center of each cluster is also illustrated.

objects	D	D_{DC}
silver car - green car	0.3692	0.2564
silver car - green truck	0.3430	0.2572
green car - green truck	0.0520	0.0447

Table 1. Color matching results (eq. 22) for the three moving objects of the examples illustrated in Figs. 3 and 4.

clusters, but are used for representation purposes.

Finally, for the three extracted objects of the previously described examples, the color matching results are illustrated in Table 1. As can be seen in the last row, two of the objects (a car and the truck) are similar in terms of color ($D_{MC} \simeq 0.05$, $D_{DC} \simeq 0.04$), whereas the matching between the silver car and the other two objects leads to values of $D_{MC} > 0.3$ and $D_{DC} > 0.2$.

6 Conclusions and Further Work

In this paper an integrated scheme for moving object extraction and recognition is proposed, aiming at the detection of objects of specific shape (contour) and color. In this direction, three different methods of the literature are revised, extended and integrated together: (a) moving object tracking, (b) contour affine-invariant normalization, and (c) dominant color extraction. After following these three steps, we decide on the similarity between two (or more) objects, according to appropriate criteria. In this work, we test the proposed integrated scheme in three simple examples, where the ground-truth is available; this is mainly done to verify our assumptions. We are currently working on extending this scheme, using an appropriate database of real-world sequences, for object-based video retrieval.

References

1. G. Akrivas, M. Wallace, G. Andreou, G. Stamou and S. Kollias, "Context-Sensitive Semantic Query Expansion," *IEEE International Conference Artificial Intelligence Systems AIS-02*. (to appear).
2. Y. Avrithis, Y.Xirouhakis, S. Kollias, "Affine-invariant curve normalization for object shape representation, classification, and retrieval," *Machine Vision and Applications*, 13, pp. 80-94, 2001.
3. J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
4. M.J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *CVIU*, 63(1), pp. 75-104, 1996.
5. S. Haykin, Neural Networks, Macmillan College Publishing Company, Chapt. 5, pp. 124-126, 1994.
6. P. Huber, Robust Statistics, Wiley eds., NY, 1981.
7. H. S. Ip and S. Dinggang, "An Affine-Invariant Active Contour Model (AI-Snake) for Model-Based Segmentation," *Image and Vision Computing*, 16(2), pp. 135-146, 1998.
8. M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *Int. Journal of Comp. Vis.*, 1(4), pp. 321-331, 1988.
9. V.K Madisetti and D.B Williams Eds., The Digital Signal Processing Handbook, CRC Press, 1998, Chapt. 74, pp. 20-26.
10. ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC 15938-3/FCD, *Information Technology Multimedia Content Description Interface Part 3 Visual*, October, 2001.
11. G. Tsechpenakis, Y. Xirouhakis and A. Delopoulos, "A Multiresolution Approach for Main Mobile Object Localization in Video Sequences," *International Workshop on Very Low Bitrate Video Coding (VLBV01)*, Athens, Greece, October 2001.
12. G. Tsechpenakis, N. Tsapatsoulis and S. Kollias, "Probabilistic Boundary-Based Contour Tracking with Snakes in Natural Cluttered Video Sequences," *Int. Journal of Image and Graphics (IJIG)*, accepted, 2003. http://www.image.ece.ntua.gr/∼gtsech/IJIG-DM-10.pdf