Segmentation and Tracking of Objects in Stereoscopic Video Sequences Using Extended MPEG-7 Features

Daniel Socek*a, Dubravko Ćulibrk* b, Oge Marques*a, Borko Furht*a

aDepartment of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA.
bFaculty of Technical Sciences, 21000 Novi Sad, Serbia.

Abstract
The paper presents a system for automatic video surveillance based on paired cameras in a stereoscopic setup. The system combines motion based object segmentation with tracking, and integrates depth information to achieve robust performance. The methodology encompasses object segmentation based on a class of probabilistic neural networks, shadow removal, computation of a depth map, and object tracking based on an extended set of MPEG-7-like feature descriptors including intensity, color, shape, motion and depth. To evaluate the approach, experiments were conducted on a set of outdoor sequences containing both rigid objects and moving persons. The results presented in the paper indicate that the proposed approach is able to achieve accurate segmentation and tracking and handle occlusions efficiently.

Keywords: Stereo vision, surveillance, object segmentation, tracking, MPEG7 features.

1. Introduction
A fundamental challenge facing the design of an automated system for surveillance is ensuring that it is capable of fast and accurate object detection and tracking [1]. The effectiveness of object identification and behavioral analysis depends on the quality of tracking data it receives, which in turn
depends on the quality of detected objects. An object that is never detected cannot be tracked nor identified. Likewise, a falsely detected object or incorrectly tracked trajectory wastes processing resources and reduces the performance of identification and interpretation modules.

This paper presents a new approach to object detection and object tracking in the context of an automated digital video surveillance system, employing a subset of MPEG-7 object descriptors and stereo visual data for tracking. The methods presented here are a result of a research project performed at the Center for Coastline Security Technologies at Florida Atlantic University. The work falls within the context of designing a complete system for video surveillance along the lines of VSAM[2], W4[3] and KNIGHT [4], to mention just a few. The ultimate objective is the design of a detection and tracking module that is robust to object shape, complex or moving backgrounds, and full and partial object occlusion, which is also fast and suitable for real-time monitoring and surveillance applications. The input to the module is stereo video acquired by a pair of appropriately mounted static cameras.

This paper describes an approach based on Bayesian background modeling neural networks used for robust moving object detection for each camera separately. It extends the segmentation approach proposed by Culibrk et al.[5] to include novel shadow removal methodology, based on iterative classification of the pixels on the outer rim of detected objects. To compute the depth information in real time, a novel fast method of rough object-level depth estimation from stereo video sequences is proposed. The resulting depth information is used to improve the handling of object occlusions and can be used to learn relative distances between objects (given an a priori reference). A feature-based object tracker has been designed to track the objects by matching a subset of their MPEG-7 descriptors (color, shape, motion and texture) complemented with depth information. Extracted features for objects in the current frame are compared to previously extracted features (for the same objects as well as other objects) and adapted according to measures of their discriminative power. The tracker does not incorporate any prediction. Fig. 1 shows the proposed framework.

The rest of this paper is structured as follows. Section 2 presents an overview of the related published work. Section 3 provides background information on object detection (segmentation), whereas Section 4 describes the method used by the system to estimate objects’ depth from stereo sequences. Object features used and the pertinent comparison criteria are described in Section 5, while the tracking approach based on the comparison of extracted
features is given in Section 6. Experiments and results appear in Section 7. Section 8 holds the conclusions.

2. Related Work

2.1. Motion-Based Segmentation

Moving object segmentation, in sequences taken by a stationary camera, is usually achieved by building a model of the background in a chosen feature space and comparing the current frame values to the model [1]. Early models used for motion(change)-based video object segmentation were based on smoothing the color of a background pixel over time using different low-pass filtering techniques, to create a reference background frame. The reference frame was constantly updated and used to segment the foreground objects by subtracting (differencing) it from the current frame of the input sequence. Such methods are based on the assumption that changes in the background
are slower than those of the objects to be segmented, and as such is not effective for sequences with high-frequency background changes, nor is it able to discern between moving background and foreground objects. Rosin [6] provides a good overview of these early results.

Better segmentation can be achieved by utilizing statistical background models, such as a Gaussian-based background model, where the values of each background pixel are modelled by a unimodal probability density function (PDF), whose parameters are recursively updated to follow gradual background changes within the video sequence[7]. Furthermore, the model can be significantly improved by employing a Mixture of Gaussians (MoG), in which the values of the pixels from background objects are described by multiple Gaussian distributions[8, 9, 10]. While MoG remains one of the most widely used foreground segmentation approaches, its performance is limited by underlying assumptions about the shape of the PDFs it is trying to estimate. Such models are commonly referred to as parametric.

More recently, several approaches based on nonparametric models have been proposed [11][12][5]. In 2004, Li et al. proposed a method for foreground object segmentation (background subtraction) employing a Bayes decision framework where no a priori assumptions about the scene are necessary [11]. The method has shown effective even for the sequences containing complex variations and non-periodical movements in the background. It uses binning of the features where a single probability is assigned to each bin, leading to a discrete representation of PDFs, equivalent to a square kernel kernel-density (Parzen) estimate [13]. The work of Sheikh and Shah [12] relies on a Gaussian kernel estimator for the pixel values. Like the methodology proposed here, their approach employs a shadow removal mechanism to enhance segmentation, using the disparity between luminance and chromaticity information.

In 2007, a foreground segmentation approach based on Background Modeling Neural Networks (BNNs) was proposed by Culibrk et al. [5]. The networks employ a biologically plausible implementation of Bayesian classifiers and nonparametric kernel based density estimators. Results superior to those of Li et al. and MoG with 30 Gaussians were reported. Although nonparametric models impose no constraints on the shape of the estimated probability density function, they are generally computationally expensive. The BNNs address the problem of computational complexity of the kernel based background models by exploiting the parallelism of neural networks. The BNN foreground segmentation forms the initial step of the proposed approach. The segmentation is enhanced with a shadow removal module,
that has not been used in the original segmentation approach. The resulting approach produces significantly better segmentation results in the presence of shadows.

While most approaches exploit spatial correlation of pixel values at the morphological post-processing stage only and treat the sequence as a set of independent pixel processes for background modelling purposes [9][11][5], some recently proposed methods attempt to exploit the spatial correlation better. Carminati and Benois-Pineau [14] use regularization based on a Markov Random Field to provide spatio-temporal coherence of detection, while Culibrk et al. [15] use texture descriptors extracted from pixel neighborhoods with stable texture to enhance the segmentation.

2.2. Shadow Removal

Since shadow moves with the object, it is usually detected as part of the object in foreground segmentation. As Prati et al. concluded in a recent survey [16], there is no generally accepted method for removing shadows from a segmented video object. Instead, different approaches to shadow detection and removal should be taken when addressing different kinds of applications. Several of the recently proposed approaches attempt to remove the shadows from the segmented video objects by eliminating the intensity values and exploiting the chromatic similarity between the shadow and the background image. Different color spaces have been used for this purpose, including YCbCr [17], YUV [18], HSV [19] and HSL [20]. An HSL-based method by Grest et al. from [20] has been adopted into several recent proposals for video object segmentation tracking [21][22], demonstrating good performance. In our system, an iterative contour-based variant of the scheme by Grest et al. has been used.

2.3. Depth Estimation

Depth information have been used in several tracking systems in the past. For instance, Rehg et al. [23] used it to extract additional information about the position of the objects, needed by their smart kiosk application. This was done after the segmentation and tracking. Bae et al. [24] proposed a stereo object tracking method based on the disparity motion vectors (DMVs). In the tracking system by Bae et al., a block-based matching algorithm is employed to compute DMVs which are then used to both detect and track moving objects. In the work by Dang et al. [25], the authors propose fusing the disparity map and optical flow with extended Kalman filters to both detect
and track moving vehicles. The optical flow and disparity map are derived by computing the block-based motion fields. Recently [26], depth has been used to improve the quality of initial object detection for the purpose of multiple person tracking in sequences taken by a pair of moving stereo cameras. The stereo cues are used to reduce the number of false positives created by the initial object detection performed by pedestrian recognition approaches operating on a single frame, rather than a foreground segmentation algorithm. The tracking itself is performed by an extended hypothesize-and-test model selection framework, relying on a graphical model of the scene. Unlike in the previously proposed stereo-based tracking schemes, within the work presented, depth information is used as an additional object feature which aids object tracking. The objects are detected using a powerful neural network based object detector and tracking is accomplished by a classification of detected objects in each temporal unit using complex set of local and global features.

The most recent algorithms that reportedly achieve the closest depth estimation to the ground truth depth are based on computationally complex procedures involving over-segmentation of color images [27] and [28], making them unsuitable for the purpose at hand. For example, an approach from [27] reportedly takes between 14 and 25 seconds on 2.1GHz general-purpose 64-bit processor to estimate depth from a pair of CIF stereo images. For the ranking of many algorithms in terms of accuracy, the reader is referred to [29] as well as the constantly updated Middlebury stereo web page http://vision.middlebury.edu/stereo. Faster algorithms achieve a less accurate estimation, but are the sole option when real time performance in video processing is desired.

In [30], Birchfield and Tomasi proposed an approach to estimating disparity map from stereo images using comparisons of pixel values in corresponding scanlines in left and right images. The approach is considered extremely fast, but a typical PC implementation of Birchfield-Tomasi algorithm still takes a few seconds for a stereo pair of CIF images with a smaller value selection of maximum disparity search parameter \(d_{\text{max}}\) (e.g. 20 or less). Fortunately, solving the general problem of stereo correspondence is not necessary in the proposed system, since it does not need to perform stereo matching on the background. To achieve real time performance, the problem is restricted to finding the depth of a segmented object as a whole, which is computationally much simpler problem and could be solved in a straightforward region shift matching that is more than capable of achieving depth estimation in
real-time.

2.4. Tracking for Visual Surveillance

When the general problem of tracking moving objects is concerned, targets are typically followed using classic tracking approaches, which model the characteristics of object motion, such as Extended Kalman Filters (EKF)[31] and particle filters[32]. These methods rely on a first-order Markov assumption and hence carry the danger of drifting away from the correct target. This has subsequently been improved upon by optimizing data assignment and considering information over several time steps, as in Multi-Hypothesis Tracking (MHT) [33][31] and Joint Probabilistic Data Association Filters (JPDAF)[34]. Unfortunately, the combinatorial nature limits those approaches to consider either only few time steps [33] or only single trajectories over longer time windows [35], [36]. Recently, Ess et al.[26] proposed a hypothesize-and-test model selection framework that works online and simultaneously optimizes detection and trajectory estimation for multiple interacting objects and over long time windows, intended for tracking objects in sequences taken by moving stereo cameras.

When automatic surveillance is concerned, however, the preferred tracking approach is tracking by matching object features [4]. Among the earlier automated surveillance systems, Pfinger [37] is perhaps the best known. It tracks the full body of a person in the scene that contains only one unoccluded person in the upright posture. It uses a unimodal background model to locate the moving person. In Rehg et al. [23], a smart kiosk is proposed that can detect and track moving people in front of a kiosk by using face detection, color, and stereo. The adaptive MoG background subtraction method of Stauffer and Grimson [9] is able to deal with slow changes in illumination, repeated motion from background clutter, and long-term scene changes. Their system tracked segmented objects using a multiple hypothesis tracker. When object detection has been performed, mean-shift tracking proposed by Comaniciu et al.[38] is an effective way of tracking objects using their color histogram.

Ricquebourg and Bouthemy [39] proposed tracking people by exploiting spatiotemporal slices. Their detection scheme involves the combined use of intensity, temporal differences between three successive images, and comparing the current image to a background reference image, which is reconstructed and updated online. Boult et al. presented a system for monitoring uncooperative and camouflaged targets[7]. The W^4 uses dynamic appearance models
to track people [40]. Single persons and groups are distinguished using projection histograms, and each person in a group is tracked by tracking the head of that person.

In 2007, Shah et al. [4] described a system called “Knight”, that models an object using a combination of its color, shape, and motion models. A Gaussian distribution represents the spatial model. The color model is a probability density function (PDF) that a normalized histogram approximates. Each pixel in the foreground region votes for an objects label, for which the product of color and spatial probability is the highest. Each region in the current frame is assigned a label if the number of votes from the regions pixels for the object is a significant percentage—say $T_p$—of all the pixels belonging to that object in the last frame. If two or more objects receive votes greater than $T_p$ from a region, it is assumed that multiple objects are undergoing occlusion. The position of a partially occluded object is computed by the mean and variance of pixels that voted for that particular object. In case of complete occlusion, a linear velocity predictor is used to update the occluded objects position. The spatial and color models are updated for objects that are not undergoing occlusion.

Although the nature of surveillance applications requires that extracted features to have small computational cost, small storage requirements, and be invariant to translation, rotation and scaling, MPEG-7 descriptors are not typically used for tracking. Murthy et al. [41] considered a rudimentary semi-automatic segmentation and tracking approach based on MPEG-7 color layout and texture descriptors. They proposed using a graph-cut algorithm to achieve foreground segmentation and showed initial tracking results where thresholds for object feature comparison are set by hand. The approach of Murthy et al. did not consider occlusions, scenes cluttered with objects that have similar features and other hard tracking situations. The approach proposed herein relies on a much larger set of features to achieve reliable tracking. Distance measures are formulated for each of the features and majority voting employed to track objects. The features are continuously updated to create a stable object template that can be used for tracking. This is especially important for recovering the object track after occlusion.

3. Object Detection through Foreground Segmentation

Without robust detection of video objects, subsequent actions, such as object tracking and classification, would be infeasible. In the proposed sys-
tem, object detection is accomplished through moving object segmentation, as is typically done when surveillance from stationary cameras is concerned [4]. In such applications, segmentation should be able to overcome obstacles inferred by the presence of complex, moving background, which often occurs in the outdoor surveillance footage.

3.1. Object Detection with Background Modeling Neural Networks

A BNN [5] is a neural network designed to serve both as a statistical model of the background at each pixel position in the video sequences and as a highly-parallelized background-subtraction algorithm. The network is an unsupervised learner. It collects statistics related to the dynamic processes of pixel feature values changes. The learned statistics are used to classify a pixel either as foreground or background in each frame of the sequence.

Probabilistic motion (change) based background subtraction methods rely on the following supposition: pixel feature values corresponding to back-
ground objects will occur most of the time, i.e. more often than those pertinent to the foreground. Thus, if a classifier is able to effectively distinguish between the values occurring more frequently than others it should be able to achieve accurate segmentation. In a BNN the segmentation problem is formulated to enable the use of Bayes decision rule to achieve segmentation as follows. For a certain frame $t$, one is trying to estimate the dependent variable ($\Theta_i \in \{f, b\}$). The event of pixel at location $i = (x_i, y_i)$ being part of the foreground corresponds to $\Theta_i = f$, while $\Theta_i = b$ when the pixel is pertinent to background. A Bayesian decision rule for simple pixel classification is formulated as:

$$
\Theta = \begin{cases} 
    f, & \text{if } \pi_{bi} P_{bi}(v) < \pi_{fi} P_{fi}(v); \\
    b, & \text{otherwise.}
\end{cases}
$$

(1)

where $P_{bi}(v)$ is the PDF of background occurring at pixel $i$, $P_{fi}(v)$ is the PDF of foreground occurring at pixel $i$, $\pi_{fi}$ and $\pi_{bi}$ are prior probabilities of foreground and background occurring.

Fig. 3 shows the structure of a BNN. The central part of the BNN is the classification subnet which contains four layers of neurons. Input neurons of this network simply map the inputs of the network, which are the values of the features for a specific pixel. Each input neuron is connected to all pattern neurons. The output of the pattern neurons is a nonlinear function of Euclidean distance between the input of the network and the stored pattern for that specific neuron. The only parameter of this subnet is the smoothing
parameter of the Parzen estimator [13] used for estimating the PDFs $P_{bi}$ and $P_{fi}$. The output of a single pattern neuron corresponds to the value of a single gaussian of the PDF estimation for the observed pixel value. The output of the summation units of the classification subnet is the sum of their inputs. The subnet has two *summation neurons*, each of them connected to all pattern neurons. The output values of the summation neurons correspond to initial Parzen estimates of $P_{bi}$ and $P_{fi}$ for the pixel value observed. These estimates are input to the last layer, containing a single *output neuron*. The final output of the network is a binary value indicating whether the pixel corresponds to foreground or background.

In the classification subnet of BNN, the weights between the pattern and summation neurons are used to store the the prior probabilities inferred for the pattern neuron value ($\pi_{fi}$ and $\pi_{bi}$). Since these values are unknown, rules were formed which allow the BNN to estimate them based on the observed parts of a pixel process and the frequency of specific feature values observed. The weights of these connections are updated with each new value of a pixel at a certain position received (i.e. with each frame), according to the following recursive equations:

\[
W_{ib}^{t+1} = f_c((1 - \frac{\beta}{N_{pn}}) \cdot W_{ib}^t + MA^t \beta) \\
W_{if}^{t+1} = 1 - W_{ib}^{t+1}
\]

where $W_{ib}^t$ is the value of the weight between the $i$-th pattern neuron and the background summation neuron at time $t$, $W_{if}^t$ is the value of the weight between the $i$-th pattern neuron and the foreground summation neuron at time $t$, $\beta$ is the learning rate, $N_{pn}$ is the number of the pattern neurons of BNN, $f_c$ is a clipping function defined by (4) and $MA^t$ indicates the neuron with the maximum response (activation potential) at frame $t$, according to (5).

\[
f_c(x) = \begin{cases} 
1, & x > 1 \\
x, & x \leq 1
\end{cases}
\]

\[
MA^t = \begin{cases} 
1, & \text{for neuron with maximum response;} \\
0, & \text{otherwise.}
\end{cases}
\]

To form a complete background-subtraction solution a single instance of a BNN is used to model the features at each pixel of the image. The features used in our model are RGB color values, as depicted in Fig. 2.
In a hardware-implementation, the segmentation speed of the BNN approach, due to the parallel operation of neural networks, does not depend on the size of the frame. The delay of the network (segmentation time) corresponds to the time needed by the signal to propagate through the network and time required to update it. In a typical FPGA implementation this can be done in less than 20 clock cycles, which corresponds to a 2ms delay through the network, for a FPGA core running at 100ns clock rate. Thus, the networks themselves are capable of achieving a throughput of some 500 fps, which is more than sufficient for real-time segmentation of video sequences.

The core design of a BNN is provided in [5]. The basic BNN approach has recently been improved in [42] by allowing an automatic selection of the width of kernels used to estimate the PDFs, thus making the entire BNN object segmentation process dependant on a single control parameter.

3.2. Removal of Shadows

Shadows associated with moving foreground video objects are typically segmented along with the objects. The presence of shadows negatively affects the accuracy of feature extraction phase as it adds significant noise to the shape features, but also to color and texture features. Since these appearance features are used for object tracking in our system, a method for removing shadows from segmented objects is employed to enhance the initial segmentation results.

Grest et al. [20] proposed a shadow removal method which uses HSL color space to split the color information from the brightness values in the visual object. Shadow removal based on this method have been applied to several recent proposals for object segmentation and tracking, including [21] and [22]. In this paper we also adopt a variant of method by Grest et al.

In [20], pixel values that are similar in Hue and Saturation are considered pertinent to the shadow. This method assumes the following properties: (1) a shadow pixel is darker than the corresponding pixel in the background image, (2) the texture of the shadow is correlated with the corresponding texture of the background image. To limit the effects of falsely classified shadow pixels to the exterior of the segmented object, we chose to apply the shadow removal iteratively to the inner contour of the object boundary.

The modified HSL-based shadow removal approach from [20] is summarized as follows:

1. Convert color pixels of an object $O$ to HSL color space
2. Obtain contour of $O$, denoted $C(O)$

3. Trace contour $C(O)$ and remove all pixels in $O$ for which Hue and Saturation values are close to the corresponding Hue and Saturation values in the background image estimated in the BNN segmentation process according to the distance criteria from [20].

4. Repeat process from step 2 until no pixels are deleted from the contour.

The proposed contour-based approach is illustrated in Fig. 4. Internal contour with 4-connected pixel neighborhood is used. In each pass of the algorithm the shadow is gradually removed from the object without affecting its interior region. Additional experimental results related to shadow removal are depicted in Section 7. The reader is referred to web page http://www.cse.fau.edu/~oge/research.html where for a number of test sequences all detected objects before and after shadow removal can be downloaded.

Figure 5: Example of shadow detection and removal from segmented objects (grey area represents removed shadow).
4. Object Depth Estimation from Stereo

Depth information regarding tracked video objects represents a powerful feature capable of distinguishing objects with similar color, shape and texture features. Depth can be estimated from a pair of stereo frames using a stereo correspondence algorithm.

Given a pair of stereo images, the correspondence problem refers to determining the match sequence for each corresponding scanline in a pair of stereo images. The match refers to an ordered pair \((x, y)\), where \(x\) and \(y\) are the positions in same scanlines of left and right stereo pair, respectively, such that the pixel values corresponding to these positions, \(F^L(x)\) and \(F^R(y)\), represent images of the same scene point. Here, it is assumed that the stereo images are properly aligned so that the scanlines are the epipolar lines. Unmatched pixels are labeled as being occluded and adjacent occluded pixels which are bounded by non-occluded pixels are referred to as an occlusion.

The disparity of a pixel position \(x\) in the left scanline that matches the pixel \(y\) in the right scanline is defined as the difference \(x - y\), while the disparities of the pixels in an occlusion are assigned the farther of the two bounding regions. Approaches to the stereo correspondence problem construct the so called disparity map, which is also known as the depth map or the depth estimation, since it described the discrete estimation of the third spatial dimension.

Figure 7 demonstrates the configuration we used to capture stereo images. Two cameras “left” and “right” were fixed and pointed at the same scene. The cameras were placed approximately ten centimeters apart on the same, level horizontal plane, creating a slight disparity between the image captured by the left camera and the corresponding image from the right camera. This disparity was then used to estimate the depth of objects in the captured scene.

The general problem of stereo correspondence brings an unnecessary computational load when the proposed system is concerned, since there is no need to perform stereo matching on the background. This process is depicted in Figure 7a. To achieve real-time performance, depth estimation is restricted to the problem of finding the depth of a segmented object as a whole, which is computationally much simpler problem which can be solved in a straightforward object shift matching that is more than capable of achieving depth estimation in required time.

The disparity of an object is determined by the best match of an object
Figure 6: Process of estimating depth of the entire video object using disparity measure from left and right stereo frames (top two figures). The best correspondence (scanline-wise) between the object in the left and right frame is taken as the best estimate of the disparity of the object. In this example, Object 2 is estimated to be closer to the camera than Object 1 since \( d_2 > d_1 \).

in the left stereo frame with a horizontal search in the right stereo frame. The horizontal shift which produces the smallest SAD (sum of absolute differences) is declared the best match, and the amount of shift is used to approximate the disparity. This approach is similar to a full-pixel block-based motion search in video coding, except that the search is object-based and restricted to a horizontal axis. The horizontal search range is bounded by the maximal disparity range \( d_{\text{max}} \).

Let \( F^L \) and \( F^R \) denote the \( n \)th frame in the synchronized sequences captured with left and right camera, respectively. The disparity \( d \) of an object \( O \in F^L \) produced by a BNN segmentation of the left frame is determined as follows:

\[
d(O) = \min_{0 \leq r \leq d_{\text{max}}} \left( \sum_{j=0}^{bh} \sum_{k=0}^{bw} s_{i,j,k} \right),
\]

\[
s_{i,j,k} = \begin{cases} 
|B(O)_{k,j} - F^R_{bx+k-1,by+j}|, & \text{if } M(O)_{k,j} = 1; \\
0, & \text{otherwise},
\end{cases}
\]

where \( |\cdot| \) denotes the absolute value, \( M(O) \) denotes the binary shape mask of
Figure 7: Depth estimation with: (a) general stereo correspondence method (in this instance Birchfield-Tomasi algorithm from [30]), and (b) the proposed object-based stereo correspondence method. The running time of (a) is in seconds while the running-time of (b) is in milliseconds on a standard PC.

$O$, $B(O)$ the bounding box of $O$, while $bw$, $bh$ and $(bx, by)$ denote their width, height, and the coordinates of their top-left corner within $F^L$, respectively. $d_{max}$ denotes the maximal disparity (search range).

This process allows the system to use depth as one of the strong discriminatory features needed for successful tracking of detected objects. The proposed depth estimation method is depicted in Figure 7b. In comparison to the standard depth estimation methods which are applied on the entire stereo picture, the proposed method is significantly faster, hence allowing for a performance needed in surveillance applications.
5. Features for Tracking

In order to successfully track video objects across multiple frames, a set of representative features is extracted and compared with previously encountered objects. There are two types of features extracted in the proposed system: (1) locality features, such as depth, size and position, and (2) global appearance features, such as color, shape and texture. The locality features include all features of an object that are dependant on object location properties and which are expected to undergo changes during tracking. These features are likely susceptible to 3D isometric transformations including rotation, translation and scaling. On the other hand, global appearance features are deemed likely to remain constant during tracking, and are robust to isometric transformations (to a certain degree). While shape is arguably not invariant over the tracking time for non-rigid objects, it is likely to undergo periodic changes, so that its “invariance” is captured at certain time intervals.

Due to the nature of surveillance applications, the extracted features are desired to have small computational cost, small storage requirements, and, in case of appearance features, be invariant to translation, rotation and scaling. Since most of the MPEG-7 descriptors corresponding to object’s global appearance adhere to these requirements, a number of features used in the proposed system are either entirely or partially based on these standard descriptors. While similarity metrics for descriptors are not defined in the MPEG-7 standard, several commonly used distance metrics were adopted which have been proven to be effective in the experiments performed. In addition to descriptors provided by MPEG-7 and depth feature provided by the depth estimation algorithm, a few additional easily computable locality features are used to aid the tracking.

5.1. Locality Features

The size of a detected object is characterized by its bounding box size. The bounding box size is represented as a two dimensional vector \((\Phi_{bw}, \Phi_{bh})\) where \(\Phi_{bw}\) denotes the width and \(\Phi_{bh}\) the height of object’s bounding box. The position of object \(O\) is represented by the centroid of its polygonal approximation with \(n\) equidistant points, \((x_i, y_i), \ i = 0, \ldots, n - 1\), randomly selected from its contour \(C(O)\). The centroid, also known as the center of gravity or the center of mass, is given as a point \((\Phi_{cx}, \Phi_{cy})\) as follows:
\[
\Phi_{cx} = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),
\]
\[
\Phi_{cy} = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i),
\]
\[
A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i),
\]

where \((x_n, y_n) = (x_0, y_0)\) to create an imaginary closed polygon with \(A\) representing its area. \(n\) was set to 10 in the experiments performed, yielding centroids of decagonal approximations of objects’ shapes, as illustrated in Figure 8.

Depth estimation algorithm described in section 4 produces an estimation of depth for each object in a frame. Object depth, denoted by \(\Phi_d\), is represented as the disparity value calculated for an entire object. The value of \(d_{max}\) was set to 31, thus allowing 32 depth planes and limiting the depth feature size to 5 bits.

The distance between two locality features is measured using the standard Euclidean distance \(|| \cdot ||\) leading to the following metrics used to measure the distance of locality features of objects \(O_1\) and \(O_2\):

\[
D_i = ||\Phi_i(O_1) - \Phi_i(O_2)||,
\]

where \(i \in \{bw, bh, cx, cy, d\}\).
5.2. Global Appearance Features

5.2.1. Color Features

The MPEG-7 descriptor specifies a set of dominant colors in an arbitrarily shaped region (object) [43]. The following subset of feature components has been selected from the actual MPEG7 dominant color descriptor, to be used within the system proposed: number of dominant colors (up to 8), indices of dominant colors in RGB color space quantized to 3 bits per channel, and percentages of dominant colors.

The distance metric $D_c$ used for the color descriptor is based on a quadratic histogram distance measure (QHDM) from [44]. If a color feature $\Phi_c(O_j)$ of a video object $O_j$ is denoted by its size $n_j$, three dimensional indices of dominant colors $c_{ji}$ and corresponding percentages $p_{ji}$, then the distance metric between two features $\Phi_c(O_1)$ and $\Phi_c(O_2)$ is defined as:

$$D_c = \sum_{i=1}^{n_1} p_{1i}^2 + \sum_{j=1}^{n_2} p_{2j}^2 - \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} 2a_{ij}p_{1i}p_{2j},$$

(12)

where $a_{ij}$ is the similarity coefficient between colors $c_{1i}$ and $c_{2j}$. The similarity coefficient is defined as:

$$a_{ij} = \begin{cases} 1 - \|c_{1i} - c_{2j}\|/\alpha T_d, & \|c_{1i} - c_{2j}\| \leq T_d; \\ 0, & \|c_{1i} - c_{2j}\| > T_d, \end{cases}$$

(13)

where $\| \cdot \|$ denotes standard Euclidean norm and $T_d$ the threshold for maximum distance allowed for similar colors, with control rate parameter $\alpha$. For the proposed dominant color descriptor 0.5 and 0.05 for $T_d$ and $\alpha$ were used, respectively.

5.2.2. Shape Features

The MPEG-7 standard provides three shape descriptors: region shape descriptor, contour shape descriptor and 3-D shape descriptor. 3-D shape descriptor is not used in the system proposed since it deals with 2-D projections of the real-world captured by fixed stereo, cameras where 3-D shapes are unknown. The system utilizes both region and contour shape descriptors since they provide mutually exclusive concise discriminatory shape information about the object.

The region shape descriptor uses a complex 2-D Angular Radial Transform (ART) coefficients defined on a unit disk in polar coordinates as:
\begin{equation}
F_{nm} = \int_{0}^{2\pi} \int_{0}^{1} V_{nm}(\rho, \theta), f(\rho, \theta) \rho \, d\rho d\theta,
\end{equation}

where \( f(\rho, \theta) \) is an image function in polar coordinates, and \( V_{nm}(\rho, \theta) \) is the ART basis function separable along the angular and radial directions:

\begin{align}
V_{nm}(\rho, \theta) &= A_m(\theta) R_n(\rho), \\
A_m(\theta) &= \frac{1}{2\pi} \exp(jm\theta), \\
R_n(\rho) &= \begin{cases} 1, & n = 0; \\
\frac{1}{2\cos(\pi n\rho)}, & n \neq 0. \end{cases}
\end{align}

According to the MPEG-7 standard [43], ART coefficients are calculated for 3 radial and 12 angular directions \((0 \leq n < 3, 0 \leq m < 12)\). The coefficients are first normalized for scale invariance by dividing each coefficient with \( V_{00} \), and then quantized to a 4-bit representation, denoted by magnitude of ART, using a non-uniform binning.

A total of 35 4-bit magnitude of ART values are stored as features, thus requiring only 17.5 bytes of storage. The descriptor captures characteristic features of the shape well, enabling similarity-based retrieval. It is robust to partial occlusion of the shape and to perspective transformations (rotation, translation and scaling). In the proposed system, it is used to characterize the object’s shape and to determine the periodicity in its motion.

The distance between two region shape descriptors is measured using the Euclidean distance as

\begin{equation}
D_s = \sqrt{\sum_{i=1}^{35} (\Phi_s(O_1)[i] - \Phi_s(O_2)[i])^2},
\end{equation}

where \( \Phi_s(O_j)[i] \) is the \( i \)th magnitude of ART value of region shape descriptor of \( O_j \).

5.2.3. Contour Shape

Contour shape MPEG-7 descriptor describes a closed contour of an object or region. It is based on the Curvature Scale Space (CSS) representation of the contour.
A number of equidistant points are arbitrarily selected on the contour, and two series created: $X$ comprising of grouped $x$-coordinate values, and $Y$ containing all $y$-coordinate values. The contour is then gradually smoothed by a repetitive application of a low-pass filter with the kernel (0.25, 0.5, 0.25) to $X$ and $Y$. The filtering process is terminated when the contour becomes convex since the concave parts eventually flatten-out.

Vertical coordinates, denoted by $y_{css}$, are defined as the number of passes of the filter in the given point. Contour curvature function zero-crossing points separate concave and convex parts of the contour. Each zero-crossing is marked on the horizontal line corresponding to the smoothed contour and at the $x_{css}$ location corresponding to the position of this zero-crossing along the contour. The CSS image has characteristic peaks. The coordinate values of the prominent peaks $(x_{css}, y_{css})$ in the CSS image are extracted. The peaks are ordered based on decreasing values of $y_{css}$, transformed using a non-linear transformation and quantized. Finally, the eccentricity and circularity of the contour are also calculated, quantized and stored. The circularity is defined as:

\[
\text{circularity} = \frac{\text{perimeter}^2}{\text{area}},
\]  

while eccentricity is given by

\[
\text{eccentricity} = \sqrt{\frac{i_{20} + i_{02} + \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}{i_{20} + i_{02} - \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}},
\]

\[
i_{02} = \sum_{k=1}^{N}(y_k - y_c)^2;
\]

\[
i_{11} = \sum_{k=1}^{N}(x_k - x_c)(y_k - y_c);
\]

\[
i_{20} = \sum_{k=1}^{N}(x_k - x_c)^2,
\]

where $N$ is the number of points inside the contour shape, and $(x_c, y_c)$ is the center of mass of the shape.

The system stores and uses only global curvature which describes circularity $\Phi_{gc}$ and eccentricity $\Phi_{ge}$ of the object’s contour $C(\mathcal{O})$. Euclidean distances
(which are in this instance equivalent to the absolute differences), denoted $D_{gc}$ and $D_{cc}$, are used to measure the distances between global curvature features of two video objects.

5.3. Texture Features

MPEG-7 edge histogram descriptor is used to characterize object’s texture. This descriptor, denoted as $\Phi_t$, specifies the spatial distribution of five types of edges in local image regions. There are four directional edges (horizontal, vertical, 45 degree, and 135 degree edge) and one non-directional edge in each local region called a sub-image. According to the standard MPEG-7 Edge Histogram descriptor the input image is divided into into 4x4 non-overlapping blocks (sub-images). Thus, a total of 16 non-overlapping sub-images are obtained and for each sub-image a local edge histogram with 5 bins is generated. Hence, a total of $16 \times 5 = 80$ histogram bins is obtained.

The descriptor consists of 80 3-bit values called bin count. These values are non-linearly quantized to a 3-bit value corresponding to counts of 5 different types of edges within each of the 16 sub-images.

The distance between texture features of objects $O_1$ and $O_2$ is defined as the Euclidean distance between the bin counts:

$$D_t = \sqrt{\sum_{i=1}^{80} (\Phi_t(O_1)[i] - \Phi_t(O_2)[i])^2},$$  

where $\Phi_t(O_j)[i]$ is the value of the $i^{th}$ edge histogram bin count of the texture feature of object $O_j$.

6. Tracking of Video Objects based on Feature Update and Comparison

In the context of video surveillance, tracking of detected video objects across multiple frames refers to distinguishing all instances of a given object in the successive frames from the point the object enters the observed scene until it leaves it. Tracking difficulties can arise when the tracked object is occluded with other objects during tracking, gets obstructed by the background, or when scene is cluttered with objects that have similar features. When no tracking difficulties are present, tracking of objects essentially becomes a correspondence problem between objects in the previous frame and objects in the current frame.
To solve the frame-to-frame object correspondence, the proposed system relies on the assumption that both locality and global appearance features of an object are similar from frame to frame. Frame capture typically occurs at very high frequencies (25 or 30 Hz) allowing only relatively small changes in size, position, depth and overall appearance of an object to occur between subsequent frames. This is illustrated in Fig. 9 (a, b and c), showing the self-similarities of various features of a video object during tracking. It can be observed that smallest distances occur around the diagonal of the matrix indicating that the largest similarities occur in frames close to one another. A more complete comparison of overall and frame-to-frame self-similarities of four different video objects from the experimental set of stereo sequences is summarized in Table 1, where $\overline{D}_\phi$ and $\overline{D}^\phi$ denote the average overall and frame-to-frame distances between feature $\phi$ of the same tracked object.

<table>
<thead>
<tr>
<th></th>
<th>Daniel</th>
<th>Lakis</th>
<th>Liam</th>
<th>Carlos</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{D}_c$</td>
<td>0.001</td>
<td>0.001</td>
<td>0.009</td>
<td>0.006</td>
</tr>
<tr>
<td>$\overline{D}^c$</td>
<td>0.00032</td>
<td>0.00012</td>
<td>0.00046</td>
<td>0.00043</td>
</tr>
<tr>
<td>$\overline{D}_s$</td>
<td>23.108</td>
<td>23.743</td>
<td>24.694</td>
<td>24.400</td>
</tr>
<tr>
<td>$\overline{D}^s$</td>
<td>13.684</td>
<td>14.213</td>
<td>13.866</td>
<td>16.000</td>
</tr>
<tr>
<td>$\overline{D}_{gc}$</td>
<td>7.789</td>
<td>10.177</td>
<td>7.116</td>
<td>7.961</td>
</tr>
<tr>
<td>$\overline{D}^c_{gc}$</td>
<td>2.715</td>
<td>3.137</td>
<td>2.255</td>
<td>2.640</td>
</tr>
<tr>
<td>$\overline{D}_{ge}$</td>
<td>5.391</td>
<td>5.446</td>
<td>5.127</td>
<td>6.015</td>
</tr>
<tr>
<td>$\overline{D}^c_{ge}$</td>
<td>1.159</td>
<td>1.362</td>
<td>1.193</td>
<td>1.520</td>
</tr>
</tbody>
</table>

Table 1: Comparison of objects’ overall and frame-to-frame average distances in feature space for several features

In order to measure the frame-to-frame self-similarity of a video object $O$, a feature-distance vector $\langle \overline{D}_\phi(O) \rangle$, $\phi \in \{bw, bh, cx, cy, d, s, gc, ge, t\}$, is constructed and constantly updated using the extracted features in those
Figure 9: Feature distances of video object during tracking: (a), (b) and (c) show plots feature distances of color, region shape and texture features for object Daniel, respectively. Light diagonals indicate largest similarity for neighboring frames.
frames in which $O$ is not occluded with other tracked objects in the scene (and more generally, where there are no tracking difficulties observed for $O$).

Without loss of generality, it can be assumed that initially no objects are contained within the surveillance scene. When an object first enters the scene, it is labelled as new. A new object given a unique object ID and its features are extracted and stored. Also, a new feature-distance vector $\langle \overline{D}_\phi(O) \rangle$ is formed with its values set to 0. In the next frame $F_2$, the tracking system must establish a correspondence between the newly detected object $O^1$ in the previous frame $F_1$ and its corresponding image $O^2$ in the current frame. The following correspondence method is only used for matching the second instance of the newly detected object: If $O^2_1, \ldots, O^2_n$ is the list of objects in the current frame, then $O^1$ is identified as $O^2_j$ for that $j \in \{1, \ldots, n\}$ for which the distance of their features is minimal according to the majority voting principle. Majority voting is used since the feature distance measures are not normalized. The vectors of feature distances are compared among themselves coordinate-wise (feature-wise) so that each feature is separately compared with the same feature in other vectors and a sorting index is given in ascending order. The vector with the smallest sum wins as the closest one.

For each tracked object $O$ its feature-distance vector $\langle \overline{D}_\phi(O) \rangle$ is updated from its $(n - 1)^{th}$ instance to $n^{th}$ using the following recurrence relation adapted from [45]:

$$\overline{D}_\phi(O)^{(n)} = \overline{D}_\phi(O)^{(n-1)} + \frac{D_\phi(O, O) - \overline{D}_\phi(O)^{(n-1)}}{n}. \tag{22}$$

The method based on majority voting does not fully utilize the discriminatory information inherited in objects’ feature-distance vector. Thus, a similarity measure, denoted as $S$, is proposed for solving the object’s frame-to-frame correspondence problem after the second instance of the tracked object. When no tracking difficulties are detected, the object $O^N_j$ from $N^{th}$ frame is matched with an object from previous frame $O^{N-1}_i$ as follows:

$$O^N_j \sim O^{N-1}_i, \quad \text{if } S(O^{N-1}_i, O^N_j) \geq T_S, \tag{23}$$

$$S(O_1, O_2) = \sum_{i=1}^{k} \lambda_i(O_1, O_2), \tag{24}$$

$$\lambda_i(O_1, O_2) = \begin{cases} w_i, & D_\phi(O_1, O_2) \leq \rho_i; \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$
where $\langle \phi_1, \ldots, \phi_k \rangle$ are the used features, $w_i$ is the weight associated with feature $\phi_i$, and $\rho_i$ is the close proximity radius for features $\phi_i$. $\langle \phi_1, \ldots, \phi_k \rangle = \langle bw, bh, cx, cy, d, c, s, gc, ge, t \rangle$ so $k = 10$, $w_i$'s are all set to $1/k$, and close proximity radius is set to be linearly dependant on the current average frame-to-frame feature distance $\rho_i = aD_{\phi_i} + b$, where $a = 2$ and $b = 1$, except for the color feature where $b = 0.001$ since it uses a non-Euclidean distance. $S(O_1, O_2)$ represents the similarity measure of features of objects $O_1$ and $O_2$ which is based on the weighted sum of the frame-to-frame similar feature where the feature vectors are tested for similarity in own feature domain. The final parameter $T_S$ controls the lowest percentage of observed similarity needed for a match. The value of 0.7 for $T_S$ was used in the experiments, meaning that objects are matched if at least 70\% of their features are close.

When there are no matches for an object $O_i^{N-1}$, the system checks for possible occlusion. The occlusion is checked for by observing the projected trajectory of $O_i^{N-1}$ using the shift of $O_i^{N-1}$ from frame $N - 2$ to $N - 1$. The shift is given by its position features as $(\Phi_{cx}(O_i^{N-1}) - \Phi_{cx}(O_i^{N-2})$, $\Phi_{cy}(O_i^{N-2}) - \Phi_{cy}(O_i^{N-1})$). If the bounding box of the shifted $O_i^{N-1}$ overlaps with the bounding box of any of the objects in frame $N$, then the potential occlusion is detected. At that point the actual object masks are compared to determine if the objects occlude or not. If their masks overlap in any of the shape pixels, the objects are labelled as occluded. The detected occlusions in our system are tracked just like any other objects except with an additional information as to which previously tracked objects constitute the occlusion blob.

On the other hand, if there is no match for object $O_j^N$ in the system, it is concluded that either $O_j^N$ split from the occlusion or that it is a new object. If $O_j^N$ is in the close proximity of any of the currently tracked occlusions, occlusion splitting is discerned. Otherwise the object is tracked as new.

When occlusion splitting occurs, one must be able to determine which of the objects contained in the occlusion split from the occlusion blob $K$. To accomplish this, given the knowledge and features of objects constituting the occlusion, a subset of features denoted by stable features is used. Stable features are such features of a tracked object that are either roughly constant or periodic in nature. Global appearance features are considered stable due to their invariance to isometric transformations. As far as locality features are concerned, while position is not considered stable, depth and size features are considered stable only if their rough constancy is observed, which is the case for objects moving approximately perpendicular to the camera view. The concept of stable features follows the assumption that frame gap between
observed features of an object formed as a result of object’s occlusion is too large to use all features for similarity measure as it is the case for the frame-to-frame similarity, as illustrated in Fig. 9 and Table 1.

To determine which of the objects from $\mathcal{K}$ is in correspondence with $O_j^N$, the stable features of $O_j^N$ are compared to stable features of all tracked instances of $O_i$ and using majority voting the instance of $O_i$ with the smallest distance is selected, denoted $d_1$. This is then done for all other objects from $\mathcal{K}$, ending with a set of distances $d_1 \ldots d_n$. Finally if $d_i = \min(d_1, \ldots, d_n)$, then $O_j^N$ is set to correspond to object $O_i$. The feature-distance vector for $O_i$ is at that point updated accordingly with feature information from $O_j^N$.

7. Experiments and Results

A PC based application has been developed in order to evaluate the performance of the proposed object detection and tracking approach. Since the proposed method requires stereo sequences, and no such sequences are available as a benchmark, suitable surveillance-like stereo sequences taken outdoors have been created. Eight different sequences containing both rigid moving objects and persons were used to test the approach. We present evaluation results based on 2164 frames containing moving objects undergoing occlusion for extended periods of time and changing direction of motion while under occlusion. Tracking results for the data set can be obtained from http://www.cse.fau.edu/~oge/research.html, and all the sequences can be obtained from authors upon requests. Figure 10 shows the recall and precision values calculated based on tracking results obtained for each object present in the sequences. With the recommended parameters, the system achieved perfect tracking (recall and precision value of 1) for 61% of all objects present and recall and precision very close to ideal in all other cases.

Sample object detection results for eight outdoor sequences are illustrated in Fig. 11. For each sequence the original frame is shown, its segmentation result before and after the shadow removal, and detected objects along with their estimated depth. Accurate segmentation using a combination of BNN approach and an effective shadow removal can be observed in Fig. 11. In addition, objects’ depth also appear to be estimated accurately.

Fig. 12 shows the tracking results for the set of experimental sequences. Each newly detected object is assigned different color.

To evaluate the applicability of the system in a multi-camera scenario, additional experiments were performed on the full batch of sequences, pro-
Figure 10: Recall and precision for objects present in the test sequences.

cessing them in the order presented in Fig. 12. The features of detected objects from previously processed sequences were kept and used for tracking in subsequent sequences. It can be observed from Fig. 12 that all the different objects have been correctly identified by the tracker (their colors are different). For each currently tracked object, a trajectory is shown in corresponding color. Robustness of tracking in the presence of object occlusion can be observed even in cases when the tracked objects have similar color, shape and texture (in this case two walking persons in the same outfit).

8. Conclusion

An approach to robust detection and tracking of video objects from stereo videos, suitable for intelligent surveillance applications, is presented in the paper. The methodology utilizes video segmentation based on Bayesian neural networks to achieve accurate and fast segmentation of video objects. Segmentation results are further improved by a novel shadow removal algorithm capable of accurately removing shadows from detected video objects in real-time, thus improving the accuracy of objects appearance features. A stereo correspondence algorithm is proposed that is capable of estimating depth on the object level in real-time. Finally, a set of localized and global appearance features and corresponding object-similarity metrics is defined, that can be used to solve the problem of tracking effectively. The method is able to detect occlusion as well as to resolve an object splitting from occlusion and continue tracking it. Experimental results demonstrate robust tracking and detection of video objects in several stereo sequences.
References


Figure 11: Detection of video objects and their depth in several experimental outdoor sequences. Red arrows are used to indicate the sequential dependency of different detection phases.
Figure 12: Tracking of detected video objects based on their locality features (depth, size and position) and global appearance features (color, shape and texture) in several different outdoor sequences. The results demonstrate robustness in the presence of outdoor conditions and stable tracking after object occlusion.