# A Video-Based Drowning Detection System

Alvin H. Kam, Wenmiao Lu, and Wei-Yun Yau

Centre for Signal Processing, School of Electrical and Electronic Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

**Abstract.** This paper provides new insights into robust human tracking and semantic event detection within the context of a novel real-time video surveillance system capable of automatically detecting drowning incidents in a swimming pool. An effective background model that incorporates prior knowledge about swimming pools and aquatic environments enables swimmers to be reliably detected and tracked despite the significant presence of water ripples, splashes and shadows. Visual indicators of water crises are identified based on professional knowledge of water crisis recognition and modelled by a hierarchical set of carefully chosen swimmer descriptors. An effective alarm generation methodology is then developed to enable the timely detection of genuine water crises while minimizing the number of false alarms. The system has been tested on numerous instances of simulated water crises and potential false alarm scenarios with encouraging results.

## 1 Introduction

Automated video surveillance has been growing in prominence as an efficient way of intelligently detecting 'interesting' events in real-time [1],[5],[8],[14]. Such systems typically consist of a computer vision component that detects, segments and tracks moving objects and an inference module that interprets detected motions as events with semantic meaning. The technical challenges faced are thus two-fold. First are problems related to object detection and tracking such as dealing with shadows, lighting changes and effects of moving elements of the background. Secondly and no less challenging is the interpretation of the objects' motions into a description of detected actions and interactions. This paper provides fresh insights into these and additional unique problems faced in the development of a novel real-time video surveillance system capable of detecting potential drowning incidents in a swimming pool.

Our system consists of a network of overhead poolside cameras strategically placed such that the entire pool can be covered by the combined camera views. Each camera is mounted high and tilted downwards at a sharp angle towards the pool in order to minimize foreshortening effects and reduce occlusions while covering a large field of view. Video signals from each camera are processed by a computer vision module which detects, segments and tracks multiple swimmers within the visual field. The highly dynamic nature of a swimming pool precludes the use of traditional approaches based on background subtraction [11],[16], temporal differencing [9] or optical flow [15]. Furthermore, the significant presence

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2353, pp. 297–311, 2002.

of water splashes, shadows and the unpredictable effects of light refraction from disturbed water pose incredible challenges that could only be overcome using various novel techniques.

Intelligence to detect potential water crisis situations is incorporated into our event inference engine using a 3-step strategy. Firstly, visual indicators of a water crisis situation are identified based on professional knowledge of water crisis recognition [12],[13]. These indicators which contain specific behavioural traits universally exhibited by all troubled swimmers are then modelled by a hierarchical structure of swimmer descriptors. Low-level descriptors are first extracted from tracked swimmers based on which higher-level descriptors which carry more semantic meaning are progressively derived. The highest level descriptors model potential water crisis situations in a direct way. Finally, by combining expert domain knowledge and experimentation, we determine the appropriate duration for each modelled water crisis situation to be manifested continuously before an alarm is raised. This final step enables the timely detection of genuine water crises while minimizing the number of false alarms.

The remainder of the paper is organized as follows: section 2 describes the background model as well as the swimmer detection and tracking algorithm of the system. Section 3 explains the design of the event inference module while section 4 evaluates the performance of the system with various experimental results. We summarize our work and discuss some future directions in section 5.

## 2 Background Scene Modelling and Swimmer Tracking

In this section, we will describe:

(a) a robust background model capable of handling the unique dynamic nature of a swimming pool
(b) the computational models used for swimmer detection amidst the presence of shadows, splashes and noise
(c) the tracking strategy used in determining swimmer correspondence

### 2.1 Background Scene Modeling

The background of most swimming pools, unlike natural scenes, are relatively simple, consisting of very few distinct classes[1], for example corresponding to the water and lane dividers. Regions corresponding to each background class however experience considerable boundary movements from frame to frame due to water ripples. A good strategy for building a feasible model of the background is to first perform an unsupervised segmentation of the empty pool scene in order (i) to determine the number of distinct classes within the scene and (ii) build a global statistical model of the colour properties of each class. Then, in order to model the boundary movements of each background class, the spatial region covered

[1] *Classes* refer to groups of pixels that share similar colour characteristics and are significantly different from each other.

by each class is morphologically *dilated* using a suitable structuring element to generate an enlarged 'region-of-influence'. The main idea is to ultimately enable the detection of a swimmer pixel as one having a low probability of being realizations of any background classes whose region-of-influence include the pixel in question.

During system initialization, each video camera is designated a specific pool region to monitor, referred to as the *area of interest* (AOI). Pixels of the empty background scene of the AOI is then mapped to a selected colour space. Salient features whose recovery is necessary to identify distinct classes of the background, correspond to clusters in this colour space. An unsupervised clustering algorithm is essentially needed to infer both the number of clusters (classes) and the respective cluster centers (class properties) of the background scene.

A robust and efficient solution to this problem consists of using the kernel based mean shift procedure [2],[3]. The procedure essentially operates by calculating the *mean shift* vector of an image point $\mathbf{x}$ around the feature neighborhood region covered by a uniform kernel of chosen width $h$, $S_h(\mathbf{x})$:

$$M_h(\mathbf{x}) = \frac{1}{n_{\mathbf{x}}} \sum_{\mathbf{X}_i \in S_h(\mathbf{x})} (\mathbf{X}_i - \mathbf{x}) \tag{1}$$

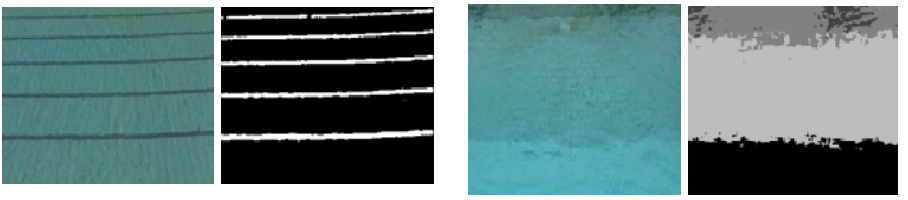
where $n_{\mathbf{x}}$ denotes the number of image points falling within $S_h(\mathbf{x})$. With a non-zero density gradient over the region $S_h(\mathbf{x})$, $M_h(\mathbf{x})$ is inherently pointing in the direction of most rapid increase of the density function and have a length proportional to the magnitude of the gradient. A mode-seeking clustering procedure can thus be devised, consisting of computation of the mean shift vector, $M_h(\mathbf{x})$ and translation of the kernel $S_h(\mathbf{x})$ at $\mathbf{x}$ by $M_h(\mathbf{x})$, i.e. $\mathbf{x}' = \mathbf{x} + M_h(\mathbf{x})$, and repeating this step iteratively for successive locations of $\mathbf{x}'$ until the magnitude of $M_h(\mathbf{x})$ approaches zero. Each image point thus 'gravitate' towards a point of convergence which represents a local mode of the density in the colour space.

The converged points are therefore the positions of cluster centers, which are essentially class representatives of dominant colours within the background scene. The scene could then be segmented by comparing the colour of each pixel with these class representatives and assigning each pixel to the class with the most similar colour. The number of distinct classes is generally determined by the choice of the kernel width $h$. Fortunately, the inherent robustness of the algorithm makes it possible to use the same kernel width to produce highly consistent segmentations for different background pool scenes.

The HSI (hue, saturation and intensity) colour space is chosen as it has been found to be more effective for swimmer segmentation compared to other colour spaces such as RGB or CIE Lab. Figure 1 shows the empty background scenes of two different pools and their respective unsupervised segmentation results in the normalized[2] HSI colour space using $h = 0.05$. As could be seen, the algorithm is able to effectively segment out distinct classes within each image[3].

[2] Hue, saturation and intensity features are normalized by scaling each feature to lie between 0 and 1.

[3] A small constant is typically added to the RGB values of very dark pixels before transformation to avoid the singularity of the hue feature (with negligible appearance

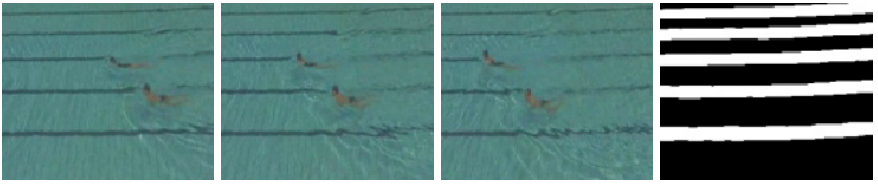


(a) (b)

**Fig. 1.** Empty background scenes of two swimming pools and their respective unsupervised segmentation results in the normalized HSI colour space utilizing mean shift clustering

Colour properties of each of the $k$ distinct classes of the background, $\{C_i\}_{i=1,2,..,k}$, are modelled using a multivariate Gaussian distribution with mean, $\boldsymbol{\mu}_{i,t}$ and full covariance matrix, $\boldsymbol{\Sigma}_{i,t}$; the suffix $t$ denoting the time-varying nature of the parameters.

The spatial region covered by each class is then morphologically dilated using a suitable structuring element to generate an enlarged 'region-of-influence' to model boundary movements due to the effects of water ripples. Figure 2 (a)-(c) illustrates the considerable boundary movements of the lane divider and water regions compared to that of the calm empty scene of figure 1(a). The enlarged regions-of-influence of the lane dividers (shown in figure 2(d)) and water (covering the entire image) provides effective spatial support allowances due to boundary movements.



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

**Fig. 2.** (a)-(c): Water ripples changing background classes' boundaries from that of a calm empty scene; (d): dilated regions-of-influence of the lane divider class (structuring element used: 10-by-15 matrix of all ones)

### 2.2 Swimmer Detection and Tracking

To detect swimmers in the pool, the normalized Mahalanobis distance is computed between each pixel of the AOI, $x_j$ and the most similar background class whose region-of-influence include the pixel in question:

$$D(x_j) = \arg\min_i \{D_m(x_j|C_i)\} \tag{2}$$

changes to the pixels involved). The cyclic property of hue is also given special consideration using methods outlined in [17].

where $D_m(x_j|C_i) = \ln|\boldsymbol{\Sigma}_{i,t}| + \left(\mathbf{x}_j - \boldsymbol{\mu}_{i,t}\right)^T \boldsymbol{\Sigma}_{i,t}^{-1}\left(\mathbf{x}_j - \boldsymbol{\mu}_{i,t}\right)$ with $\mathbf{x}_j$ corresponding to the HSI feature vector of pixel $x_j$ and $C_i$ being background classes (with mean colour, $\boldsymbol{\mu}_{i,t}$ and covariance, $\boldsymbol{\Sigma}_{i,t}$) whose region-of-influence include $x_j$.

The system will consider a pixel to potentially correspond to a swimmer if the measure given by equation(2) is above a pre-determined threshold (typically zero). This may however include pixels due to water splashes or shadows which are not explicitly included into the background model. Through our experiments, water splashes are found to correspond to pixels having the following colour characteristics: (i) normalized hue being above 0.40, (ii) normalized saturation being below 0.27, and (iii) normalized intensity being above 0.90 or the maximum intensity amongst all background classes, whichever is higher. Moreover, we also found that shadow pixels have the common characteristic of possessing a normalized saturation value above 0.25. Fortunately, these colour attributes are considerably different from that of a typical swimmer. These facts enable a swimmer pixel to be detected on the basis of:

(i) it being neither a water splash nor a shadow pixel **and**
(ii) its dissimilarity measure of equation(2) being above the pre-determined threshold

If sufficient number of swimmer pixels are detected for a particular frame, a multivariate Gaussian appearance model of swimmers could be built with its mean and covariance parameters computed from the colour properties of these pixels. The close similarities between swimmer pixels in the HSI colour space makes it feasible to form a single collective model for all swimmers. Pixel classification for subsequent frames could then be implemented on all non water-splash or shadows pixels using maximum likelihood:

$$\begin{aligned} X_j &= \arg\max_{C_i} p\left(X_j = C_i|\boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}\right) \\ &= \arg\min_{C_i}\left[\ln|\boldsymbol{\Sigma}_{i,t}| + \left(\mathbf{x}_j - \boldsymbol{\mu}_{i,t}\right)^T \boldsymbol{\Sigma}_{i,t}^{-1}\left(\mathbf{x}_j - \boldsymbol{\mu}_{i,t}\right)\right] \end{aligned} \tag{3}$$

where $X_j$ denotes the class label of a non water-splash or shadow pixel $x_j$ with HSI feature vector $\mathbf{x}_j$ while $C_i$ now comprises the newly introduced 'swimmer class' as well as all background classes whose region-of-influence include $x_j$.

This classification scheme has been found to produce considerably better results compared with the procedure utilizing a pre-determined threshold. Our swimmer detection strategy thus consists of using the former scheme until sufficient swimmer pixels are detected, after which maximum likelihood classification is utilized. For either detection schemes, all background pixels will be labelled with the most similar background class to be used in the updating of background model parameters later on. A binary map separating swimmer and non-swimmer pixels is also formed. A connected component algorithm [4] is subsequently applied to remove small isolated regions with the resulting contiguous blobs corresponding to individual swimmers in the frame.

The trajectory of each swimmer blob is then assigned an associated first order Kalman filter that predicts the blob's position and velocity in the next

frame. As the Kalman filter is the 'best linear unbiased estimator' in a mean squared sense, the filter equations correspond to the optimal Bayes' estimate for Gaussian processes.

Normalized distances between detected swimmer blobs and predicted swimmer positions are calculated using:

$$D_m\left(d_i, \boldsymbol{\psi}_j\right) = \left(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{d_i}\right)^T \boldsymbol{\Sigma}_{d_i}^{-1} \left(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{d_i}\right) \tag{4}$$

where $d_i$ represents a detected swimmer blob with position centroid, $\boldsymbol{\mu}_{d_i}$ and position covariance, $\boldsymbol{\Sigma}_{d_i}$ while $\boldsymbol{\psi}_j$ denotes the Kalman filter predicted position of swimmer $j$. Compared to the Euclidean distance, this metric has the added advantage of incorporating the shape information of swimmers.

A correspondence between a swimmer blob and a Kalman filter predicted position is assumed if: (i) their distance given by equation(4) is minimum compared with all other predicted positions, and (ii) this minimum distance is less than a certain maximum threshold. Note that this matching criteria may result in matching more than one swimmer blob per predicted position. This is entirely desirable as in these cases, the multiple blobs typically correspond to parts of the same swimmer, fragmented due to noise or water splashes.

Another issue to be resolved concerns occlusions, i.e. cases where two or more nearby swimmers merge and are detected as a single large blob. Occlusions correspond to instances where a detected swimmer blob:

(i) yields comparatively small normalized distances with two or more predicted swimmer positions which are not matched to other swimmer blobs in the current frame
(ii) is of a considerably larger size than each of the above corresponding swimmer blobs in the previous frame

The high appearance similarities between swimmers precludes the use of standard occlusion handling techniques which rely on constructing appearance models based on colour or geometrical features [6],[10]. We can however make use of the fact that the way our cameras are mounted (high up and tilted downwards at a sharp angle) ensures minimal degree of occlusion between swimmers. Thus whenever occlusion occurs, it is possible to approximately determine which pixels correspond to each partially occluded swimmer within the large detected blob. This is done by calculating the normalized Mahalanobis distance (in the *spatial* domain) between each pixel of the large blob and each corresponding swimmer blob in the previous frame (with the position mean adjusted to the predicted position given by the respective Kalman filter). Each pixel in the blob is then labelled in accordance to the swimmer label yielding the minimum distance:

$$\Psi_j = \arg\min_{o_i} \left[\ln\left|\boldsymbol{\Sigma}_{o_i}\right| + \left(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{o_i}\right)^T \boldsymbol{\Sigma}_{o_i}^{-1} \left(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{o_i}\right)\right] \tag{5}$$

where $\Psi_j$ denotes the swimmer label of pixel with coordinates $\boldsymbol{\psi}_j$ within the large detected blob, while $o_i$ represents one of the (presently occluded) swimmer blobs in the previous frame, with its corresponding position covariance, $\boldsymbol{\Sigma}_{o_i}$

and predicted position centroid, $\boldsymbol{\mu}_{o_i}$. Figure 3 shows an example of the devised partial occlusion handling scheme. Although the procedure is not faultless, it assigns pixels belonging to each swimmer fairly effectively for most cases.

**Fig. 3.** Illustration of partial occlusion handling scheme. 1st row: sample frames of a video sequence containing instances of partial swimmer occlusion (2nd, 3rd and 4th frames); 2nd row: detected swimmer blobs; 3rd row: pixels corresponding to individual swimmers depicted in different shades of grey

Leftover unmatched predicted swimmer positions are then analyzed to determine if they correspond to swimmers making their way out of the monitored area (i.e. their predicted positions lie within the borders of the AOI). If so, they are removed from the tracking list; otherwise, the system will assume that the swimmers have been left out during the segmentation process[4]. In the latter case, the missing swimmer's silhouette in the previous frame is then artificially included as an additional blob in the current frame, centred on its predicted position. Similarly, unmatched swimmer blobs in the current frame are analyzed to determine if they are swimmers making their way into the monitored area (i.e. their centroids lie within the borders of the AOI). If so, they will be designated as new swimmers and included in the tracking list; otherwise, they are simply regarded as segmentation errors and are subsequently erased.

### 2.3 Extracting Low-Level Descriptors

Multiple low-level descriptors are extracted from each swimmer during the tracking process. For a swimmer covering a region $R$ containing $N$ pixels within the ROI Cartesian coordinate system, the descriptors extracted include the:

[4] This can sometimes happen if pixel regions of a swimmer are too fragmented resulting in their removal by the connected component algorithm.

(i) **Centroid**:

$$\bar{x} = \frac{1}{N} \sum_{(x,y)\in R} x, \quad \bar{y} = \frac{1}{N} \sum_{(x,y)\in R} y \tag{6}$$

(ii) **Orientation angle**, defined as the angle of axis of the least moment of inertia and given by:

$$\theta = \frac{1}{2} \tan^{-1} \left[ \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right] \tag{7}$$

where $\mu_{p,q}$ denotes the $(p, q)$ central moments:

$$\mu_{p,q} = \sum_{(x,y)\in R} (x - \bar{x})^p (y - \bar{y})^q \tag{8}$$

A positive and a negative value of the argument denominator in equation(7) correspond to a measurement with reference to the horizontal and vertical axis respectively. Either way, the angle is measured in an anti-clockwise direction from the corresponding axis.

(iii) **Parameters of the best-fit ellipse**, where the best-fit ellipse is the ellipse whose second moment equals that of the swimmer [7]. Let $a$ and $b$ denote the lengths of the major and minor axes of the best-fit ellipse respectively. The least and the greatest moments of inertia for an ellipse are:

$$I_{min} = \frac{\pi}{4} a b^3, \quad I_{max} = \frac{\pi}{4} a^3 b \tag{9}$$

For orientation $\theta$, the above moments can be calculated as:

$$I'_{min} = \sum_{(x,y)\in R} \left[ (x - \bar{x}) \cos\theta - (y - \bar{y}) \sin\theta \right]^2 \tag{10}$$

$$I'_{max} = \sum_{(x,y)\in R} \left[ (x - \bar{x}) \sin\theta - (y - \bar{y}) \cos\theta \right]^2 \tag{11}$$

For the best-fit ellipse we want $I_{min} = I'_{min}$, $I_{max} = I'_{max}$, which gives:

$$a = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[ \frac{(I'_{max})^3}{I'_{min}} \right]^{\frac{1}{8}}, \quad b = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[ \frac{(I'_{min})^3}{I'_{max}} \right]^{\frac{1}{8}} \tag{12}$$

The $a$ and $b$ parameters provides a reasonably good model of the length and width of the tracked swimmer.

Besides the above descriptors, we also keep track of the amount of water splash pixels within each swimmer's vicinity, the cumulative area of pixels covered by each swimmer and the swimmer's average colour saturation. The utility of these information for water crisis inference will be explained in the section 3.

### 2.4 Updating Model Parameters

Parameters of the background and swimmer models cannot be expected to stay the same for long periods of time. Background model parameters must be updated to cater for changes in the amount of water disturbances in the pool and overall lighting conditions. Likewise, swimmer model parameters need updating to closely reflect appearance attributes of swimmers being tracked.

The parameter updating strategy employed is simple but effective. Parameters of the multivariate Gaussian model of each class, namely its mean and full covariance are simply recalculated based on the colour values of pixels being classified as belonging to the class for all frames since the last update.

Our system operates at the rate of **8** frames per second. In order to maintain the accuracy of swimmer detection, we have found that the background model parameters need to be updated every frame whereas swimmer model parameters need updating once about every 24 frames. This updating frequency mismatch is due to the fact that the covariance matrix of each background class is highly dependent on the amount of water disturbances in the pool, which could change quite significantly within a short period of time. On the other hand, the appearance attributes of swimmers remain roughly the same over longer periods of time.

## 3 Water Crisis Inference

According to [12],[13], there are strictly two types of water crises: **distress** situations and **drowning** situations. Drowning situations can be further subdivided into groups involving active and passive victims. Distressed situations involve swimmers with varying degrees of skill who are unable to swim to safety due to fatigue or other reasons. A distressed swimmer is either positively or neutrally buoyant (due to his swimming skills) and exhibit voluntary actions such as actively struggling or waving. Active drowning situations meanwhile involve non-swimmers who due to their lack of swimming skills, alternate between negative and neutral buoyancy. The resultant feeling of suffocation triggers a constellation of universal instinctive struggle movements. Both distress and active drowning victims' bodies are perpendicular in water and they are not able to move in a horizontal or diagonal direction. Passive drowning victims slip under water without waving or struggling because of a sudden loss of consciousness[5].

Primitives of the above conditions could be modelled using a number of intermediate-level descriptors which are derived from low-level descriptors extracted during the tracking process:

1. **Speed**: A swimmer's (translational) speed is computed as the difference in centroid positions which are averaged over non-overlapping 8-frame blocks. The way our cameras are mounted minimizes foreshortening effects and thus

[5] This type of situation is generally caused by a heart attack, stroke, hyperventilation, a blow to the head, cold water immersion or excessive drinking of alcoholic beverages.

the rate of change of a swimmer's centroid could roughly be taken to represent his swimming speed.

2. **Posture**: A swimmer is typically regarded as being in a vertical position if his orientation angle is measured against the vertical axis and as being in a horizontal position otherwise. Position is however deemed undetermined if the ratio of the body's length to its width (computed using parameters of the best-fit ellipse) is consistently close to unity[6]. Posture is defined as the dominant position maintained by a swimmer over a non-overlapping 8-frame window.
3. **Submersion index**: A sinking swimmer usually exhibits a higher colour saturation as light reflected from his body passes through a certain depth of water which adds a blanket of bluishness to the skin appearance. This is clearly shown in figure 4 where the mean saturation value of a detected swimmer increases as he is progressively sinking deeper below the water surface. The submersion index is defined as the difference between the swimmer's mean colour saturation (averaged over non-overlapping 8-frame blocks) and the (bounded) minimum saturation value of the swimmer since being tracked.
4. **Activity index**: A swimmer's activity index is defined as the ratio between the cumulative area of pixels covered by the swimmer over an 8-frame period and the average area of the best-fit ellipse over the same duration. Figure 5 illustrates this concept of activity level measurement; note the significantly different activity index calculated for a swimmer in distress and one treading water.
5. **Splash index**: The number of splash pixels within an area extending 50 pixels in all 4 directions beyond the swimmer's bounding box is computed for each frame. The splash index is defined as the maximum number of splash pixels detected inside this area over a moving 24-frame window.

These intermediate-level descriptors clearly carry more semantic meaning than their low-level counterparts which they are derived from.

High-level descriptors are derived from intermediate-level descriptors, which are first segregated into distinct groups covering a range of numerical values. Membership for each grouping is determined based on pre-defined percentiles of corresponding descriptor values from a database of normal swimmers as shown in table 1.

There are a total of four high-level descriptors which are computed as follows:

1. *PD* or **P**otential **D**istress: Correspond to a swimmer who has: (i) low speed, (ii) a vertical or undetermined posture, (iii) a high activity index and (iv) a high splash index.
2. *PAD* or **P**otential **A**ctive **D**rowning: Correspond to a swimmer who has: (i) low speed, (ii) a vertical or undetermined posture, (iii) at least a medium activity index and (iv) at least a medium submersion index.

[6] In these cases, the swimmer is simply too 'round' for the orientation angle to be estimated reliably.

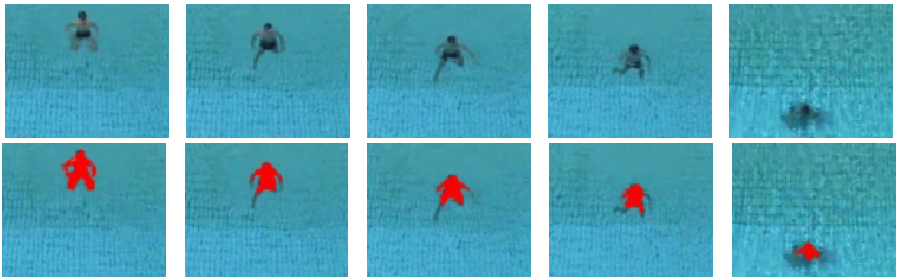



**Fig. 4.** Illustration of the effects of water submersion on a swimmer's colour saturation. 1st row: sample frames of a video sequence; 2nd row: segmented swimmer region, depicted with the mean normalized saturation value



**Fig. 5.** Difference in activity index for fast and slow movements. Columns 1-4: sample frames of a video sequence with best-fit ellipse enclosing detected swimmer; column 5: cumulative area of pixels covered by detected swimmer over 8-frame period

**Table 1.** Groupings of intermediate-level descriptors according to selected thresholds and their corresponding numerical range

| Descriptor | Group | Threshold | Numerical Range |
|---|---|---|---|
| Speed | Very slow | < 1 percentile of normal swimmers | < 2 pixels/sec |
| | Slow | < 5 percentile of normal swimmers | Btw 2 and 7 pixels/sec |
| | Normal-slow | < 10 percentile of normal swimmers | Btw 7 and 10 pixels/sec |
| | Normal | > 10 percentile of normal swimmers | > 10 pixels/sec |
| Submersion Index | Above water | < 75 percentile of normal swimmers | < 0.05 |
| | Borderline | Btw 75 and 85 percentile | Btw 0.05 and 0.075 |
| | Under water | > 85 percentile of normal swimmers | > 0.075 |
| Activity Index | Low | < 75 percentile of 'stationary' swimmers | < 1.8 |
| | Medium | Btw 75 and 85 percentile | Btw 1.8 and 2 |
| | High | > 85 percentile of 'stationary' swimmers | > 2 |
| Splash Index | Low | < 75 percentile of 'stationary' swimmers | < 400 |
| | Medium | Btw 75 and 85 percentile | Btw 400 and 600 |
| | High | > 85 percentile of 'stationary' swimmers | > 600 |

3. *PPD* or **P**otential **P**assive **D**rowning: Correspond to a swimmer who has: (i) very low speed and (ii) a high submersion index
4. *N* or **N**ormal: Correspond to a swimmer who does not exhibit any of the 3 states above.

These high-level descriptors are Boolean variables (yes-no) and are updated every second (block of 8 frames) due to their dependence on intermediate-level descriptors which are refreshed with this frequency. Their relationship with lower level descriptors is depicted within the full multi-level descriptor set in figure 6.

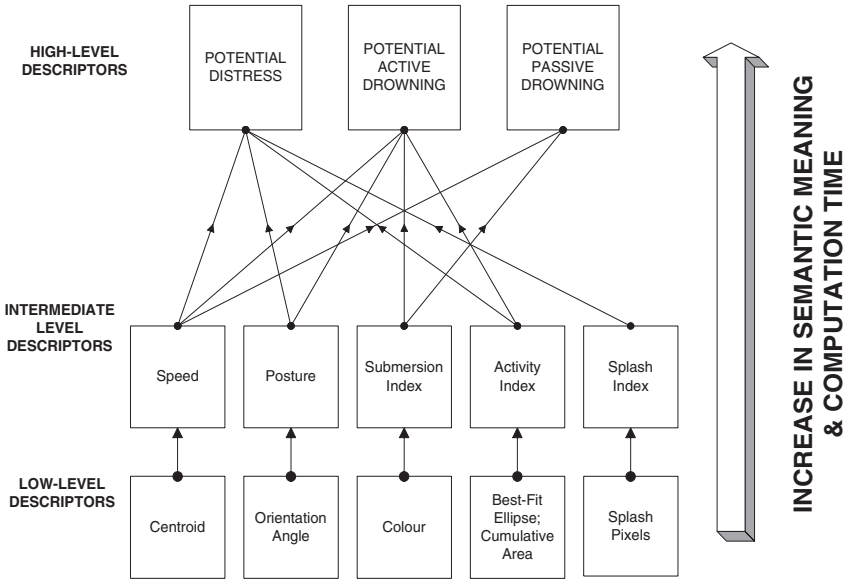



**Fig. 6.** The multi-level swimmer descriptor set

Our system is designed to raise two different types of alarms, a *yellow* alert providing initial warning and a *red* alert generating an alarm call. A yellow alert will be raised if a swimmer registers a (a) PD status for **5** consecutive seconds or (b) PAD status for **5** consecutive seconds or (c) PPD status for **5** consecutive seconds. This alarm status will be elevated to a red alert if (a) the PD status is registered for **8** consecutive seconds or (b) the PAD status for **10** consecutive seconds or (c) the PPD status for **10** consecutive seconds.

The main difficulty in evaluating our system's performance is the absence of footages of real distress or drowning occurrences at swimming pools. Fortunately, there is a very unique lifeguard training video entitled *On Drowning* (distributed by the American Camping Association) which is one of the very few sources of recorded *actual* near drowning situations. In fact, the New York State Health Department currently recommends that all camps show this video to their staff to familiarize them with the appearance of water crises indicators. We have several volunteers study this video very carefully in order to simulate water crises behaviour as realistically as possible to test the performance of our system.

## 4 Experiments and Performance Evaluation

A total of about one hour's worth (≈ 29,000 frames) of relevant video sequences was used to evaluate the system. From this amount, we set aside about half of all sequences that does not contain water crises simulations to derive threshold values for the grouping of intermediate-level descriptors. The remaining sequences were used to test the system.

Figure 7 depicts a typical case of a realistic water distress simulation by one of our volunteers and the corresponding system response. A professional lifeguard verifies that our volunteer exhibits symptoms of water distress from frames 96 to 170 of the recorded video sequence. As could be seen, the swimmer's activity and splash indices increased dramatically during this period, hereby illustrating the effectiveness of these descriptors in detecting water distress. Our system registers a PD status for the swimmer from frame 104; a yellow alert is raised by frame 144 and a red alert by frame 168 (about 9 seconds from onset of distress). The swimmer is returned to the normal state and the red alert withdrawn by frame 176.

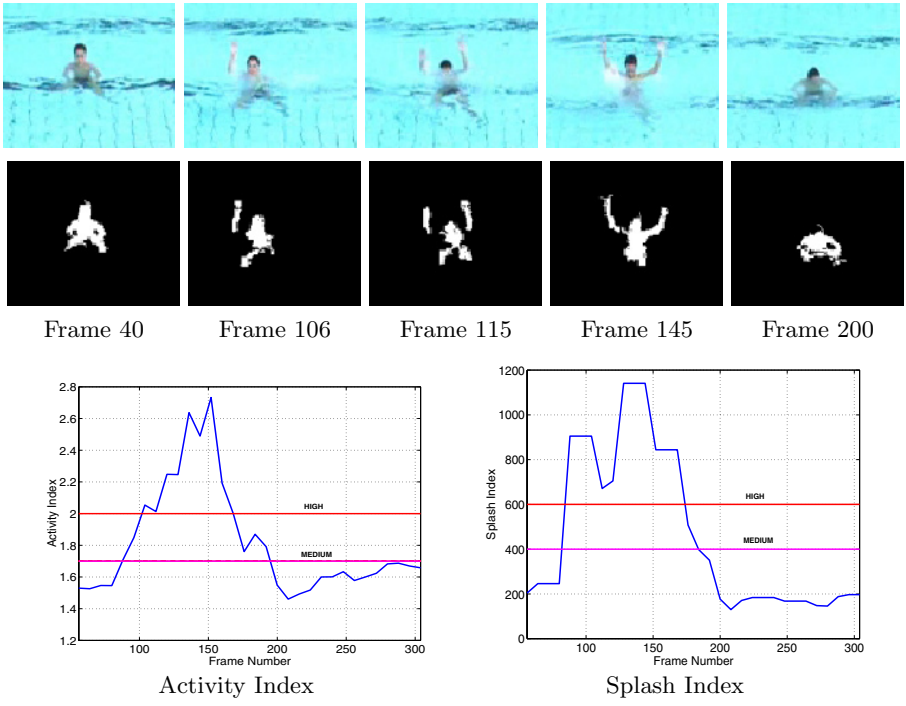



**Fig. 7.** Simulated water distress situation. First two rows: sample frames of the video sequence and corresponding swimmer segmentation. Third row: Graphs tracking changes in activity and splash indices of swimmer

Table 2 provides more performance evaluation of our system by depicting the comparison between the actual and detected water crises frames and the delay in

triggering a red alert from the onset of a (simulated) crisis. All water crises are realistically simulated by our volunteers and the system is able to successfully detect each situation with a red alert delay of between 8.4 and 12.5 seconds from the onset of a crisis.

**Table 2.** Comparison between actual and detected water crises frames

| Sequence | Water Crises Frames | Detected crises frames | Red alert delay |
|---|---|---|---|
| Distress1 | distress (105-438) | distress (120-440) | 9.9 secs |
| Distress2 | distress (31-229) | distress (56-240) | 11.1 secs |
| Distress3 | distress (301-520) | distress (304-520) | 8.4 secs |
| Distress4 | distress (12-541) | distress (24-552) | 9.5 secs |
| Distress5 | distress (85-219) | distress (88-222) | 8.4 secs |
| ADrown1 | active drowning (131-213) | active drowning (135-224) | 10.5 secs |
| ADrown2 | active drowning (95-190) | active drowning (103-192) | 11.0 secs |
| ADrown3 | active drowning (173-247) | active drowning (177-263) | 10.5 secs |
| ADrown4 | active drowning (230-315) | active drowning (247-328) | 12.1 secs |
| PDrown | passive drowning (172-365) | passive drowning (192-376) | 12.5 secs |

The system's response is also evaluated on a variety of potential false alarm scenarios involving a myriad of harmless but possibly 'deceiving' swimmer antics. This exercise basically evaluates the system's response on non-troubled swimmers who are not actively swimming. As could be seen in table 3, the number of false alarms generated by the system is minimal. False red alerts of water distress arise in extreme cases of abrupt movements and violent water splashes sustained over a considerable period of time. Natural occurrences of these swimmer behaviour are generally quite rare.

**Table 3.** System response to various potential false alarm scenarios

| Sequence | Description | False Alarms |
|---|---|---|
| Tread | Treading water | None |
| Play1 | Shallow underwater manoeuvres | None |
| Play2 | Continual abrupt movements with violent splashings | 2 red alerts |
| Play3 | Water surface antics with considerable splashings | None |
| Play4 | 'Playing dead' on water surface | None |
| Play5 | Deep underwater manoeuvres | None |
| Play6 | Combination of numerous manoeuvres with splashings | 1 yellow alert |

## 5 Summary and Discussion

In this paper, we investigated numerous technical challenges faced in the development of a novel real-time video surveillance system capable of detecting potential drowning incidents in a swimming pool. This work provides a number of interesting insights into human detection and tracking within a dynamic environment and the recognition of highly complex events through the incorporation of domain expert knowledge.

Perhaps a more important contribution lies in the generic computational framework for semantic event detection where visual information is progressively and systematically abstracted by moving up a hierarchical structure of descriptors. The developed framework is thus extensible for other applications by using domain-independent low-level descriptors to generate domain-specific intermediate descriptors and relying on established rules derived from the specific domain to facilitate high-level inference.

The immediate future work is for us to expand the sophistication of the existing descriptor set to facilitate more accurate event modelling. We are also currently investigating a structured approach to integrate inference and machine learning which enables the system performance to improve automatically over time.

## References


1. Buxton, H., Gong, S.: Advanced Video Surveillance using Bayesian Networks. Proc. IEEE Workshop Context-Based Vision (1995) 111–122.
2. Cheng, Y: Mean Shift, Mode Seeking and Clustering. IEEE Trans. Patt. Anal. Machine Intell. vol. 17(8) (1995) 770–799.
3. Fukunaga, K., Hosteler, L.D.: The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Trans. Info. Theory vol. 21 (1975) 32–40.
4. Gonzalez, R., Woods, R.: Digital Image Processing, Addison Wesley, Massachusetts (1992).
5. Haritaoglu, I., Harwood, D., Davis, L.: W4: Real-Time Surveillance of People and their Activities. IEEE Trans. Patt. Anal. Machine Intell. vol. 22(8) (2000) 809–830.
6. Intille, S., Davis, J., Bobick, A.: Real-Time Closed-World Tracking. Proc. IEEE Conf. Comp. Vision Patt. Recog., San Francisco (1997) 697–703.
7. Jain, A.: Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs (1989).
8. Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., Russell, S.: Towards Robust Automatic Traffic Scene Analysis in Real-Time. Proc. Int. Conf. Patt. Recog. (1994) 126–131.
9. Lipton, A., Fujiyoshi, H., Patil, R.: Moving Target Classification and Tracking from Real-Time Video. Proc. IEEE Workshop Appls. Comp. Vision (1998) 8–14.
10. Lu, W., Tan, Y.P.: A Color Histogram Based People Tracking System. Proc. IEEE Int. Symp. Circuits & Systems, Sydney (2001).
11. Makarov, A.: Comparison of Background Extraction based Intrusion Detection Algorithms. Proc. IEEE Int. Conf. Image Process. (1996) 521–524.
12. Pia, F.: Observations on the Drowning of Non-Swimmers. Journal Phy. Ed. (1974).
13. Pia, F.: The RID Factor as a Cause of Drowning. Parks and Recreation (1994).
14. Stauffer, C., Grimson, W.: Learning Patterns of Activity using Real-Time Tracking. IEEE Trans. Patt. Anal. Machine Intell. vol. 22(8) (2000) 747–757.
15. Wixson, L.: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE Trans. Patt. Anal. Machine Intell. vol. 22(8) (2000) 774–780.
16. Wren, C., Azarbayejani, A., Darrel, T., Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. IEEE Trans. Patt. Anal. Machine Intell. vol. 19(7) (1997) 780–785.
17. Zhang, C., Wang, P.: A New Method of Color Image Segmentation based on Intensity and Hue Clustering. Proc. IEEE Int. Conf. Patt. Recog., Barcelona (2000) 613–616.