Video-based Training Registration for Swimmers

E. Pogalin[1,2], A.H.C. Thean[2], J. Baan[2], N.W. Schipper[2], A.W.M. Smeulders[1]

[1]Intelligent Systems Lab Amsterdam, Faculty of Science, University of Amsterdam, The Netherlands

[2]Business Unit Monitoring Systems, Netherlands Organization for Applied Scientific Research (TNO), The Netherlands

Abstract

During the last decade, performance improvements in top sports have been increasingly driven by technological innovations. This paper discusses the application of video analysis for training registration in swimming. In current practice, coaches have limited means to evaluate objectively and quantitatively how a training session was carried out. We propose the use of a video-based registration system in order to help the coach in acquiring such information. The system uses multiple cameras to cover the swimming pool. By using a simple background modeling and blob tracking method swimmers are tracked and their lap times are estimated. The main limitation of the system is the failure to detect swimmers at the pool ends while they are resting or underwater. This can lead to the necessity to perform manual interactions to associate laps to swimmers and a systematic underestimation of the lap times of typically 1.5 seconds. Our results can be used to formulate training guidelines that can help overcome some limitations of the system so that, with little or no manual effort, the system could be used in practice to do quantitative measurements. The information on the actual training performance of the swimmers could then be compared with the training schedule made beforehand and used to further optimize the training program.

KEY WORDS: TRAINING REGISTRATION, VIDEO ANALYSIS, TRACKING, SWIMMING

Introduction

In recent years, performance improvements on top sports have been more and more driven by technological innovations. The rapid development of computer and information technology creates new opportunities to analyze aspects of sports that were previously out of reach. One particular application of technology that is growing in popularity is the use of video to analyze sport training (Wang et al., 2004). For example, it has been used to analyze a hand-ball game (Perše et al., 2006), soccer (Xu et al., 2004) and golf (Urtasun et al., 2005). The goal of video analysis is to evaluate the athlete’s performance during training sessions, thereby providing assistance to coaches and athletes. This paper will discuss the application of video analysis to swimming, in particular to the training sessions of top swimmers.
Using current practice, swimming coaches spend a lot of time making training programs, but they have few objective and quantitative measures to evaluate how the training was carried

out. The evaluation is limited to visual inspection and some time measurements with a stopwatch. At the end of the session, the swimmers fill in an evaluation form on how they experienced the session and to measure physical tiredness, lactate measurements are made by drawing blood samples. If several swimmers are training together the coach cannot give full attention to each of them and it is difficult to log all previous training data to make comparisons between training sessions. Our aim is to develop a video analysis system that will provide quantitative evaluation metrics for full training sessions of multiple swimmers and allow training schedules to be optimized.

In the last decade, video-based motion analysis has been used to improve technical performance in swimming. Usually underwater cameras are used to record swimmers and during playback the coach can show them where they should correct or improve the strokes. The interactive video software developed by Dartfish (Dartfish, 2006) has been used by many coaches all over the world. By inspecting the recording and manually inserting various markers on the video, the coach can measure aspects of the swimmer’s motions. Another type of registration is hand force measurement, combined with video recording (STR, 2006). During the training, the developed Aquanex system measures the forces exerted by both hands in real-time. Analyzing the force curves in time makes it possible to reinforce the positive elements and identify the limiting factors. These systems are useful for giving detailed information about technique but not for registering the training sessions of multiple swimmers.

In terms of Computer Vision, the problem of monitoring swimmers is known as tracking persons in surveillance. In the last few years visual surveillance received growing attention (Hu et al., 2004). Two main approaches to this problem can be distinguished in literature, the motion-based approach and model-fitting approach.

The motion-based approach involves modeling the background scene and doing motion segmentation to detect foreground objects (see Stauffer & Grimson, 1999 and Elgammal et al., 2002). The background model is updated continuously to cope with scene variations and moving foreground objects are associated to previously-detected objects to form tracks. This approach has been applied to swimmers by Eng et al. (2006), who focused on drowning detection. Using this approach the object of interest is identified by its motion and this can have disadvantages when the tracked object stops moving. It is most appropriate for stationary cameras in scenes that do not show strong variation and is computationally efficient.

The model-fitting approach focuses on modeling the object to be tracked and fitting this model directly to the input data. Using this approach the object of interest is identified by its appearance and this can have disadvantages when the object appearance is highly-variable or unpredictable. Baumberg & Hogg (1995) and Gavrila & Philomin (1999) used a silhouette-based representation to track humans in video sequences, but when the object is partially occluded this method will not perform correctly. Nguyen & Smeulders (2004) learned the intensity pattern of the object of interest and the local surrounding background, which allowed robust tracking even under severe changes of both the foreground and the surrounding background. However, this algorithm has to be initialized manually and tracking multiple objects requires more computing power when compared to the first approach. Viola et al. (2005) introduced a method for detecting pedestrians in surveillance video which learns the appearance of full human figures from a large dataset. The trained pedestrian detector can be run in almost real-time (4 fps) but it does not perform well with partially-occluded human figures.

This paper presents results obtained with a prototype video analysis system based on a motion-based approach to swimmer tracking. The system is not required to work fully-automatically and is designed for compliant subjects who are willing to adhere to certain guidelines about behavior in a controlled environment. The results obtained allow us to identify important issues for the design of future systems. Our analysis includes considerations of hardware, swimming pool infrastructure, training schedule guidelines and interfaces but we pay particular attention to algorithm requirements and performance evaluation.

Methods

This section describes the hardware setup used to capture the training session, the data processing algorithm and the performance evaluation method used to analyze the results.

We recorded 45 minutes of video during training sessions of the Dutch youth national swimming team. Only three lanes were used during the training. The bottom lane was occupied by a single swimmer who swam back and forth in separate halves of the lane. The middle lane was occupied by two swimmers who each swam back and forth in separate halves of the lane. The third lane was occupied by four swimmers who swam in a circulating pattern i.e. swimmers moved in a different direction in each half of the lane. To have an overview of the whole 50m pool length we used 3 PAL cameras (with 720x576 pixels resolution each) with overlapping views elevated approximately 4m high at the pool side on tripods (Figure 1). The use of tripods is motivated by the fact that training sessions are often held in different pools so we need a mobile setup that can be easily built up and broken down. The cameras are connected to 3 frame grabbers that are installed on a dual-core Xeon 3GHz PC. Figure 2 shows the sample views from each camera.

Figure 1. The hardware setup used to record the training session. The 3 PAL cameras are elevated at the pool side on tripods.

Figure 2. Sample views from the three PAL cameras.

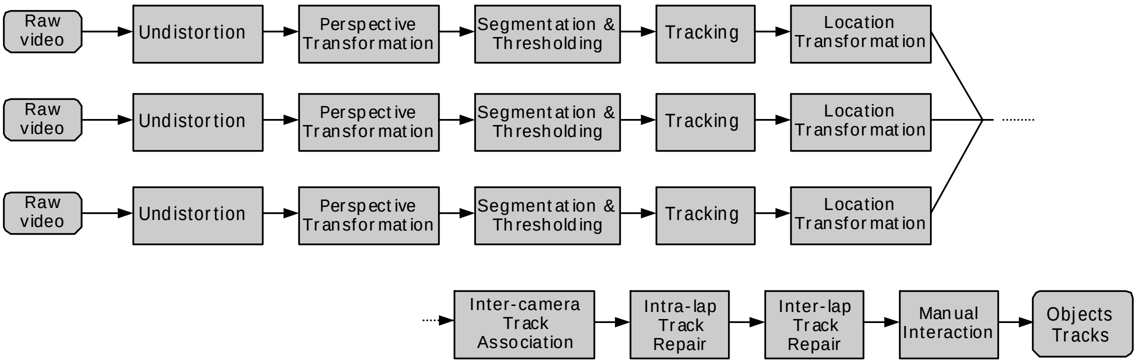


Figure 3. Block diagram of the data processing. The system uses the raw recording videos as input and produces object tracks as output.

The processing block diagram can be seen in Figure 3. The recordings made during data acquisition are the input of the processing. At the end, a list of objects tracks, which represent the swimmers laps, will be generated. The following list gives a short description of what each block does:

- *Undistortion:* This step corrects the distortion introduced by the lens. The distorted input image (Figure 2, left) is transformed to a rectified image (Figure 4, left). The camera distortion parameters are estimated beforehand using images of a calibration checkerboard in six orientations and the MATLAB Calibration Toolbox (Bouguet, 2006).

- *Perspective Transformation:* Perspective transformation is applied on sub-region of the image (e.g. yellow rectangle in Figure 4, left). This sub-region is defined manually for each camera and is specific to a given setup. The transform regularizes the geometry of the view and gives the impression of a view from directly above the pool (Figure 4, right).

- *Segmentation & Thresholding:* This step is used to separate the foreground from the background (See Figure for the detailed block diagram). Firstly, each RGB pixel of the input image is transformed to the red chrominance (Cr) component of the YCbCr color space using the following relation:

  $$Cr = 128 - (0.4998R + 0.4185G + 0.08128B)$$

  Since the water has strong blue component and the swimmers have strong red component this transformation enhances the distance between the color of swimmers and the color of water. After this transformation, a background model B_t is built and updated continuously by taking the running average of image frames:

$$B_t = (1-\alpha)B_{t-1} + \alpha I_t$$

with I_t as the current frame and α the update rate (how fast the model forgets previous frames). Finally, the image difference between the current image and background model image is then transformed to binary image by using hysteresis thresholding (Davies, 2005) to obtain groups of pixels (blobs), which represents the swimmers. The low and high intensity thresholds were determined by visual inspection of the binary output image and set to 10 and 16 respectively. Only blobs containing more than 60 pixels are admitted to the output, because smaller blobs are likely to be the result of noise caused by water movements and so on.

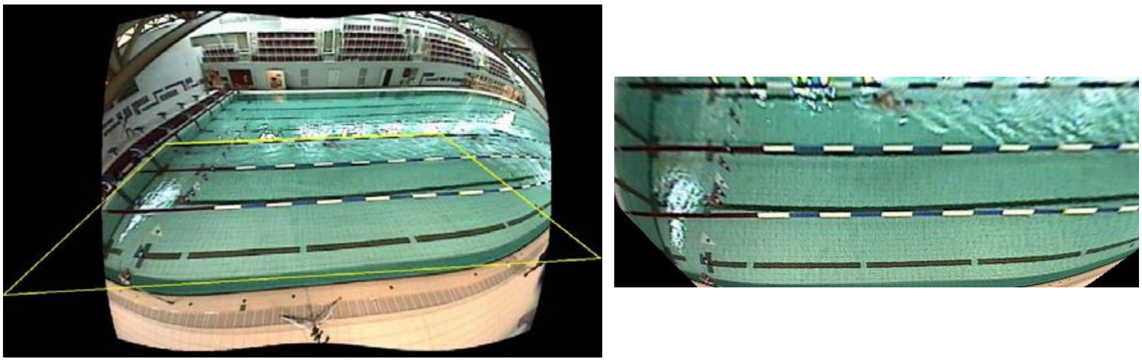
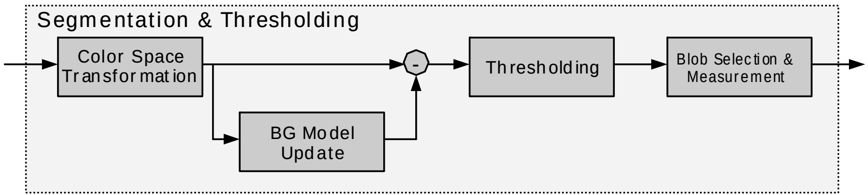


Figure 4. Perspective transformation from undistorted image. The region marked with the yellow rectangle on



the left image is transformed to the right image. The rest of the image is discarded.

Figure 5. Detailed block diagram of the Segmentation & Tracking block.

- *Tracking*: This step associates blobs from previous video frames with the current one (Figure 6) to form tracks. If blobs from the current frame and the previous overlap they are associated. The interaction handling is used to deal with the cases when more than one association is possible. For example, when swimmers approach each other the segmentation step may detect a single blob containing multiple swimmers. This type of interaction is handled by considering the existing individual swimmer's tracks as lost (no association possible) and creating a new track for the combined blob. When the single blob splits again into two separate blobs, the track from the single blob is considered as lost and two new tracks are generated for each swimmer. If an existing track cannot be associated to a blob in the current frame (i.e. lost track), linear extrapolation of the blob motion is used to predict the blob position and associations to the predicted positions are possible for up to 10 frames, after which it will be removed from the active track list and passed on to the next processing step.

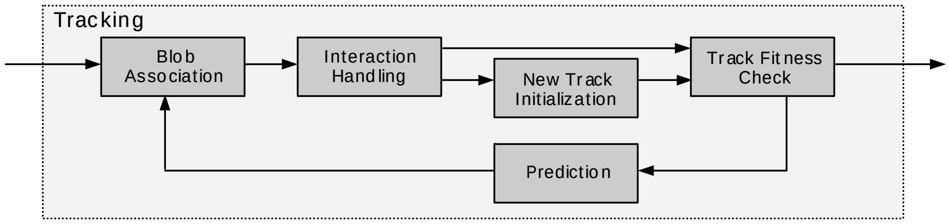
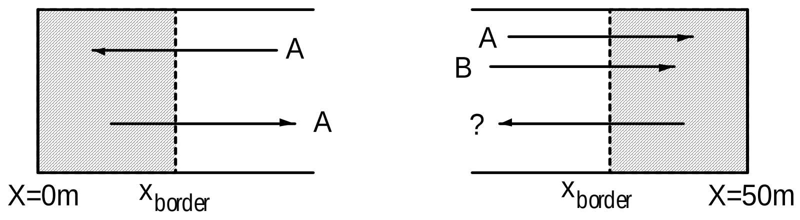


Figure 6. Detailed block diagram of the Tracking block.

- *Location Transformation:* This step performs a conversion from pixel coordinates to real world coordinates (i.e. meters). Since we know how long and how wide the pool is (the lane width is an official standard and the length can be determined by counting the stripes of the lane dividers), the pixel-to-world relationship can be calculated.
- *Inter-camera Track Association:* The track information from the multiple video streams is combined in this step. In the overlapping region between two camera views, tracks of swimmers leaving one view and entering an adjacent view are combined by associating tracks with the best degree of overlap. However, tracks observed to be traveling in opposite directions are not considered for association. This constraint is used because we know that swimmers should not turn back in the middle of the pool.
- *Intra-lap Track Repair:* Within a lap the track from a single swimmer may be fragmented (i.e. broken down into several short tracks). These track fragments can be grouped together using the following equations:

$$s_{i+1} = \arg\min_{s_j} d\left(\bar{s}'_{i,end}, \bar{s}_{j,start}\right), \quad \bar{s}_i = [x, y]^T, \;\; dir(s_i) = dir(s_j)$$

  where $d\left(\bar{s}_i, \bar{s}_j\right)$ is the Euclidean distance between two points, $\bar{s}_i$ the starting or end point of track s_i in 2D and $dir\left(s_i\right)$ is the motion direction of the specified track (either left or right). For each fragmented track s_i, the next track $s_{i+1} = s_j$ with the same direction of motion and minimum 2D Euclidean distance between the extrapolated end point of s_i' and the starting point of s_j is found. The end point of s_i is extrapolated to the candidate track starting point s_j in order to select the most probable candidate if there is a large gap between the current track and the next candidate track. Once a whole lap is recovered, the gaps between fragmented tracks are filled using linear interpolation.
- *Inter-lap Track Repair:* During the training session swimmers reaching the end of the pool either turn and start a new lap (usually performing a tumble turn), or take a rest. To maintain the track of a swimmer between laps the following rule is introduced. Any track that enters a region 5 meters from the end of the pool is associated to the next track that leaves that region as long as no other track enters the region in the meantime. When multiple tracks enter the region before any tracks leave it is not possible to maintain the track automatically (see Figure 7). These events are recognized automatically and require manual correction.



(a) (b)

Figure 7. The scheme used to group laps belonging to each swimmer in a lane, at the ends of the pool. (a) If one swimmer enters the region and some time later leaves and no other swimmers are resting, then the two laps can be easily grouped. (b) When multiple swimmers enter the region and some time later one swimmer leaves, two possibilities arise. Either swimmer A left first or swimmer B.

- *Manual Interaction:* The last step is to perform inter-lap track repair manually for those tracks that cannot be automatically associated. Since no user interface has yet been implemented this is done by inspecting the video and editing the results file.

Performance evaluation is done using two measures: frame-based metrics and object-based metrics (Bashir & Porikli, 2006). Firstly, a ground truth set is constructed and compared to the automatic detections in selected frames. Secondly, the individual tracks of objects are analyzed as separate entities. The ground truth set of swimmer locations was created using the ViPER tool (Doermann et al., 2000). We created a sparse set with an interval of 100 frames as a trade-off between amount of work needed and the reliability of the ground truth set. A bounding box covering the head and torso is used to represent the location of a swimmer. The arms and legs are excluded because they are underwater frequently, which make them difficult to use as a reliable boundary. Accurate lap times cannot be measured using a sparse ground truth so the start and end times for each lap were also noted: the start and end times are when the swimmers break or make contact with the pool wall, either by using their arms or feet.

Frame-based metrics measure how well the swimmers are detected in each frame. First, the following standard quantities were measured:

- True Positive (TP): The number of bounding boxes for which both ground truth data and system results agree on the presence of a swimmer.
- False Positive (FP): The number of bounding boxes for which the system falsely detects a swimmer.
- False Negative (FN)**:** The number of bounding boxes for which the system misses a swimmer.

In the above definitions, the matching between the ground truth data and the system results is done by looking at the overlap between the two. If the bounding box centre of the tracking results is located inside a ground truth bounding box, it is considered a match. Furthermore, the following metrics are derived from the above definitions:

- Tracker Detection Rate = TP / (FN + TP)
- False Alarm Rate = FP / (TP + FP)

These metrics were calculated for three cases: when we consider all bounding boxes, when we leave out bounding boxes in the ends of the pool (i.e. the first 5 meter and last 5 meter), and finally do the latter again for each separate lane.

Object-based metrics are based on the complete laps of each swimmer. The failure rate of the algorithm is measured by counting the number of manual interactions needed to identify every lap of a given swimmer as a single, continuous track. A single interaction is defined as the connection of one track fragment to another. Note that this definition does not include the number of interactions required to decide which fragments to connect. For an operational system a user could interact with the system to connect two track fragments using a maximum of two mouse clicks (selecting the end point of one track and the starting point of the other track) but examination of the video clip may be needed to decide which fragments to connect. After the complete laps have been identified for each swimmer, with manual correction if necessary, the measured lap times are compared to the ground truth.

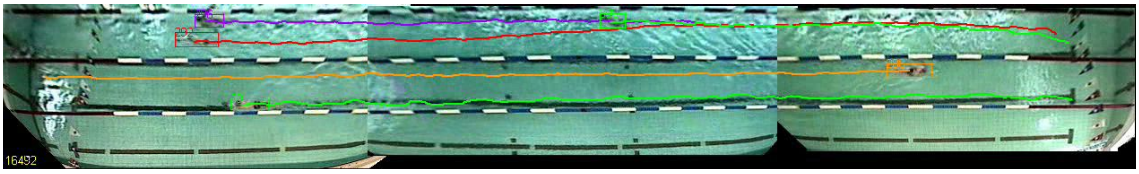


Figure 8. The visualization output of the tracking result. Each swimmer is represented by a bounding box with its track over time.

Table 1. The detection performance of our tracking method measured by the frame-based metrics.

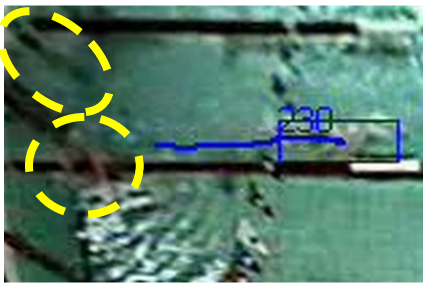
	# True Positives	# False Positives	# False Negatives	Tracker Detection Ratio	False Alarm Rate
All lanes	3328	250	1153	0,74	0,07
All lanes excluding pool ends	3094	250	199	0,94	0,07
Lane 1 (lower) excluding pool ends	238	22	23	0,91	0,08
Lane 2 (middle) excluding pool ends	1023	25	54	0,95	0,02
Lane 3 (upper) excluding pool ends	1819	220	132	0,93	0,11

Results & Discussion

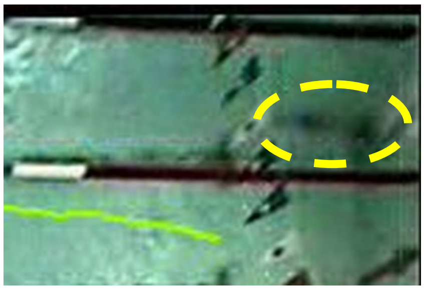
Setting up the mobile data acquisition system took about 40 minutes. Care had to be taken to position the cameras correctly in order to achieve the right degree of overlap between the 3 views. During data processing a total of 12 points in the 3 views had to be selected manually to register the images correctly. The prototype contains no mechanism to ensure that the camera position and viewing are repeatable between training sessions and within training sessions the camera positions can move if the tripods or the connecting wires are accidentally displaced, forcing recalibration of the system. In general, the more variable the camera and scene geometry is the higher the burden on the computer vision algorithm and these factors favor the use of multiple fixed acquisition systems at different locations rather than a single mobile system.

Figure 8 shows an example of the tracking algorithm output. The detection performance measured for 45 minutes of video using frame-based metrics is presented in Table 1. For all swimmers in all lanes the tracker detection ratio is 0.74 and the false alarm rate is 0.07. Many of the false negatives are caused by the swimmers at the ends of the pool: when bounding boxes within 5 meters from the pool ends are excluded from the analysis the detection ratio improves to 0.94 and the false alarm rate remains 0.07. The reason for this is that swimmers at the ends of the pool are often resting or underwater, meaning that their motion or color properties are less distinguishable from their environment. If the pool ends are excluded and the different lanes are compared, the best results are obtained from lane 2. In lane 3 the relatively high false alarm rate is due to bright reflections, caused by light entering from a window on the other side of the pool, that are sometimes mistaken for swimmers. The lower tracker detection ratio in lane 1 is due to imperfect calibration of the setup which causes the swimmer to partly fall outside the view and be undetected by the prototype system.

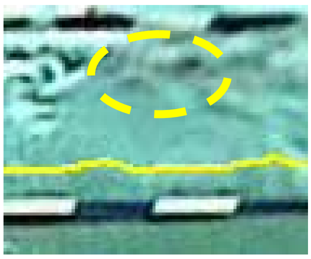
Figure 9 shows examples of 4 failure modes for the detection algorithm. Many of the false negatives reported above are caused by a failure to detect swimmers at the ends of the pool when they are resting (Figure 9a) or underwater (Figure 9b). In addition, water waves can cause false negatives by covering the swimmer (Figure 9c) and water reflections can cause false positives (Figure 9d).



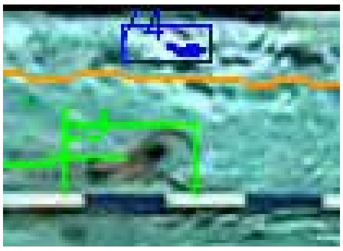
(a)



(b)



(c)



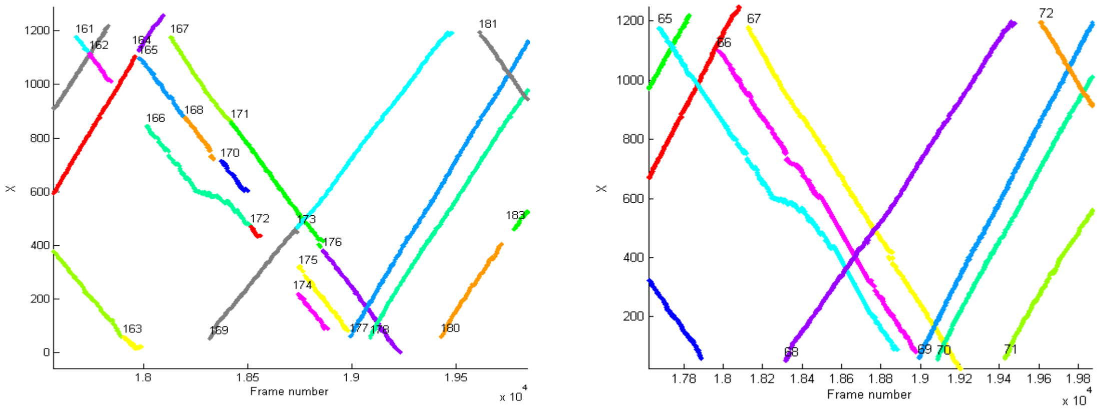
(d)

Figure 9. The types of difficulty encountered during the whole training registration process (best viewed in color): (a) Resting swimmers and (b) swimmers underwater could not be detected (shown here with yellow dashed circles), (c) water reflections and water ripples caused misdetection (undetected swimmer within yellow dashed circle) and (d) false detection (upper rectangle).

Table 2. Comparison of the number of manual interactions needed in three stages during the track repair method.

		Number of laps in Ground Truth set	Number of manual interactions needed		
			Before track repair	After grouping fragmented tracks	After grouping laps to swimmers
Lane 1	Swimmer 1	23	75	22	0
Lane 2	Swimmer 2	54	131	53	6
	Swimmer 3	51	57	50	6
Lane 3	Swimmer 4	52	132	51	27
	Swimmer 5	53	98	52	28
	Swimmer 6	58	104	57	17
	Swimmer 7	47	99	46	17
	Total	338	696	331	101

Table 2 shows the number of manual interactions needed to correctly identify the complete laps of each swimmer. Before track repair an average of 100 manual interactions per swimmer would be required to correct the results because the tracks of individual laps are fragmented. After automatically grouping fragmented tracks each lap is represented by a single track, but laps have not been associated to swimmers (See Figure 10). After grouping laps to swimmers the system is not able to solve instances when two or more swimmers are resting at the same end of the pool. These situations are recognized by the algorithm but have to be resolved manually. They never occur in lane 1 which is occupied by a solitary swimmer and often occur in lane 3 which is shared by four swimmers. After all tracking methods have been applied the number of manual interactions required for correct swimmer tracking is at best none and at worst 28 depending on which swimmer is tracked. Note that trials showed that no manual interactions were necessary to accurately track the swimmers in lane 2 if the algorithm made use of the fact that each swimmer used only one half of the lane and the vertical position was used for inter-lap track repair. For elite athletes this constraint usually applies because it most closely matches race behavior.



(a) (b)

Figure 10. (a) The visualization of the fragmented tracks (best viewed in color). Each line segment is a collection of centre of gravity points of the detected swimmer blobs. The segments are drawn here using only the x coordinate of the centre points and the frame numbers (time axis). Upward lines in time represent swimmers going from left to right. In the matching, the y coordinate is also considered. For example, segment 172 (red) should be matched to segment 174 (magenta). Without using the extrapolation, segment 172 would have been matched to segment 175 (yellow), which has the same direction and is the closest. (b) The results after grouping the fragmented tracks. The gaps between tracks are now filled with interpolated values.

Table 3. Results from the object-based evaluation: statistics of the elapsed time discrepancy and the average speed discrepancy.

	Elapsed time discrepancy (after repair)	
	Mean (sec)	Standard Deviation (sec)
Swimmer 1	2,62	1,61
Swimmer 2	1,69	2,33
Swimmer 3	0,52	0,60
Swimmer 4	2,04	2,08
Swimmer 5	1,29	2,35
Swimmer 6	0,43	0,74
Swimmer 7	1,57	1,74

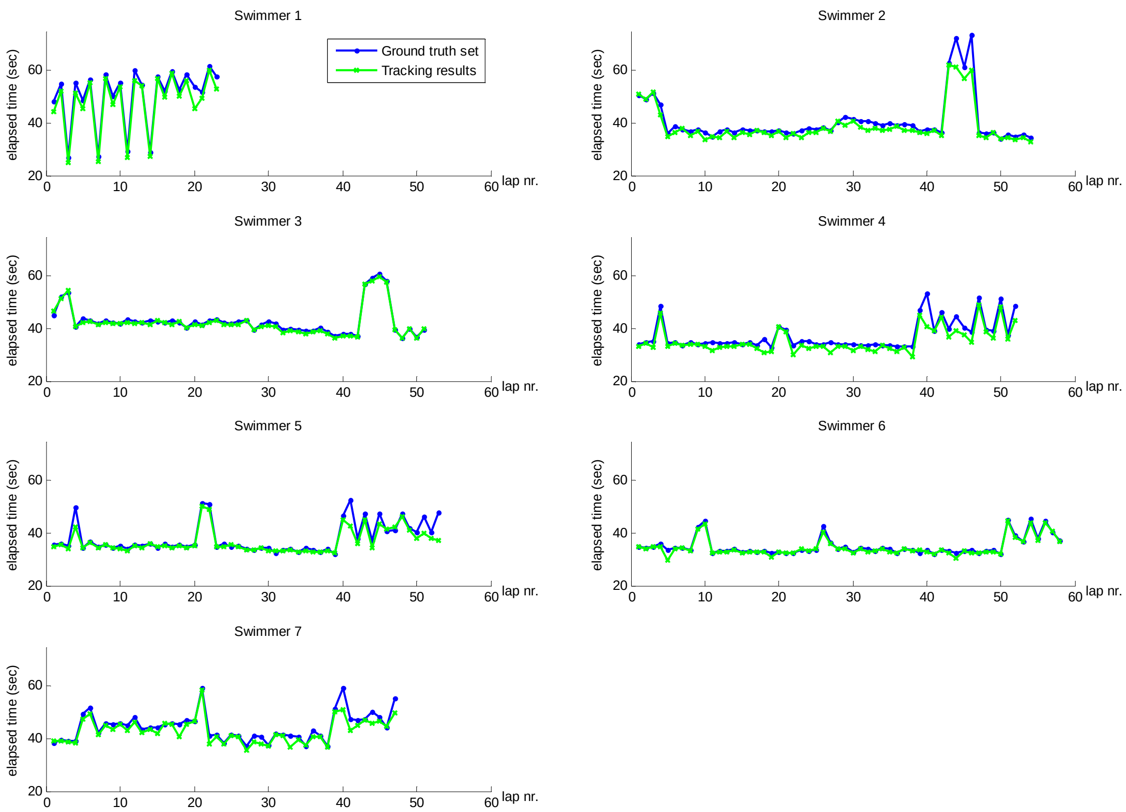


Figure 11. The elapsed time plots for all laps of all swimmers. The ground truth lap times (blue) are compared to the tracking results (green) after manual correction. Patterns dictated by the training schedule can be discerned in these plots, for example, swimmer 1 swims with a pattern of a few slow laps and 1 fast lap during the first 15 laps.

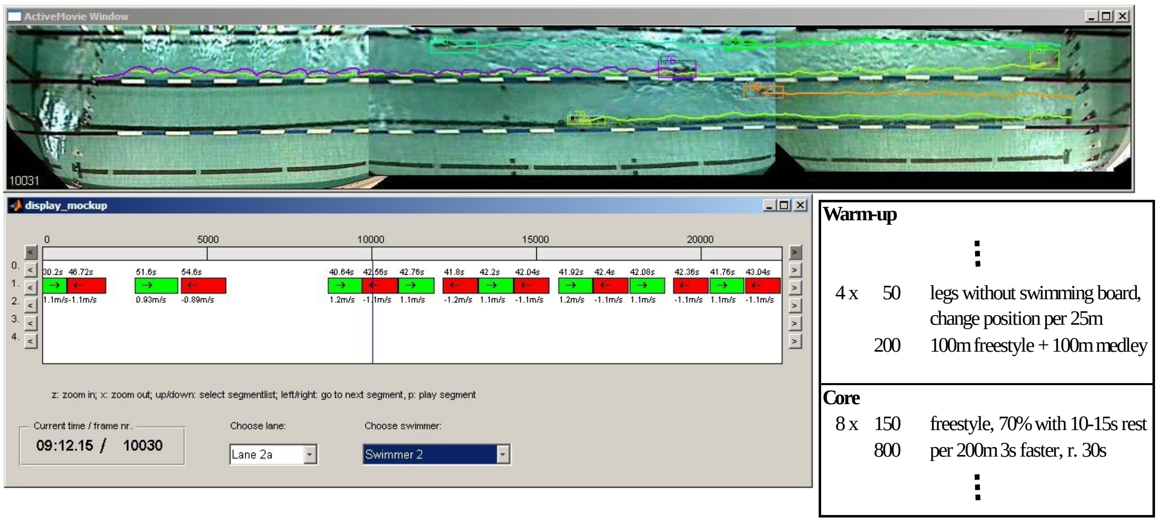


Figure 12. Comparison of the registered laps of a swimmer to the training schedule. This part shows the performance of swimmer 3 (upper halve middle lane) during the transition between warm-up and core training. Note that each segment in the visualization represents a 50m lap and the swim direction is indicated by different segment colors.

The lap times of all swimmers are depicted in Figure 11. The mean and the standard deviation of the measurement discrepancies are summarized in Table 3. The measurement discrepancy is defined as the estimated value subtracted from the ground truth value. The mean discrepancy for the lap time measurements is typically 4% of the lap time. It can be seen from Table 3 that the automatically-measured lap times are systematically shorter than the true lap times, sometimes with a significant difference when swimmers move slowly. This is mainly caused by the failure of the algorithm to initialize tracks at the moment the swimmers start a lap. At the start of a lap, the swimmers are often underwater after completing a tumble-turn and may not be detected until they have moved several meters from the pool wall and resurfaced (see Figure 8). Trials indicated that linearly extrapolating the tracks to the pool edge using the average speed of the first 4 seconds of each track improved the lap time estimates but that linear interpolation is not optimal because swimmers show a tendency to go faster at the beginning of a lap.

To allow a coach to check that a swimmer has completed a training session as instructed it is necessary to map the written training schedule to the system output. It has been found that the registered laps show patterns that can make this possible and this issue will be the subject of future work. Figure 12 shows an example of a swimmer swimming the last 2x100m of a warm-up, taking a rest, and beginning the core training by swimming groups of 150m (3 laps) with 10-15 seconds rest in between.

Conclusions and Future Work

We have described results from a prototype system designed to register the training sessions of swimmers using video recordings. These results allow us to draw conclusions about design requirements for future systems and likely performance limitations.

Regarding video acquisition hardware we have observed that the mobile prototype requires time and specialist knowledge to assemble and that changes in the scene and camera geometry will be source of variability in the system performance. We conclude that fixed systems are likely to offer advantages over mobile systems in terms of data quality and repeatability of results. For mobile systems a simpler set-up with fewer cameras and easier assembly and calibration would be advantageous.

Regarding computer vision techniques we have shown that, although joining laps is difficult, it is possible to track each lap swum by 7 swimmers taking part in a 45 minute training session automatically using a simple background modeling and blob tracking method. Many difficulties encountered during tracking, such as intra-lap track fragmentation and inter-lap track association can be corrected by using constraints about the swimmers motions, which are predictable and repetitive. The main limitation on the algorithm performance is the failure to detect swimmers at the ends of the pool that are either resting or underwater. The failure to detect resting swimmers means that manual interactions are necessary to distinguish between swimmers who take a rest together at the same end of the pool. The failure to detect swimmers swimming underwater after tumble turns causes lap times to be systematically underestimated by typically 1.5 seconds. The tracking output shows patterns of laps and pauses that can make it possible to manually map the written training schedule of the coach to the system output to obtain an overview of the training session as performed.

Our results show that the performance of the system can be highly dependent on the way in which the swimmers behave and how the training sessions are organized. For example, much more manual interaction is necessary to correct the tracking results when swimmers rest together at the same end of the pool or share the same swim lane and swim in a circulating pattern rather than swimming backwards and forwards in the same part of the lane. One way to improve the performance of the system without further algorithm development is for the training sessions to be organized in a way that avoids these situations and develop guidelines for the coaches and swimmers. The use of colored swim caps could also improve the segmentation performance at the ends of the pool and allow swimmer positions and identities to be tracked more accurately.

Acknowledgements

This research is supported by the BSIK-project MultimediaN and is made possible by the collaboration with NOC*NSF, the Dutch Olympic Committee and National Sport Federation. The authors would like to thank the reviewers for their useful comments and suggestions.

References

Bashir, F. & Porikli, F. (2006). Performance Evaluation of Object Detection and Tracking Systems. *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 7-14.

Baumberg, A.M. & Hogg, D. (1995). An Adaptive Eigenshape Model. *Proceedings of the Sixth British Machine Vision Conference, vol. 1*, 87-96.

Bouguet, J.-Y. (2006). Camera Calibration Toolbox for MATLAB, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html [02.11.2006].

Dartfish (2006), Dartfish Video Software Solutions, http://www.dartfish.com [02.11.2006].

Davies, E.R. (2005). *Machine Vision: Theory, Algorithms, Practicalities*. San Fransisco: Morgan Kaufmann Publishers.

Doermann, D. & Mihalcik, D. (2000). Tools and Techniques for Video Performance Evaluation. *Proceedings of the International Conference on Pattern Recognition*, 4167-4170. http://sourceforge.net/projects/viper-toolkit/ [02.11.2006].

Eng, H.-L., Wang, J., Wah, A.H.K.S. & Yau, W.-Y. (2006). Robust Human Detection within a Highly Dynamic Aquatic Environment in Real Time. *IEEE Transactions on Image Processing*, *15(6)*, 1583-1600.

Elgammal, A., Duraiswami, R., Harwood, D. & Davis, L.S. (2002). Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance. *Proceedings of the IEEE*, *90(7)*, 1151-1163.

Gavrila, D.M. & Philomin, V. (1999). Real-time Object Detection for Smart Vehicles. *Proceedings of IEEE International Conference on Computer Vision*, 87-93.

Hu, W., Tan, T., Wang, L. & Maybank, S. (2004). A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, *34(3)*, 334-352.

Nguyen, H.T. & Smeulders, A.W.M. (2004). Tracking Aspects of the Foreground against the Background. *Proceedings of European Conference on Computer Vision*, 446-456, LNCS 3022.

Perše, M., Kristan, M., Perš, J., & Kovačič, S. (2006). A Template-based Multi-Player Action Recognition of the Basketball Game. *Proceedings of ECCV Workshop on Computer Vision Based Analysis in Sports Environments*, 71-82.

Stauffer, C. & Grimson, W.E.L. (1999). Adaptive Background Mixture Models For Real-time Tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 245-252.

STR (Swimming Technology Research) (2006). Aquanex for Improving Performance, http://swimmingtechnology.com/aquanex.htm [02.11.2006].

Urtasun, R., Fleet, J.F. & Fua, P. (2005). Monocular 3–D Tracking of the Golf Swing. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 932-938.

Viola, P., Jones, M.J. & Snow, D. (2005). Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, *63(2)*, 153-161.

Wang, J.R. & Parameswaran, N. (2004). Survey of Sports Video Analysis: Research Issues and Applications. *Proceedings of Pan-Sydney Area Workshop on Visual Information Processing*, *36*, 87-90.

Xu, M., Orwell, J. & Jones, G. (2004). Tracking Football Players with Multiple Cameras. *Proceedings of IEEE International Conference on Image Processing*, 2909-2912.