Sensor Fusion for Video Surveillance

Lauro Snidaro, Gian Luca Foresti
Dept. Mathematics and Computer Science
University of Udine
via delle Scienze, 206, 33100 Udine
ITALY
{snidaro, foresti}@dimi.uniud.it

Ruixin Niu, Pramod K. Varshney
Dept. of Electrical Engineering and Computer Science
Syracuse University
Syracuse, NY 13244
USA
{rniu, varshney}@syr.edu

Abstract – *In this paper, a multisensor data fusion system for object tracking is presented. It is able to track in real-time multiple targets in outdoor environments. The system can take advantage of the redundant information coming from different sensors monitoring the same scene. The measurements (positions of the targets) obtained from the available sources are fused together to obtain a more accurate estimate. Data fusion is performed considering sensor reliability at every time instant. A confidence measure has been employed to weight sensor data in the fusion process. Compared to single camera systems, the adopted approach has produced more accurate and continuous trajectories, reducing calibration and segmentation errors.*

Keywords: Surveillance systems, multisensor data fusion, object detection.

1 Introduction

The recent events of September 11, 2001, have demonstrated that there is a need for improving surveillance capabilities of public areas (e.g., airports, metro or railway stations, parking lots, tunnels or bridges, etc.) in order to prevent terrorist acts. Advanced multisensor surveillance systems represent a possible answer to prevent terrorist attacks by enhancing monitoring and control capabilities of remote human operators on large environments. Such systems can perform real-time intrusion detection and/or suspicious event detection in complex environments. New generation surveillance systems [1, 2, 3, 4], that have to manage large amounts of visual data (optical, infrared, etc.), and the recent development of sensor technology and computer networks have contributed to an increasing interest in distributed systems for real-time information fusion [5, 6, 7].

In this paper, a system that integrates optical and infrared (IR) sensors to support 24 hours per day real-time visual-based surveillance of outdoor environments, is proposed. IR and optical sensors are at the first level of the proposed architecture. Video signals of each physical sensor are first processed to extract moving image regions, called blobs [8], and features are computed for target tracking, classification, and data fusion procedures. This integration allows to improve the accuracy of object localization at higher levels, which is based on the ground plane hypothesis [8] and object recognition [9]. At the first level, specialized Processing Nodes (PN) track each detected blob on the image plane and transform 2D blob positions (in the sensor coordinates system) into 3D object positions (in the coordinates of the monitored environment's map). Each first level PN is committed to the surveillance of a sub-area of the monitored environment.

In surveillance systems, target tracking is of paramount importance. The user is generally interested in estimating the position and velocity of the objects in the scene, as well as their trajectory. The estimates are affected by two kinds of noise: process noise and measurement noise. The former is due to the uncertainty of the model that describes the motion of the objects in the scene. Target trajectories are in fact not completely predictable and sometimes targets are deliberately non-cooperative and maneuvering in an unpredictable manner (military and surveillance applications). However, filtering techniques require the definition of some model in order to predict the next target's state: for this reason the actual measurements can differ substantially from predictions.

Measurement noise is primarily caused by the acquisition process and by coordinates transformation algorithms (if present). This noise can severely affect the observation of the current state of a target, therefore also affecting the following prediction phase.

While process noise can be reduced by adopting multi-model filtering techniques like the IMM-estimator [10], measurement noise is commonly tackled by adopting more accurate sensors. However, recent advancements in cameras and processing technology have made the multi-sensor solution more desirable.

Greater system robustness and performance is in fact achievable with a suite of sensors and through data fusion techniques [11]. A well-known practice in radar applications [12], data fusion is now being considered for video-based systems [13]. Recent works have addressed the tracking of humans and vehicles with multiple sensors [14, 15, 16, 17]. The main hurdle of the additional computational requirements has been removed by the great processing power of today's CPUs. Moreover, intelligent sensors, able to perform on their own a great deal of the required computation, are also available.

Even though data fusion cannot squeeze increased performance out of a set of unreliable cameras [18], very interesting results can be attained adopting several standard ones.

A multi-sensor tracking system is here presented. It employs multiple video-cameras monitoring the same area. The general architecture is discussed in the following section. The system explicitly takes into account sensors' accuracy in the fusion process: a reliability factor is defined and described in Section 2.4. Preliminary experimental results are presented for two configurations: the first one involves homogeneous sensors (two color cameras), while in the second two heterogeneous sensors are employed (optical and infrared cameras) for monitoring an outdoor area.

2 Architecture and processing

The adopted architecture follows the guidelines of recent video-surveillance systems [1, 2]. It is composed of several static sensors and processing nodes (PNs) for each area of interest, as shown in Figure 1.

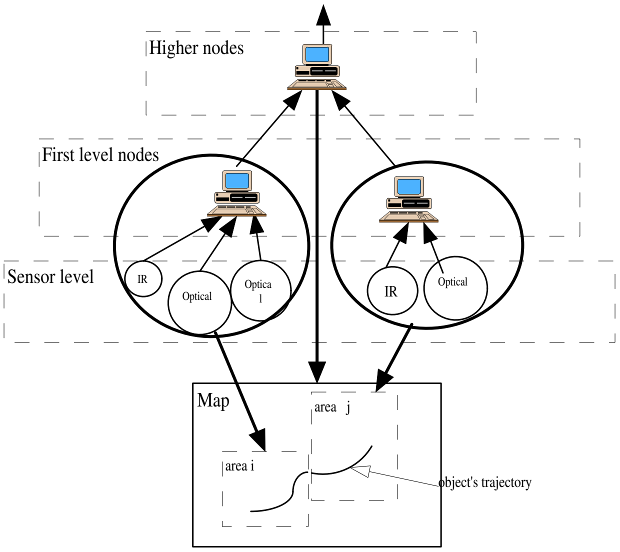

Fig. 1: Distributed architecture of the system

The sensors monitoring the same area are connected to the same PN which is responsible for the tracking of the objects in that area.

A PN is also in charge of all the image processing steps, if intelligent sensors are not available. In particular, they are responsible for running the algorithms needed to identify moving objects (through change detection) from each video source: a)image differencing; b) filtering; c) blob extraction. These are well-known in computer vision applications; a detailed description can be found in [19, 20, 21].

Blob extraction is the last low level processing step, which yields blobs of the moving objects in the scene. At this point, features (dimensions, area, centroid coordinates, etc.) can be extracted from each blob. These attributes are fundamental in the following processing phase: object tracking.

2.1 Blob extraction

This processing step occurs at sensor level and pinpoints moving regions in the image through change detection algorithms [1, 2, 19, 20, 22]. Motion detection and blob extraction is exploited following a layered background subtraction approach [23]. Change detection is performed using an algorithm for automatic threshold computation based on Euler numbers [24]. The background is updated using a Kalman filter [21]. Frame by frame subtraction is also applied in conjunction to improve detection results [23]. Morphological filters are also applied to improve the quality of the extracted blobs by removing spurious pixels due to noise, and by enhancing the regions' connectivity [23].

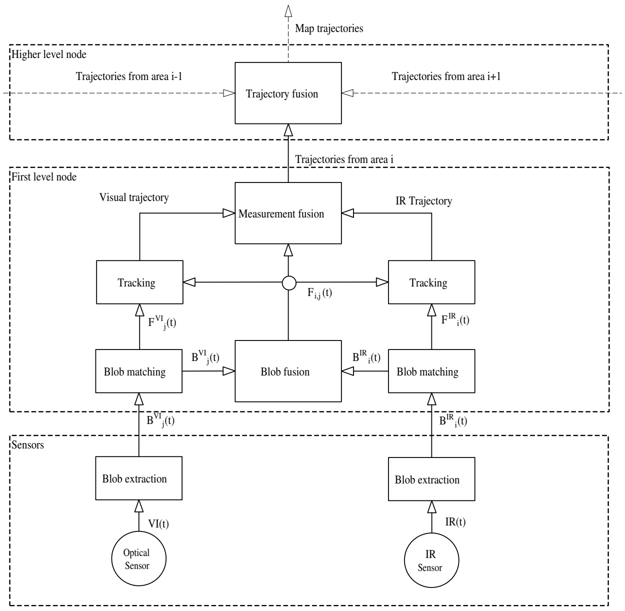

Fig. 2: Processing steps.

Following the scheme in Figure 2, at time t, images $\mathbf{VI}(t)$ and $\mathbf{IR}(t)$ are produced respectively by the optical and infrared sensor. Each sensor applies the blob extraction procedure thus obtaining the arrays $\mathbf{B^{VI}}(t)$ and $\mathbf{B^{IR}}(t)$ containing the blobs extracted on the current frame at time t respectively by the optical and the infrared sensor.

In Figure 9, two cameras (color and infrared) are monitoring from the same location the same area in the presence of fog and low illumination. The color camera obviously performs poorly in this situation, as can be seen in the first row of pictures of Fig. 9. This influences the blob extraction procedure as can be seen in the second row of images. The silhouette of the walking person is not extracted correctly. The IR camera instead gives a more useful video signal (third row of images) as the person is clearly visible in the scene. In the processed frames, the blob is correctly extracted (last row of Figure 9).

2.2 Target tracking

In a video surveillance system, usually multiple objects exist in the scene. For example, in a parking lot during daytime, there could be many objects, such as people and vehicles, that move around.

The system needs to maintain tracks for all objects simultaneously. Hence, this is a typical multi-sensor multi-target tracking problem: measurements should be correctly assigned to their associated target tracks and a target's associated measurements from different sensors should be fused to obtain better estimation of the target state.

A first tracking procedure occurs locally to each image plane, detected moving regions (blobs) have to be matched with the objects present in the previous frame (at the previous time instant). For each sensor, image processing is performed to extract blobs as indicated in Section 2.1. The system then executes an association algorithm to match the current detected blobs with those extracted in the previous frame. A number of techniques are available, spanning from template matching, to features matching [23], to more sophisticated approaches [25]. The approach used in our system was twofold, exploiting the Meanshift predictions [25] and matching blob features (Hu moments, base/height ratio, etc.).

To perform data fusion, a common reference frame is needed, and the sensors have to be registrated on it. Generally, a 2D top view map of the monitored environment is taken as a common coordinates system [8, 9], but even the GPS may be employed to globally pinpoint the targets [23]. The former approach is obviously more straightforward to implement, as a well-known result from projective geometry states that the correspondence between an image pixel and a planar surface is given by a planar homography [26, 27]. The pixel usually chosen to represent a blob and be transformed into map coordinates is the projection of the blob's centroid on the lower side of the bounding box [8, 9, 23].

Measurement gating and assignment is then performed. Each local procedure considers only the objects in the field of view of the corresponding sensor. For each object known at the previous time instant, only the measurements (objects' positions) falling within a gating distance (Fig. 3) are considered.

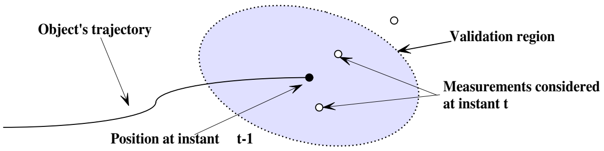

Fig. 3: Gating.

The Mahalanobis distance can be used to determine the validation region as in [28]. This step reduces the probability of erroneous associations due to noise. The measurements coming from each sensor, for a given object, falling within the gating region are fused together as described in Section 2.3.

To deal with the multi-target data assignment problem, especially in the presence of persistent interference, there are many matching algorithms available in the literature: Nearest Neighbor (NN), Joint Probabilistic Data Association (JPDA), Multiple Hypothesis Tracking (MHT), and S-D assignment. The choice depends on the particular application; detailed descriptions and examples can be found in [10, 29, 30].

The trajectory on the top-view map of every object is modeled through a linear Kalman filter, where the state vector $\hat{x} = (x, v_x, y, v_y)$ is constituted by the position and velocity of the object on the map. At every frame (the system processes 25 frames per second) a new measurement of the position is received.

In this paper, position estimates from different sensors are fused in a centralized fashion. Data fusion is performed considering sensor reliability at every time instant. A confidence measure has been employed to weight local estimates in the fusion process, as will be discussed later in Section 2.4.

2.3 The fusion process

Fusion is performed using a Kalman filter approach for the purpose of obtaining better position estimates of the observed objects. Two fusion schemes, shown in Figure 2.3, were considered during the experiments: measurement fusion and track-to-track fusion [31, 32, 33].

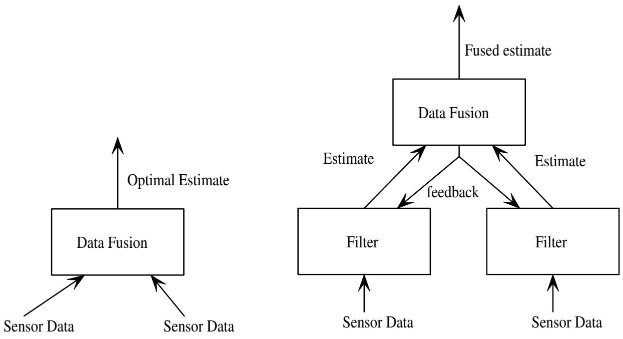

Fig. 4: Measurement and track-to-track fusion schemes.

The former scheme involves the fusion of the positions of the target (according to the different sensors) obtained right out of the coordinate conversion function, as can be seen in Figure 2.3(left). The latter performs the fusion of the local estimates, as shown in Figure 2.3(right). The measurement fusion algorithm is theoretically optimal, while the track-to-track fusion scheme has less computational requirements but is sub-optimal in nature [32].

However, when dealing with extremely noisy sensors (i.e. video sensors performing poorly due to low illumination conditions), the track-to-track scheme is generally preferred. Running a Kalman filter for each track to obtain a filtered estimate of the target's position allows the smoothing of high variations due to segmentation errors. The actual scheme employed was a track-to-track without feedback, the rationale given by computational constraints. In fact, during the experiments, the high frequency of the measurements and real-time requirements did not allow to take into account the feedback information.

The process, for each target, involves the following steps: (1) collection of measurements available from the local sensors; (2) grouping and assignment of the measurements to each target known at the previous time instant; (3) updating each target's state by feeding the associated filtered estimates to the fusion algorithm.

The fusion procedure maintains its own list of targets. Note that the second step is performed with the constraint that only a single measurement from a given sensor is to be associated with a single target in the list maintained by the fusion procedure.

To regulate the fusion process automatically according to the performance of the sensors, a confidence measure is presented in the following Section to weight local estimates.

2.4 Appearance and Appearance Ratio (AR)

Tracking accuracy can be improved through data fusion exploiting the redundancy given by the multiple-camera architecture. Fusing data collected from different sensors requires the determination of measurements' accuracy so that they can be fused in a weighted manner.

Making no distinction between measurements could lead to filter instability and erroneous estimates, especially in the presence of malfunctioning sensors. Hardware failures or unfavorable illumination conditions (e.g., optical camera used during night time) could yield very poor performance and generate segmentation errors (blobs partially extracted, noise, etc.). Therefore, considering every measurement with equal weight could fail to accomplish one of the data fusion's objectives: to not obtain a result worse than the one achievable with a single sensor [18].

The idea is to obtain from the Kalman filter a fused estimate more biased by accurate measurements and almost unaffected by inaccurate ones. Filter's responsiveness to measurements can be adjusted through the measurement error covariance matrix $\mathbf{R}$. If the eigenvalues of a particular matrix $\mathbf{R}$ are smaller than those of the other, the corresponding measurement will have a larger weight.

The following measure, called Appearence Ratio (AR), gives a value to the degree of confidence associated with the $j-th$ blob extracted at time t from the sensor s:

$$AR(\mathbf{B}^s_{j,t}) = \frac{\sum_{x,y \in \mathbf{B}^s_{j,t}} \mathbf{D}(x,y)}{|\mathbf{B}^s_{j,t}|c} \quad (1)$$

where $\mathbf{D}(x,y)$ is the difference map obtained as absolute difference between the current image and the reference one, and c is a normalization constant depending on the number of color tones used in the image. The AR is thus a real number ranging from 0 to 1 that gives an estimate of the level of performance of each sensor for each extracted blob. As can be seen in Figure 9, the AR values (reported below the bounding boxes) of the blobs extracted from the infrared sensor are considerably higher than those extracted from the optical one.

AR values are then used to regulate the measurement error covariance matrix to weight position data in the fusion process. The following function for the position measurement error has been developed:

$$r(\mathbf{B}^s_{j,t}) = GD^2(1 - AR(\mathbf{B}^s_{j,t})) \quad (2)$$

where GD is the gating distance. The function is therefore used to adjust the measurement position error so that the map positions calculated for blobs with high AR values are trusted more (i.e. the measurement error of the position is close to zero), while blobs poorly detected (low AR value) are trusted less (i.e. the measurement error equals the gating distance).

3 Results

Experiments with real video sequences have been carried out in order to test the performance of the proposed approach. Images taken from the first experiment are reported in Figure 5. Two color cameras have been employed to follow the movements of three persons walking in a courtyard. This daylight outdoor scene is simple for a vision tracking problem, but the purpose was to evaluate the accuracy of the trajectories, not the tracking itself. So the trajectories calculated by the single sensors were compared to the ground truth (markers were present on the ground) and the fusion approach.

The first row of Figure 5 shows images taken from the first sensor which was superior to quality than the second camera and proved to be more effective in detecting the walking persons. Even though the first sensor was monitoring the area with a configuration of the optics more wide-angled than the second sensor (thus detecting smaller blobs), it still performed slightly better. This is reflected by the AR values of the blobs in the second row which are generally greater than those in the fourth row. In this experiment the two sensors both performed reasonably well.

Figure 6 shows the trajectory of one of the persons in Figure 5 (indicated with an arrow) according to the first sensor, while Figure 7 reports the trajectory obtained by the second camera. The two sensors are reporting a track similar to the ground truth (black lines). Nonetheless, a better result is obtained through data fusion (Figure 8). The advantages are the following:

- the trajectory exploits the estimates of just one sensor when the other one is not giving readings (i.e. the target is out of the field of view, i.e. in Figure 5, first column, the person on the left in rows 1-2 is not present in the field of view of the second sensor, rows 3-4);

- the presence of two points of view can help disambiguate situations of partial or total occlusions (second column of Figure 5), therefore maintaining a correct and continuous tracking of the targets. Notice that the AR value was not computed for the blob detected by the first sensor since it was recognized as a compound object generated by an occlusion and therefore will not be associated to any of the three objects present at the previous time instant;

- there is an explicit weighting of the estimates in the fusion process through the AR to account for segmentation errors. Segmentation errors translate into trajectory errors. In the third column of Figure 5 the person in the center of the scene is half concealed by a small tree: the second sensor is not giving a proper detection and gets a low AR score for that blob.

- Data fusion reduces camera calibration errors (due to the homographic transformation from image pixels to

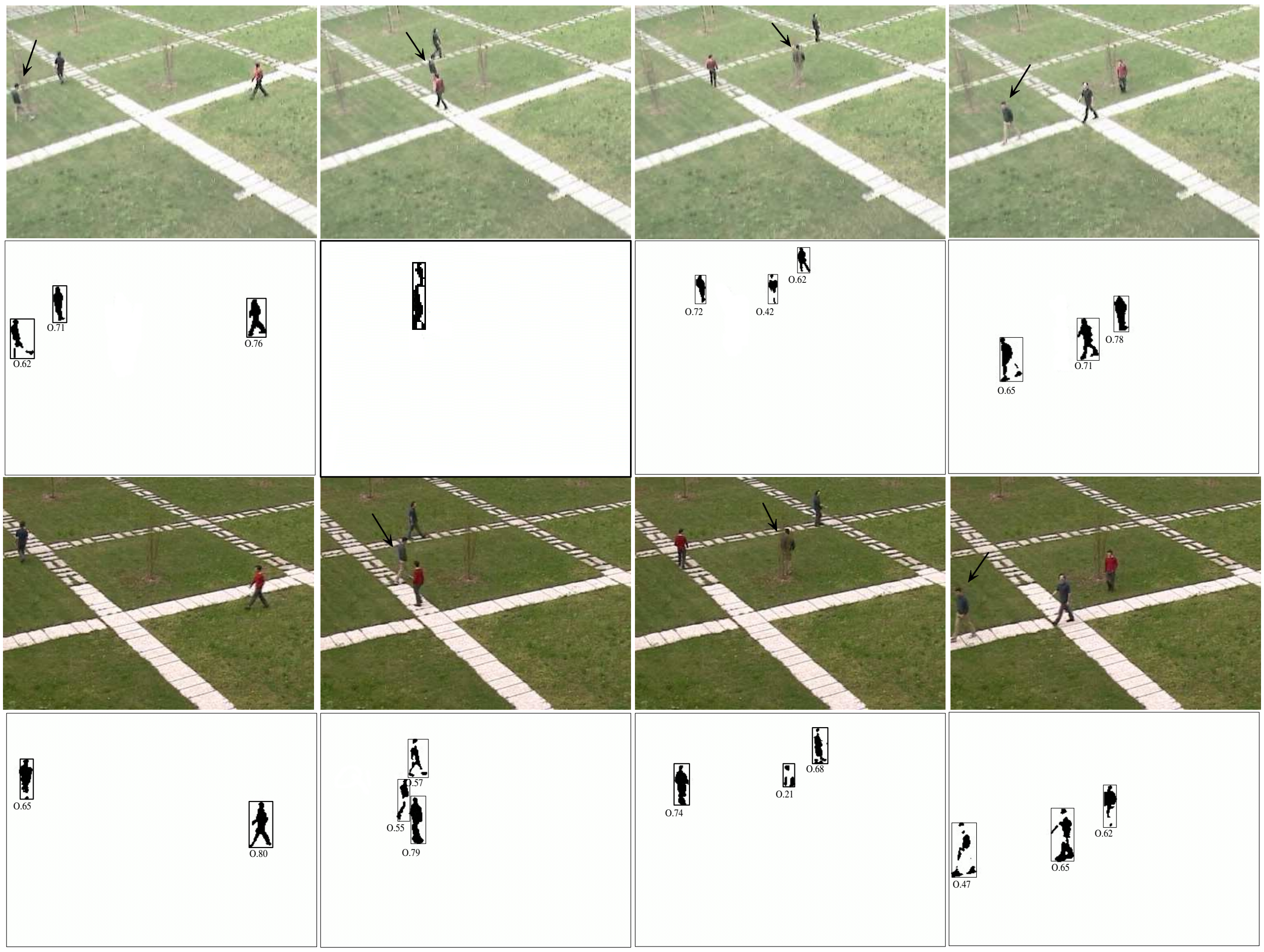

Fig. 5: Images and blobs from the courtyard sequence. AR values are indicated below each blob.

map points, Section 2.2). The first sensor gives better segmentation results, but, due to the wide-angle setup of the optics, camera calibration errors are more probable. So the fused data is more weighted on the first sensor due better video performance, but takes also into account the second sensor which suffers less from calibration errors.

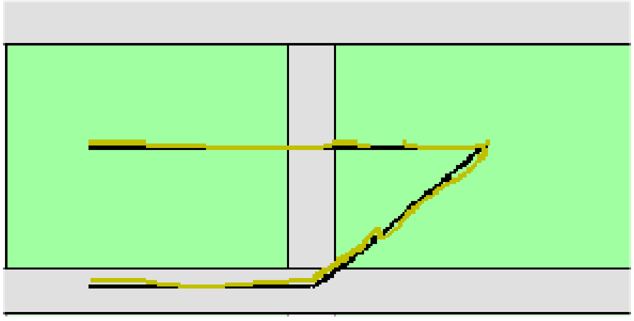

Fig. 6: Trajectory according to the first color sensor.

The performances of the two sensors and of data fusion are summarized in Table 2 where are reported the mean and standard deviation of the distance (in pixels, 1 pixel ≈ 10 cm) between measured and ground truth positions on the map of the walking person indicated by an arrow in Figure

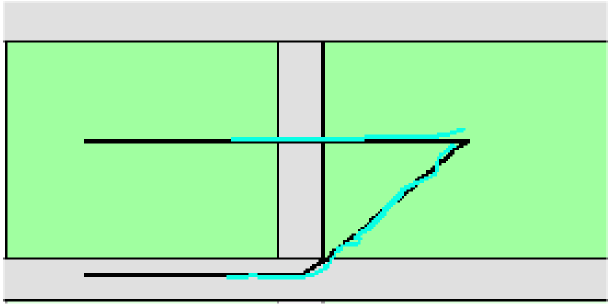

Fig. 7: Trajectory according to the second color sensor.

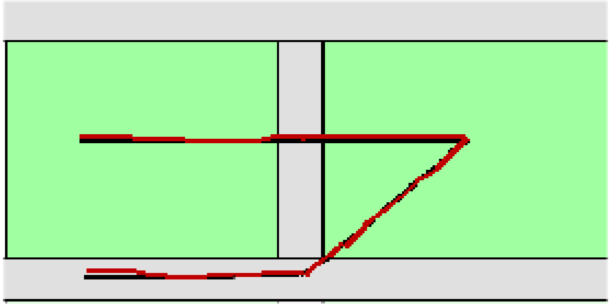

Fig. 8: Trajectory obtained through data fusion.

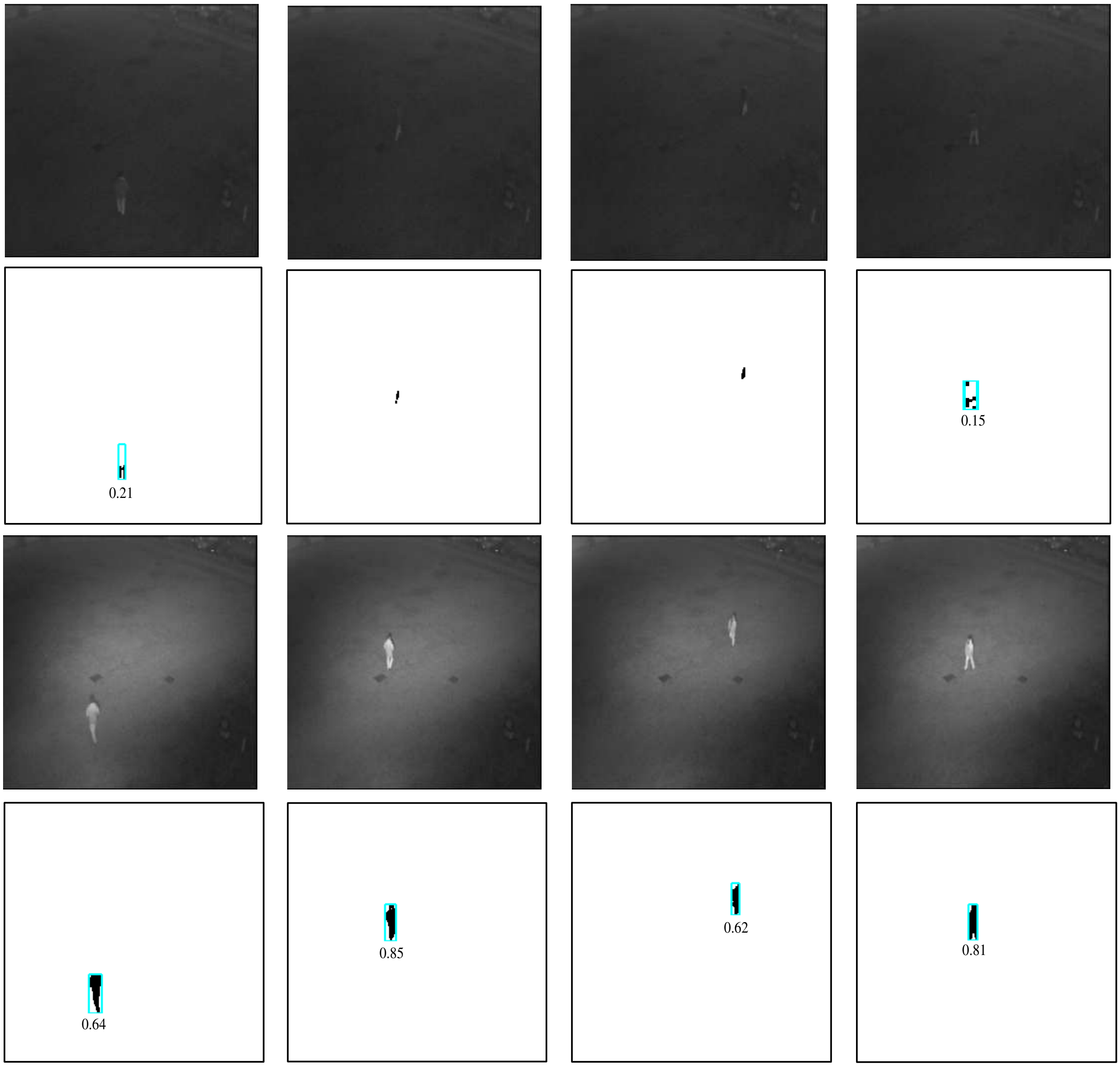

Fig. 9: Images and blobs from the fog sequence. AR values are indicated below each blob.

5. As can be seen, the two color sensors are performing similarly, but fusing their estimates allows for an overall reduction of calibration error and a trajectory more similar to the ground truth.

	Mean	σ
First color sensor	3,46	1,12
Second color sensor	2,92	1,38
Data fusion	2,12	1,02

Table 1: Mean and standard deviation (in pixels) of the distance between estimated and ground truth positions for the courtyard sequence.

Images taken from a second experiment are shown in Figure 9. The video sequences were taken at night and dense banks of fog were present. The scene was irradiated with IR rays and monitored by a color camera and a B/W camera with near infrared response.

It can be seen how the IR sensor outperforms the color camera: the AR values of the blob corresponding to the IR sensor are consistently higher. This is directly reflected by the correct segmentation of the silhouette of the person. Figure 10 shows a plot of the AR values scored by the two sensors for the blob. It can also be noted how the color camera is not able to discriminate the person as he moves away and into a fog bank (the AR values in Figure 10 are not indicated in the graph of since the blob is not detected).

This experiment shows how AR values can be used to to automatically and dynamically select the best sensors available. A threshold is set, if a given sensor extracts a blob with AR value above threshold, then its estimate is considered in the fusion process, otherwise it is considered unreliable and it is discarded. In this case, the AR values given by the color camera were always below threshold and therefore has not contributed to generate the final trajectory

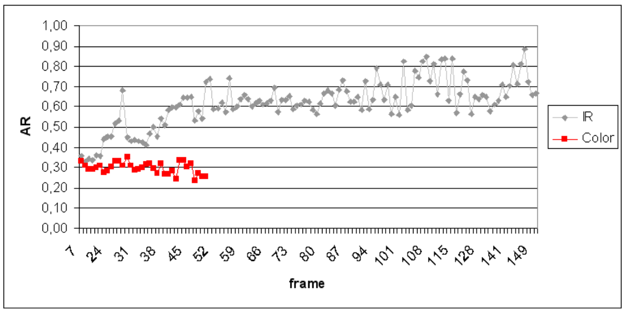

Fig. 10: AR values for the fog sequence

which was then entirely formed by IR estimates.

In Figure 11, the trajectory computed by the system is plotted with dots while the trajectory plotted in black denotes the ground truth path covered by the walking person. The two images represent the trajectory computed using optical and IR data respectively.

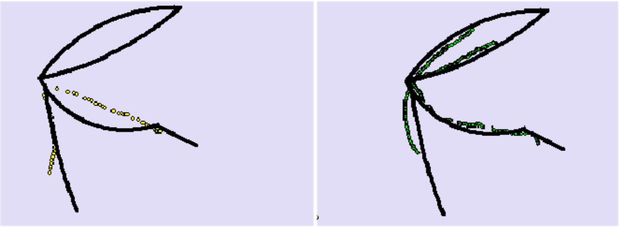

Fig. 11: Trajectory of the person according to the (left) color and (right) IR camera.

As can be seen comparing the images, the results obtained with the color camera only are poor as the trajectory is discontinuous and affected by segmentation errors. Figure 11(right) shows the trajectory computed using the IR video signal only. As expected, the trajectory is more continuous and close to the ground truth. However, the target is still temporarily lost (top right) as it traverses a dense fog bank.

Table 2 reports the mean and standard deviation of the distance (in pixels, 1 pixel $\approx$ 10 cm) from measured ground truth positions on th map of the walking person in Figure 10 for the two sensors. Notice how the IR camera is clearly performing better. This rather extreme experiment was shown to demonstrate how the AR values can be used to dynamically evaluate the performance of the sensors. This can be extremely important for a surveillance system for outdoors where weather and illumination conditions vary continuously and the sensors respond differently to these variations. Exploiting AR values to evaluate the performance of the sensors allows to choose those performing better at every time instant. This ultimately leads to obtaining accurate target detection and trajectory estimation.

	Mean	σ
IR sensor	8,66	4,42
Color sensor	17,22	9,28

Table 2: Mean and standard deviation (in pixels) of the distance between estimated and ground truth positions for the fog sequence.

Achieving better trajectory accuracy and continuity is of paramount importance for the successive steps of behavior understanding performed by a surveillance system. In particular, the trajectories of the objects in the scene have to be analyzed to detect suspicious events [1, 2]. The system can in fact be trained to discriminate between the patterns generated by the normal activities of the moving objects in the monitored space, and anomalous or suspicious movements.

4 Conclusions

In this paper, sensor reliability is explicitly considered in a multi-camera system for video surveillance of outdoor environments. A confidence measure has been defined to automatically weight redundant measurements of the targets' location coming from the different sensors in the data fusion process. In this way localization errors due to incorrect segmentation of the blobs have been reduced as well as the calibration errors due to perspective transformations. Preliminary experimental results show the effectiveness of the chosen confidence measure for automatic sensor weighting and the greater accuracy achievable by the proposed data fusion approach in comparison with single camera systems. In particular, the fusion procedure has produced trajectories that are more continuous and therefore useful for a surveillance system.

Acknowledgments

L. Snidaro and G.L. Foresti's work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project ”Distributed systems for multi-sensor recognition with augmented perception for ambient security and customization” (2002-2004).

References

[1] C. Regazzoni, V. Ramesh, and G.L. Foresti. Special issue on video communications, processing, and understanding for third generation surveillance systems. *Proceedings of the IEEE*, 89(10), 2001.

[2] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Special section on video surveillance. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 22(8), August 2000.

[3] G.L. Foresti, P. Mahonen, and C.S. Regazzoni. *Multimedia Video-Based Surveillance Systems: from User Requirements to Research Solutions*. Kluwer Academic Publishers, 2000.

[4] G.L. Foresti, C.S. Regazzoni, and P.K. Varshney. *Multisensor Surveillance Systems: The Fusion Perspective*. Kluwer Academic Publisher, 2003.

[5] D.N. Jayasimha, S.S. Iyengar, and R.L. Kashyap. Information integration and synchronization in distributed sensor netwoks. *IEEE Transactions on System, Man, and Cybernetics*, 21(21):1032–1043, Sept./Oct. 1991.

[6] A. Knoll and J. Meinkoehn. Data fusion using large multiagent networks: an analysis of network structure and performance. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 113–120, October 1994.

[7] H. Qi, S. Iyengar, and K. Chakrhbarty. Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 31(3):383–391, 2001.

[8] G.L. Foresti. Real-time detection of multiple moving objects in complex image sequences. *International Journal of Imaging Systems and Technology*, 10:305–317, 1999.

[9] G.L. Foresti. Object detection and tracking in time-varying and badly illuminated outdoor environments. *Optical Engineering*, 37(9):2550–2564, 1998.

[10] Y. Bar-Shalom and X.R. Li. *Multitarget-multisensor tracking: principles and techniques*. YBS Publishing, 1995.

[11] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, January 1997.

[12] Y. Bar-Shalom and W. D. Blair (eds.). *Multitarget multisensor tracking: applications and advances Volume III*. Artech House, 2000.

[13] G.L. Foresti, V. Murino, C. Regazzoni, and G. Vernazza. A multilevel fusion approach to object identification in outdoor road scenes. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(1):23–65, 1995.

[14] A. Utsumi, H. Mori, J. Ohya, and M. Yachida. Multiple-view-based tracking of multiple humans. In *Proceedings of the 14th ICPR*, pages 597–601, 1998.

[15] A. Nakazawa, H. Kato, and S.Inokuchi. Human tracking using distributed vision systems. In *Proceedings of the 14th ICPR*, pages 593–596, 1998.

[16] T. Sogo, H. Ishiguro, and M.M. Trivedi. Real-time target localization and tracking by n-ocular stereo. In *IEEE Workshop on Omnidirectional Vision (OMNIVIS'00)*, pages 153–160, 2000.

[17] Takashi Matsuyama and Norimichi Ukita. Real-time multitarget tracking by a cooperative distriuted vision system. *Proceedings of the IEEE*, 90(7):1136–1150, 2002.

[18] D. L. Hall and A. K. Garga. Pitfalls in data fusion (and how to avoid them). In *Proceedings of Fusion'99*, July 1999.

[19] K. Skiestad and R. Jain. Illumination independent change detection for real world image sequences. *Computer Vision Graphics and Image Processing*, 46:387–399, 1989.

[20] P.L. Rosin and T. Ellis. Image difference threshold strategies and shadow detection. In *Proceedings of the 6^{th} British Machine Vision Conference*, pages 347–356. BMVA Press, 1995.

[21] G.L. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Transaction on Circuits and Systems for Video Technology*, 9(7):1045–1062, 1999.

[22] Gian Luca Foresti. Real-time detection of multiple moving objects in complex image sequences. *International Journal of Imaging Systems and Technology*, 10:305–317, 1999.

[23] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. A system for video surveillance and monitoring. *Proceedings of the IEEE*, 89:1456–1477, 2001.

[24] L. Snidaro and G.L. Foresti. Real-time thresholding with Euler numbers. *Pattern Recognition Letters*, 24(9-10):1533–1544, June 2003.

[25] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 25(5):564–575, 2003.

[26] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, (4):323–344, 1987.

[27] O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 321–334, 1992.

[28] I. Mikić, S. Santini, and R. Jain. Tracking objects in 3d using multiple camera views. In *Proceedings of ACCV*, January 2000.

[29] S.S. Balckman. *Multiple-target tracking with radar applications*. Artech House, 1986.

[30] A.B. Poore. Multi-dimensional assignement formulation of data association problems arising from multi-target and multi-sensor tracking. *Computational Optimization and Applications*, 3:27–57, 1994.

[31] Y. Bar-Shalom and L. Campo. The effects of the common process noise on the two-sensor fused-track covariance. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(6):803–805, 1986.

[32] Y. Bar-Shalom and T.E. Fortman. *Tracking and data association*. Academic Press, 1988.

[33] J.B. Gao and C.J. Harris. Some remarks on Kalman filters for multisensor fusion. *Information Fusion*, 3:191–201, 2002.