Cooperative tracking of moving objects and face detection with a dual camera sensor

Stefano Ghidoni, Alberto Pretto, Emanuele Menegatti
Intelligent Autonomous Systems Laboratory (IAS-Lab)
Department of Information Engineering
The University of Padua
via Gradenigo, 6/B – 35131 Padova – Italy
{ghidoni, pretto, emg}@dei.unipd.it

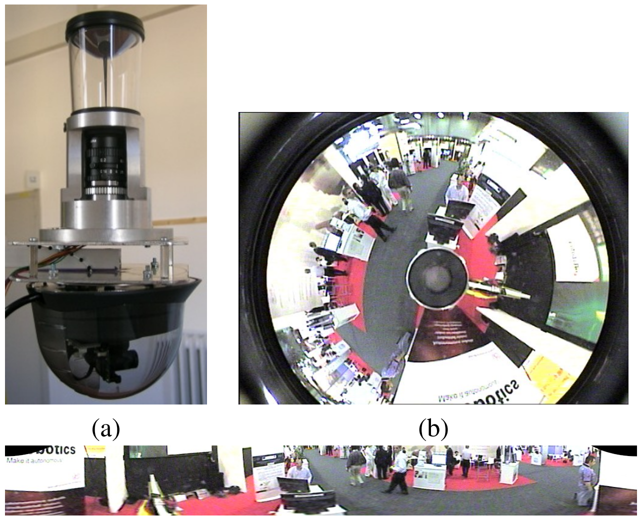
***Abstract*— This paper describes a sensor for autonomous surveillance capable of continuously monitoring the environment, while acquiring detailed images of specific areas. This is achieved by exploiting an omnidirectional camera and a PTZ camera, assembled together on a single mount. The two cameras form a single vision sensor, since data obtained processing the two images are used in a cooperative way. This system solves the problem affecting systems based on PTZ cameras only, since it does not exist a tracking system working reliably on PTZ images: the problem is solved here by performing the tracking in the omnidirectional image. This vision sensor is used in a surveillance application, that detects moving objects, and records all the faces of the people walking close to the sensor. It could be used to navigate or instruct a security or a service mobile robot.**

I. Introduction

Video surveillance is a topic that has been deeply studied by artificial vision researchers since several years. Improvements in both artificial vision algorithms and video sensors led to high quality intelligent video surveillance systems.

As long as systems that analyse video streams become smarter, it emerges the need for smarter sensors, that can be controlled by the surveillance systems themselves. Pan-Tilt-Zoom (PTZ) cameras are an example of this trend, since they can be controlled by an automated surveillance system in order to better investigate areas where, for instance, alerts have been issued. Topics like focus of attention [1] or how to manage multiple sensors [2], [3], [4] have been deeply studied.

This paper presents an innovative sensor, called Omnidome®[1], suited for intelligent video surveillance systems, and its application in the field of motion detection and face detection. The sensor, shown in fig. 1 (a), is composed by two cameras, mounted one on the top of the other. The upper one is an omnidirectional camera, i.e., a camera looking at an omnidirectional mirror, which provides a 360° view around the sensor. The lower camera is a common PTZ camera, whose zoom, pan, and tilt angles can be electronically controlled. Omnidome was licensed for commercialization to IT+Robotics srl[2].



(a) (b)

(c)

Fig. 1. A prototype of the Omnidome sensor (a). The two cameras – omnidirectional, upper, and PTZ, lower – are visible. Example of an image acquired by the omnidirectional camera (b), and the result of the unwarping process (c).

This sensor can be seen as an integrated master-slave camera system, having an omnidirectional master camera, like in [5], [6], [7], [8], [9]. The master and slave cameras are put together to form a single "multiple view" sensor. However, the master-slave model is slightly modified here, and tends to a cooperative agent, because, on one hand, the master camera issues commands to the slave, but, on the other hand, the latter sends reports back to the former, influencing its behaviour. They can be said to be two interacting intelligent vision agents.

In this paper, the sensor described above is used together with a video-surveillance application, whose aim is to monitor people walking around, and to record their faces.

The paper structure is as follows: section II describes the processing performed on the images acquired by the omnidi-

We wish to thank IT+Robotics for their support to this research. This research was partially funded by the project 2105/201/4/1268/2008 of Programma Operativo F.S.E. 2007-2013 Regione Veneto, Asse "Capitale Umano" – D.G.R. n. 1268 del 26/05/2008. Progetto finanziato con DDR n. 112 del 15/10/2008.

[1]Omnidome is a trademark of IT+Robotics srl

[2]www.it-robotics.it

rectional camera: it is presented a new technique for fusing together background subtraction and frame differencing, that often considered as alternatives. In section III, the problem of the calibration between the two cameras is tackled: unlike other works, this issue is solved here without the need of transformations to the world coordinates, which offers the sensible advantage that a complete camera calibration is not required. Section IV explains how images coming from the PTZ camera are processed in order to perform face detection, while in section V the experiments that were carried out are described. Finally, in section VI some conclusions are drawn.

II. Omnidirectional image processing

The goal of the omnidirectional camera for this application is to monitor the environment by detecting motion. Moving objects could then be further analysed, depending on their appearance. In order to meet this goal, acquired frames are unwarped, and then a motion detector based on frame differencing and background subtraction is used.

A. Image unwarping

The omnidirectional camera acquires images like that shown in fig. 1 (b). Even if it is possible to work directly on this kind of images, the omnidirectional image is unwarped into the corresponding panoramic cylinder, and all further processing operates on images that are like that in fig 1 (c). Some examples of image unwarping can be found in [5], [6], [10]. This choice is meant to ease the comparison between images acquired by the two cameras, at the cost of additional computational load, that can anyway be made negligible, like it will be described later.

B. Background subtraction and frame differencing

Once the panoramic cylinder is created, we extract the background with a combination of background subtraction and frame differencing. These two techniques are usually alternative to create the background. However, by combining them, it is possible to obtain the advantages of both, and compensate their drawbacks. The frame differencing is a technique that presents a number of negative aspects, like the high noise sensitivity, and the reduced capability of detecting the inner parts of large objects. However, the frame differencing is very robust to illumination changes, that is the weak point of background subtraction.

Background subtraction is performed by comparing the current frame with a reference image. The reference image changes slowly with time, and it is slowly updated in those regions where objects are not moving. Typically, an object needs to appear in the same position for minutes before becoming part of the background.

Frame differencing is performed by comparing the current frame with the previous two, and selecting the areas that are substantially different in at least one of the comparisons. A time window of three frames is used here in order to better extract moving objects, since this technique generates blobs which are better connected with respect to those obtained when only one frame in the past is considered. Blobs found by frame differencing are then filtered using the background image, by eliminating those portions that are similar to the background: in this way it is possible to discard all the regions where a change happened, but for which the last image contains the background.

C. Blob extraction and multi-target tracking

Starting from the image obtained with the above processing, blobs of motion are extracted: for each blob, basic descriptors are evaluated, like position, dimensions, aspect ratio, and pixel content. These descriptors are exploited for tracking objects. We adopt a voting scheme strategy for determining the matches, based on the aspect ratios, positions and vertical histograms of the blobs. Superimposition between the two blobs is not considered, because, at a frame rate of 7-10 Hz, and at certain distances, a person that walks fast can appear in two consecutive positions that have no overlap.

At this point it should be noted that, when comparing features of different blobs, the choice of working on the panoramic cylinder sensibly eases the task, which can exploit algorithms developed for perspective cameras, without the need for specific processing for omnidirectional images.

As long as a blob is being tracked for a long observation time, the tracker gets more information about its motion, and begins exploiting this information: coherence with previously observed motion is also considered when evaluating associations. Such coherence is calculated by considering the mean motion vector, referred to the X-Y plane of the unwarped image, observed in the most recent frame transitions, and comparing it with the motion vector that would result by the association being evaluated. Experimentations suggest that by considering the last 5 frames good results are achieved.

This way of processing motion information turns out to be very light from a computational point of view. Sometimes this method causes wrong blob associations, if compared with other techniques adopting more precise data filtering, but the number of wrong associations is negligible with respect to the obtained speed-up.

D. Image unwarping with CUDA

Usually, omnidirectional image unwarping is a task whose computational time is not negligible. Using an efficient method exploiting look-up tables provided by OpenCV, the algorithm needs roughly 10-20 ms to be completed. We propose the use of the CUDA technology [11] in order to sensibly reduce the CPU-time needed for the unwarping process. CUDA technology exploits the highly-parallel architecture of the graphic processor: the advantage in terms of computational speed depends on the particular graphic card being used, but is in any case outstanding, like it was observed also for other applications.

In our case, the unwarped image dimension is 1280×156. Choosing a bilinear interpolation method, that provides the best time-performance balance, the CPU needs 13 ms to obtain the unwarped image, while a Nvidia 9200 GS graphic card obtains the same result in only 1.4 ms, and a 8800 GTS card needs only 0.2 ms.

III. Inter-camera calibration

The two cameras composing the sensor need to be calibrated in order to be able to exchange information about what they are observing. In our case, the system should be able to detect moving people, recognize their faces, and record some shots of each one.

The proposed camera system takes advantage of being one single, self-contained sensor. Unlike [5], [6], there is a direct relationship between the position of a blob in the omnidirectional image, and the pan-tilt angles of the PTZ camera needed to point it.

The whole system is said to be a single sensor because it knows about its geometry, and it should be able to convert information referred to the omnidirectional image coordinate system, directly to pan-tilt angles for the moving camera. The calibration is called "inter-camera" because it provides only information on reciprocal positions and orientations, without connection with the world coordinate system.

Given a certain blob in the omnidirectional image, and a point on it, the process of recovering the pan and tilt angles that let the PTZ camera to frame it, is as follows.

A. Pan angle

To determine the pan angle, α, the sensor should know X_{ref}, the x-coordinate of the unwarped image corresponding to the reference 0° pan position of the PTZ camera. Then, to frame a point having a x-coordinate value of X_{obj}, it suffices to evaluate:

$$\alpha = \frac{X_{\mathrm{obj}} - X_{\mathrm{ref}}}{W_{\mathrm{Image}}} \times 360^\circ \,, \tag{1}$$

where W_{Image} is the width of the unwarped image. After evaluating the above equation, the system verifies that the position is compatible with the movement limits of the PTZ camera, and with the angle range that is accepted by the control interface (for instance, [-90;90] or [0;180]).

B. Tilt angle

The evaluation of the tilt angle, β, is more complicated, and needs a precise knowledge of the mirror geometry. Other works empirically measured the angle [5], [6], and created a table of tilt angles, each one corresponding to a range of y-coordinates in the unwarped image. However, this method requires an empirical calibration procedure for each mirror type, and introduces errors.

Since the mirror shape is known, it is possible to analytically determine the angle of each incoming light ray. Consider the scheme in fig. 2, where a hyperbola is depicted, representing the hyperbolic omnidirectional mirror of the system. The points A and B are the focal points, and the camera is placed so that the optical center coincides with A, pointing at the omnidirectional mirror. In the scheme, the camera and the mirror are perfectly aligned, and the optical axis of the camera intersects the mirror perpendicularly. In a 2-dimensional domain, the equation of the mirror is:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = -1 \,, \tag{2}$$

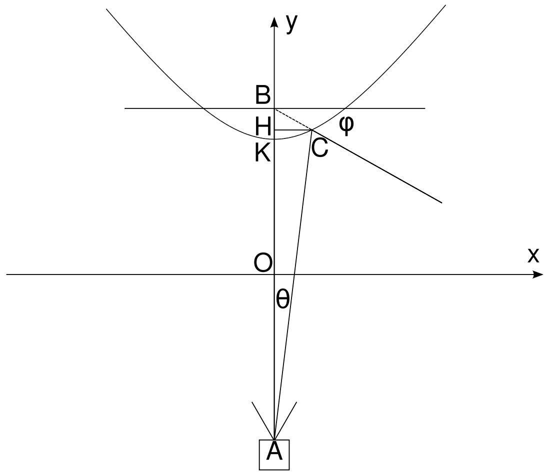


Fig. 2. Scheme of the omnidirectional camera, obtained by using a common camera looking at a hyperbolic mirror, and placed so that the optical centre coincides with one of the foci of the hyperbola.

where $a = 29.00\,\mathrm{mm}$ and $b = 40.73\,\mathrm{mm}$ for our system. Geometrically, b is the length of the segment OK. The distance between the two foci, that is the length of the segment AB, is $2c = \sqrt{a^2 + b^2}$. Finally, $c = \overline{AO} = \overline{BO}$.

Now, consider the problem of determining the direction of a light ray that appears, in the image, at a distance r to the center of the mirror, knowing that the mirror itself has a maximum radius R in the image domain. We call D the distance between the optical center A and the plane containing the largest section of the mirror, that is, the back part of the mirror; L is the maximum radius of the mirror, the one appearing with radius R in the image domain. Then the angle θ under which the point is seen will be:

$$\theta = \arctan\left(\frac{\mathrm{rL}}{\mathrm{RD}}\right) \,. \tag{3}$$

Once the angle θ is known, it is possible to evaluate the direction of the light ray that is incident on the mirror: this is angle φ in fig. 2. Similar problems have already been tackled in other works, see [12] for an exhaustive reference. Those general works, however, lack a simple formula expressing the relationship between angles θ and φ. Such a simple formula can be found in [13], but without demonstration. In the following, the demonstration is reported, since it is not trivial.

First of all, note that a hyperbola is the locus of points where the difference between the distances to the foci is a constant. Since both C and K belong to the hyperbola, then the following equality holds:

$$\overline{AC} - \overline{BC} = \overline{AK} - \overline{BK} \,. \tag{4}$$

Since:

- $\overline{AK} = \overline{AO} + \overline{OK} = b + c$, and
- $\overline{BK} = \overline{BO} - \overline{OK} = c - b$,

then:

$$\overline{AC} - \overline{BC} = 2b \,. \tag{5}$$

The cosine law applied to the triangle ABC states that:

$$\overline{BC}^2 = \overline{AB}^2 + \overline{AC}^2 - 2\overline{AB} \cdot \overline{AC} \cos\theta \ : \tag{6}$$

by substituting (5) into the above equation, and recalling that $\overline{AB} = 2c$, the following relationship results:

$$\overline{AC} = \frac{b^2 - c^2}{b - c\cos\theta}\,, \tag{7}$$

therefore:

$$\overline{CH} = \overline{AC}\sin\theta = \frac{b^2 - c^2}{b - c\cos\theta}\sin\theta\,, \tag{8}$$

$$\overline{BH} = 2c - \overline{AC}\cos\theta = \frac{2bc\sin\theta - \left(b^2 + c^2\right)}{b\sin\theta - c}\,. \tag{9}$$

The angle φ can be finally calculated:

$$\varphi = \arctan\left(\frac{\overline{BH}}{\overline{CH}}\right) = \arctan\left(\frac{2bc - \left(b^2 + c^2\right)\cos\theta}{\left(b^2 - c^2\right)\sin\theta}\right)\,. \tag{10}$$

The angle φ determines the direction of the incident light referred to the omnidirectional camera, which is mounted close to the PTZ camera. Considering that the distance between the sensor and the observed objects is large with respect to the displacement between the two cameras, the PTZ camera can be set to have the aforementioned tilt angle β equal to the angle φ.

IV. Image processing for the PTZ camera

The PTZ camera is exploited to acquire detailed images of a chosen object, selected among the moving blobs tracked by the omnidirectional camera. In this work, images provided by the moving camera are processed using a face detection algorithm. The face detector is used to classify whether the blob is a person or not; if the face is detected, it is recorded.

The face detection phase is carried on exploiting the well-known cascade of Haar-like features suggested by Viola and Jones, [14], [15]. The Viola-Jones object detection technique is based on the AdaBoost (Adaptive Boosting, [16]) learning algorithm applied to a set of Haar-like features, efficiently extracted from the images using the so-called integral image. Such technique has been chosen because of its speed and high performance.

A. Data fusion

As it has been described, the omnidirectional camera controls the PTZ unit by setting its pan and tilt angles, i.e., by choosing which moving blob should be framed. The system then uses data provided by the face detector to integrate information stored in the tracker: when a face is found, the tracking system is updated with this information. For each tracked blob it is therefore known if it has been framed by the PTZ camera, and if a face was found in the detailed images.

Information about past results of the face detection module becomes important when the tracking system is targeting multiple moving elements. In this case, a policy for moving the PTZ camera on the subjects should be chosen: this policy aims at framing the highest number of moving objects, ideally all of them. The policy gives priority to the objects that are located far from the camera, or that are moving fast, because in both cases the object may become hard to view after few time. The dimensions of the blobs are also taken into account, and too small blobs are discarded, since they would be hard to analyze.

Once the system chooses which object the PTZ camera should be pointed to, it lets the face detector work for few seconds, and records all the recognized faces; multiple shots of the same person can be taken. When the observation time is elapsed, the tracker is notified whether a face was found, or not: in the first case, the corresponding blob is assigned the lowest priority, because it has already been successfully observed, and, in principle, there is no need to go back again to it. When no face is detected, the subject either is not a person, or the person is seen in a way that does not make the detection possible (e.g. the back of the head is seen). In this case, the blob is assigned a low priority, in order to let the system focus on other objects; but further investigations on it are scheduled, trying to frame the subject after some time, when it might be seen in a different way.

V. Experiments

System performance measurement was obtained by placing the sensor in several environments, and calculating how many faces were detected with respect to the total number of people that appeared in the field of view. There are two main factors that influence performance: the capability of the system to detect moving objects, and point the PTZ camera so that it can frame people, and the performance of the face detector. In almost all cases observed during the testing phase, however, the face detector was able to detect the faces appearing in the images: in some cases the detection turned out to be very stable; in some others it appeared weaker.

In order to keep the influence of the face detector performance low, and considering that the final goal of the application is to acquire shots of the face of each person walking in the field of view, the system performance was calculated as the ratio between the number of people for which at least one image of the face was recorded, with respect to the total number of people that walked close to the sensor.

It was considered acceptable the case when, for a person, not only the face, but also some other wrong object was found by the face detector, given that at least one correct shot of the face was taken. This choice corresponds to neglecting the problem of false positives of the face detector, that is fair, since this work is not focused on face detection, but rather, exploits a face detector. False positives were anyway measured, and ranged from 0.03 to 0.4 false positives per frame, in the easiest and most complicated environment, respectively. The value of 0.4 is quite high; however, we chose not to care about it, as long as the face detector could at the same time record all the faces in the image.

A. Test scenarios

The sensor was tested in different environments. The most easy one was a corridor with few people walking: in this case, every person was correctly detected by the tracking system, framed by the PTZ camera, and at least one shot of the face was recorded. We do not report these experiments due to space limitations.

A more difficult situation was encountered when a person appeared already close to the sensor: for instance, while coming out of a room whose door was very close to the sensor. In this case, the system was quickly alerted by the omnidirectional camera, but, depending on the actual position of the PTZ camera, the latencies introduced by mechanical movements could make the system unable to frame the person.

The most difficult case was observed while testing the sensor during a fair exhibition. A lot of people were moving: some walking, some other staying in the same place, but generating minor movements. Experiments performed in this scenario were particularly useful to tune the target selection policy: at the beginning, the sensor got confused, and switched too often between the several moving objects that were correctly detected in the omnidirectional images.

The system was then better tuned, depending also on the features of the specific PTZ unit. The sensor should, in fact, adopt a policy that knows how fast the moving camera is, and, for each target, know in advance how long it will take to point it, comparing this time with the speed of the target itself.

In the last, more complicated, scenario, the sensor was able to detected and point 93% of the people walking around. Groups of people did not create any problem, apart from occlusions, since they generated a big motion pattern, were pointed by the PTZ camera, and all the faces were detected.

All faces missed by the system belonged to people detected by the tracking system, but not framed by the moving camera, because of the mechanical latencies: when the camera turned out to point the subject, the face was not visible anymore, because of the subject's movement.

Some examples of the system results can be seen in fig. 3. In (a), a sample of the omnidirectional image processing: each rectangle identifies a region where movement was detected; it can be seen that the person on the right generates multiple motion patterns. In (b) it is shown the image of the PTZ camera acquired when observing the scene in (a): the person walking around the sensor has been correctly framed; it should also be noted that the shot was taken while the camera was rotating: the walking person appears still in the image, while other people standing in the background appear blurred, due to the camera motion.

In (c), two people partially overlapping are recognized by the face detector: groups are not an issue, as long as all the faces are visible. In (d), the effect of motion blur when the PTZ camera is changing its position. When this happens, faces are detected in some cases, but the resulting dumped image would not be useful, because of the poor quality of the blurred image. In (e), an example of false positive that is difficult to eliminate, because the advertisement sign actually does have a face depicted on it. Since the whole system also tracks movement, however, it would be possible to filter this kind of false positive, given that the inter-camera calibration and the positioning system of the PTZ camera are both extremely precise.

B. Computational time

The described algorithm needs an average time of 222 ms to process each frame on a Core2 Duo processor working at 2.10 GHz. This time includes image unwarping (13 ms), omnidirectional image processing (30 ms), the GUI (4 ms) and the face detection algorithm, which is the slowest part, and needs 175 ms. This leads to the conclusion that if the face detection algorithm is not run on the whole PTZ image, but rather on a subwindow, the overall processing time can be substantially decreased. This would be possible if both the inter-camera calibration and the PTZ camera positioning were extremely precise.

Moreover, the face detection algorithm needs not be run on every frame, but rather, only when a moving object needs to be investigated. The task of looking for possible targets is therefore performed in 43 ms, that can be reduced to 31 ms (-28%) by performing the unwarping on the GPU, that takes 1 ms only.

C. Future developments

Some ideas about future developments of the system include the quality check of the images obtained by the face detection algorithm working on images acquired by the PTZ camera. In the current implementation, once the shot is taken, it is simply stored: a more efficient system should be capable of evaluating whether the quality of the image containing the face is good or not.

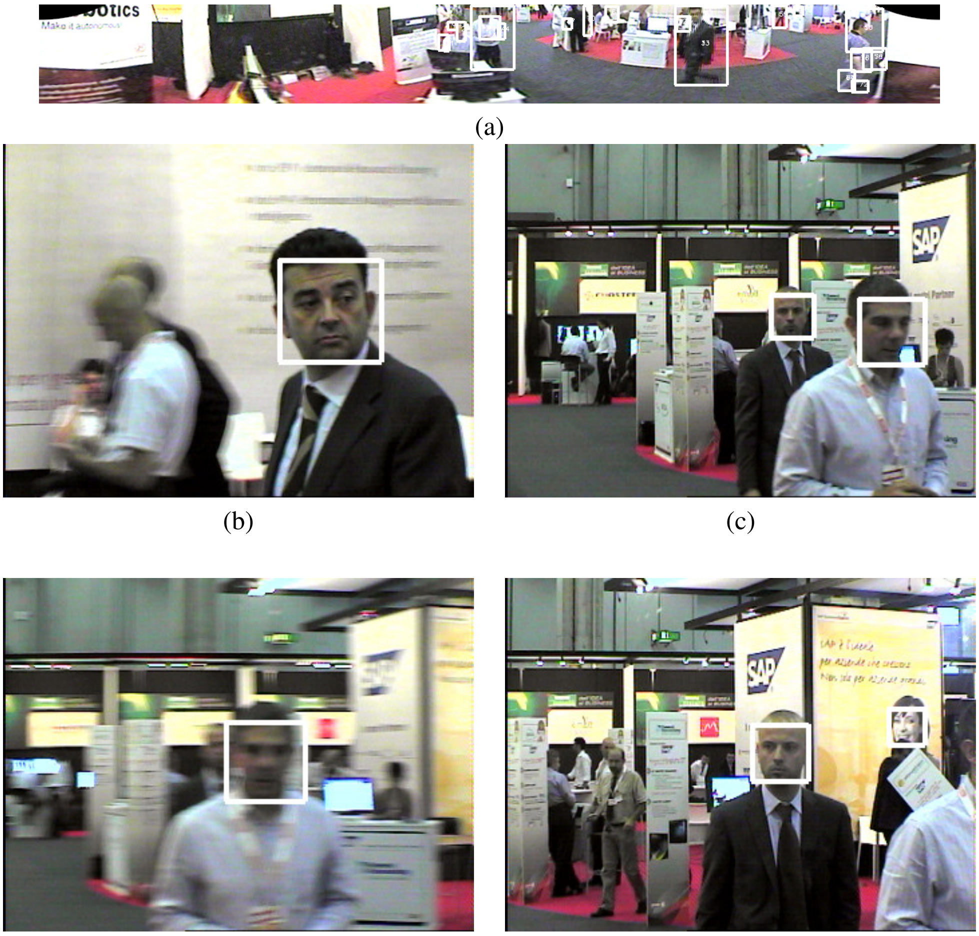
VI. Conclusions

In this paper, a sensor composed of two cameras has been presented, together with a video-surveillance application for detecting and recording the faces of the people moving around. The sensor is composed by two cameras presenting complementary advantages and drawbacks, and is capable of monitoring all the surrounding area, while, at the same time, concentrating on some details where an interesting event is occurring. The two cameras are considered as a single sensor, since processing results of both cameras are fused together to obtain better performance: in this sense, none of the two cameras can be considered the master, completely controlling the other one.

An application for detecting the faces of the people moving around the sensor has also been described; results were reported of experiments of successful real-time motion tracking and face recognition in wide environments, where data provided by the two cameras are combined together.

References

[1] J.W. Davis, A.M. Morison, and D.D. Woods. An adaptive focus-of-attention model for video surveillance and monitoring. *Machine Vision and Applications*, 18(1):41–64, 2007.



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

Fig. 3. Some results provided by the system: in (a), the output of the motion detection algorithm, and in (b) the corresponding PTZ image framing the face of a person walking nearby the sensor; the person is walking, but is defined in the image because the sensor itself is rotating: two people standing in the background are blurred due to motion. In (c) a group of people is framed, and both faces are detected. A blurred face is recognized in (d), while (e) shows a false positive of the face detector, together with a correct detection.

[2] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, Oct. 2001.

[3] H. El-Alfy, D. Jacobs, and L. Davis. Assigning cameras to subjects in video surveillance systems. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pages 837–843, May 2009.

[4] O. Javed and M. Shah. *Automated Multi-Camera Surveillance: Algorithms and Practice*. Springer Verlag, 2008.

[5] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi, and C.S. Regazzoni. A novel dual camera intelligent sensor for high definition 360 degrees surveillance. In *Intelligent Distributed Surveillance Systems (IDSS-04), IEE*, pages 26–30, Feb. 2004.

[6] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi, and C.S. Regazzoni. Dual camera intelligent sensor for high definition 360 degrees surveillance. *IEE Proceedings, Vision, Image and Signal Processing*, 152(2):250–257, Apr. 2005.

[7] S. Cagnoni, M. Mordonini, L. Mussi, and G. Adorni. Spherical rectification applied to hybrid dual camera vision systems. In *Workshop Proceedings of SIMPAR 2008, International Conference on SImulation, Modeling and Programming for Autonomous Robots*, pages 366–376. Emanuele Menegatti, Nov. 2008. Published on CD.

[8] Y. Yao, B. Abidi, and M. Abidi. Fusion of omnidirectional and ptz cameras for accurate cooperative tracking. In *IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS '06*, Nov. 2006.

[9] R. Khoshabeh, T. Gandhi, and M.M. Trivedi. Multi-camera based traffic flow characterization & classification. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 259–264, Oct. 2007.

[10] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5):1015–1026, Oct. 2008.

[11] S. Che, M. Boyer, J. Meng, D. Tarjan, J.W. Sheaffer, and K. Skadron. A performance study of general-purpose applications on graphics processors using cuda. *Journal of Parallel and Distributed Computing*, 68(10):1370 – 1380, 2008. General-Purpose Processing using Graphics Processing Units.

[12] R. Benosman and S.B. Kang. *Panoramic Vision: Sensors, Theory, and Applications*. Springer Verlag, 1st edition, 2001.

[13] H. Ukida, N. Yamato, Y. Tanimoto, T. Sano, and H. Yamamoto. Omni-directional 3d measurement by hyperbolic mirror cameras and pattern projection. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pages 365–370, May 2008.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–511–I–518 vol.1, 2001.

[15] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 734–741 vol.2, Oct. 2003.

[16] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.