http://dx.doi.org/10.21611/qirt.2004.011

# Combining Video and Thermal Imagery for Robust Pedestrian Tracking

by H. Torresan, B. Turgeon , P. Hebert* and X. Maldague*

*Electrical and Computing Engineering Dept , Université Laval, Québec, Canada*

## Abstract

In the current context of increased surveillance and security, more sophisticated and robust surveillance systems are needed. One idea relies on the use of pairs of video (visible spectrum) and thermal infrared (IR) cameras located around premises of interest. To automate the system, a robust tracking algorithm and the development of an efficent technique enabling the merging of the information provided by the two sensors becomes necessary and these are described in this paper. Results are presented for a few typical situations.

## 1. Introduction

The objective of this paper is to present a robust pedestrian tracking system which will exploit the information provided by a visible spectrum sensor and an infrared sensor, while functioning within a complex environment. To-date, few tracking systems have made use of infrared information to track people [4,5]. However, many researchers have addressed the same task using the visible part of the spectrum [1,3]. The addition of an infrared sensor will provide information which complements that obtained with visible images. The latter offer a rich content where the detection of pedestrians can however be limited by a change in lighting conditions. Infrared images generally enable a better contrast to be achieved between the pedestrian and his environment, but they are less robust to temperature and wind changes. Exploiting the complementary information obtained and improving the precision and robustness of tracking requires the development of an efficient technique allowing the merging of this complementary information.

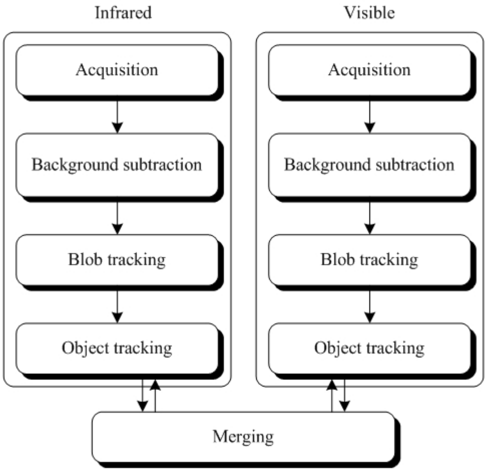

***Fig. 1.*** *Image processing flow chart.*

Merging visible and IR information can be done at different levels in the image processing. Nevertheless there is no clear evidence whether the fusion of these information should be made at the lowest, highest or at multiple levels. In this paper, we have developed a strategy where information from both channels is merged at the highest level. This is motivated by modularity where the same tracking algorithm is

http://dx.doi.org/10.21611/qirt.2004.011

implemented for both independent modalities; a merging block is added when both modalities are available.

Obviously, the main part of the work concerns image processing. An important hypothesis is that cameras do not move during the recording of one given sequence. Figure 1 presents the overall image processing algorithm. After the image acquisition, moving regions are extracted with a background subtraction algorithm. [2] In this paper, the processing algorithm for pedestrian tracking is first presented. Tracking is performed at two levels: blob and object. Next, the technique enabling the merging of the information provided by the tracking is outlined. The paper concludes with a presentation of a few results.

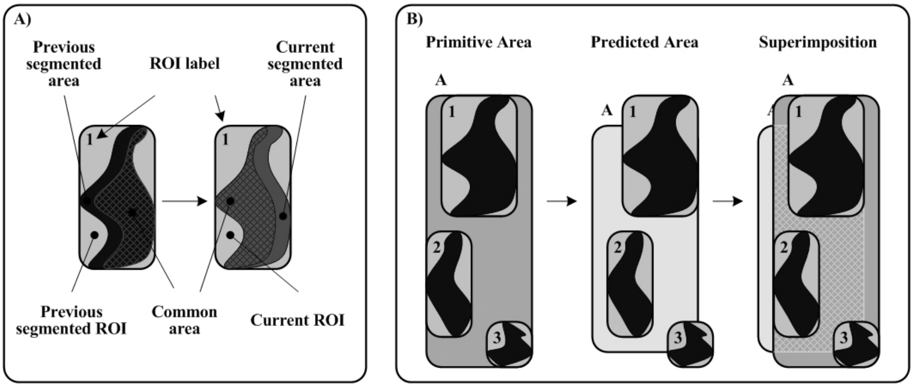

*Fig. 2. A) Blob notation. B) Object primitive and predictive areas.*

## 2. Two-level tracking

The algorithms of the first segmentation often provide data where the people are detected in the form of several blobs surrounded by noise and lacking certain body parts. The tracking algorithm presented here supports the incomplete and noisy data provided by the first segmentation. In order to do this, the tracking is performed on two levels. While the first level of the tracking algorithm consists in following the blobs in an image sequence, the second level builds on the first and tracks a combination of one or more blobs, i.e. objects. To do this, some feature parameters between blobs at time ’t’ and ’t-1’are first introduced.

“Overlapping”, $O(a,b)$, between two blobs a and b, is defined formally as:

$$O_{\max}(a,b) = Maximum(CS(a,b)/AROI(a), CS(a,b)/AROI(b)) \quad (1)$$

$$O_{\min}(a,b) = Minimum(CS(a,b)/AROI(a), CS(a,b)/AROI(b)) \quad (2)$$

where $AROI(i)$ is the area of the i$^{th}$ blob’s ROI (Region of Interest) and $CS(a,b)$ is the intersection area between the two ROI

http://dx.doi.org/10.21611/qirt.2004.011

“Similarity”, $S(a,b)$, is defined as:

$$S(a,b) = 1 - [Abs(A(a) - A(b)) / Maximum(A(a), A(b))] \quad (3)$$

where $A(r)$ is the actual area of the $i^{th}$ blob (see Figure 2a).

“Resemblance”, $R(a,b)$ ,between two ROI *a* and *b* is defined as:

$$R(a,b) = [O_{\min}(a,b) \times S(a,b)] \quad (4)$$

Figure 3 depicts all possible cases which can be met during the tracking of blobs. The maximum overlapping factor (Equation 1) is used like a criterion to initialise the follow-up of the blobs between two frames of a sequence. When a one-to-one correspondence is obtained, the same label is given to the blob of the new frame. When a complex case is obtained, a more accurate analysis must be carried out so as to reduce blobs of the complex case to a simple case (one-to-one correspondence, merging, separation, creation or destruction). An algorithm computing the resemblance factor between all of the blobs is used to simplify the complex case. The resemblance factor is based on the minimum overlapping and similarity factor and is used to eliminate much more of the correspondence between the blobs. During blob tracking, specific parameters, like the speed and the confidence, must be computed for the blob. These parameters will be used later during object tracking.

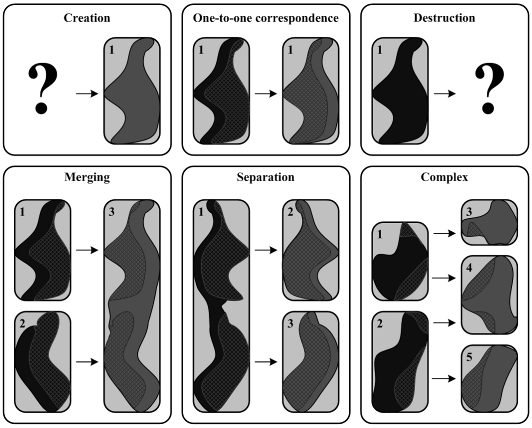

***Fig. 3.*** *Various cases that are observed in tracking.*

The confidence (C) is a feature that gives the persistence of a blob over time and is described by the following equation:

$$C(a) = \left(\sum_{b=0}^{n} R(a,b) \times C(b)\right) + 1 \quad (5)$$

where *a* represents the new blob at time t, *b* the preceding blob at time step *t-1* *a*nd *n* the number of preceding blob at time step *t-1* that are greater than one in a merging or a complex case.

At t=0, confidence is initialised to zero. As seen from Equation 5, the confidence on matching from *t-1* to *t* increases if the blob has been tracked for a long time and if the resemblance from two time steps is large.

http://dx.doi.org/10.21611/qirt.2004.011

Before we can detect a pedestrian, all blobs must be grouped together to create objects. An object can be made up of several blobs. An object is initially created from an isolated blob or many closer blobs. The object has a primitive area (PrA) that is generated from the ROI of each blob of the object and a predicted area state (PdA) that evolves (size, position and speed) differently from the primitive area (See, Figure 2b). The predicted area state is an important feature that gives a robust estimation of the real size, position and speed of the object. Several rules enable the updating of the list of blobs belonging to an object As time passes, an object at time *t* inherits the blob(s) making up the object at time t-1. A blob that appears in a predicted area is also added to the object. A blob that has an opposite movement from the object can be removed from the object. If no blob has been detected for a long period of time, the object vanishes. A non-moving object where blobs appear and disappear in time may be labelled as a noisy object such as a moving leaf in a tree. The object also has a confidence value that is computed as the average of the confidence of individual blobs comprising the object.

The size and position of the predicted area evolves with time. The size of the predicted area is updated using the dimensions of the primitive area. The sizes of both areas tend to be the same so that the predicted area is updated only if different. In this case the updated size is changed by 20 % at most. The adjustment of the position of the predicted area is first estimated using the mean speed of the predicted area of the last fifteen frames. Then, an algorithm corrects this estimated position by using the center of mass of the primitive area. This correction is limited by the difference between the position of the boundary of the predicted and primitive areas. The mean speed is updated with the new position value.

## 3. Merging

The merging algorithm improves the precision of the size and position of the predicted area computed during the *second level of tracking*. It is driven by three goals. The first one consists in establishing a correspondence between the objects detected in the visible and the IR images. For each pair of objects, the identification of the best object detected (in visible or IR images) describes our second goal. The object with the best detection are called *master* and the second one *slave*. The confidence is used as a criterion for better detection and is computed for all the objects of each frame in the sequence. In this manner the identification of the master and the slave will change rapidly for an object when fast light illumination or temperature variation are present. Our last goal consists in using the information of the *master* object to help in tracking the *slave* one. The merging process is done independently for each pair of objects. For example, if at time *t*, three objects can be detected in the visible and infrared images, two objects can be *master* in the infrared image, and one object can be a *master* in the visible image.

The merging algorithm has to determine situations where the position and the size of the predicted area need to be modified. These situations only occur when a great difference between the primitive area of the master object and the slave object is detected. In this case we enter in the “enslavement” mode where the *master* predicted area controls the *slave* predicted area. For example, if a pedestrian has a green T-shirt and walks in front of a green hedge, this person’s trunk will tend to disappear and the *slave* object will be put in the enslavement mode. The IR object will maintain a good detection and will help in tracking the pedestrian in the visible

http://dx.doi.org/10.21611/qirt.2004.011

image because the body temperature is higher than the temperature of the green hedge.

The merging algorithm is very useful in cases where two objects disappear and will allow objects to stay present in the system and allow the position of the predictive area to be assessed using the mean speed of the predictive area in the last frame. For example, if a pedestrian passes behind a tree, the objects will disappear in both images. If the pedestrian maintains his speed and direction, the object will be recovered when it appears on the other side of the tree. But if the pedestrian stops behind the tree and returns to the same side, the algorithm will create a new object.

## 4. Results

The algorithm described in the previous sections was tested on several sequences. Figures 4 and 5 illustrate various cases. It is obviously not possible to render the dynamics of these sequences in a paper and thus, some interesting situations were selected. In Figure 4, an outdoor situation of one pedestrian walking near a wall is presented and shows that the IR image can be helpful in removing shadows from the visible image. In Figure 5 two outdoor pedestrians are shown where the blobs of one pedestrian are not well detected in both IR and visible images. The merging algorithm improved detection for the predicted area of this pedestrian.

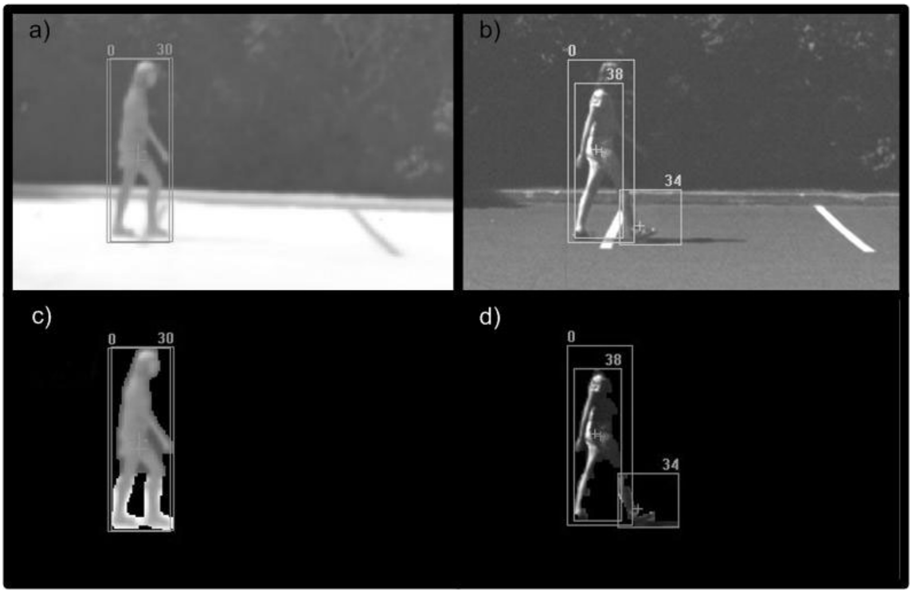

***Fig. 4.*** *Outdoor scene illustrating pedestrian extraction. a,b) Original IR and visible images. c,d) Representation of the blob detected for both IR and visible images. Note that the blob in the visible image also includes the shadow of the person. But the predicted region (labelled with zero in the upper-left corner) is the same in all pictures. The ROI of blobs (labelled 38 and 34 to the upper-right corner) are very different from the predicted regions in the visible image. The ROI of blob (labelled 30 in the upper-right corner) is similar to the predicted region labelled 0.*

http://dx.doi.org/10.21611/qirt.2004.011

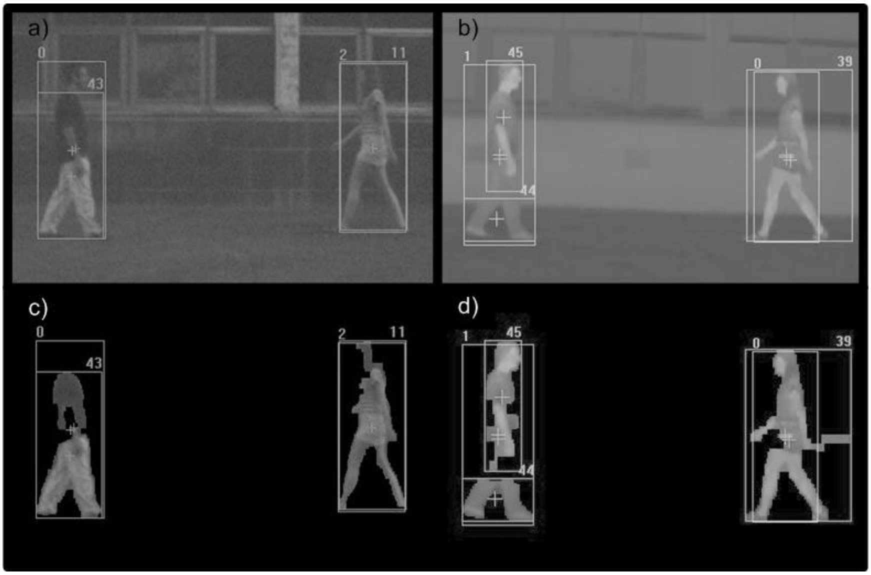

***Fig. 5.**. Night scene showing a two pedestrian extraction. a,b) Original IR and visible images. c,d) Representation of the blob detected for both IR and visible images. The rectangles with the label in the upper-left corner give the predicted area numbers (0,2,1,0) and the rectangles with the label in the upper-right corner give the blob numbers (43,11,45,44,39). We can see that the left pedestrian was not completely detected by the background subtraction algorithm in both IR and visible images. Meanwhile, the predicted area is well estimated for the two pedestrians. The right pedestrian is not well detected by the backgroundd subtraction algorithm in the IR image but the predicted region labeled 0 gives a very good estimation of the pedestrian.*

## 5. ACKNOWLEDGEMENTS

The supports of the Natural Sciences and Engineering Research Council of Canada and of the *Fonds FQRNT* of the Quebec Province are acknowledged.

## REFERENCES

[1] Haritaoglu, D.Harwood, L. S. Davis, “W4: Real-Time Surveillance of People and Their Activities,” *IEEE Transactions On Pattern Analysis And Machine Intelligence,* **22**[8]: 809-830, 2000S.

[2] A. Lemieux, M. Parizeau, “Flexible multi-classifier for face recognition systems” *Vision Interface,* S1.4, 2003 (http://kopernik.eos.uoguelph.ca/~zelek/vi2003/).

[3] Masoud, N. Papanikolopoulos, “A method for human action recognition,” *Image and Vision Computing*, **21** (2003): 729-743.C.

[4] H. Nanda and L. Davis, “Probabilistic Template based Pedestrian Detection in Infrared Videos,” *IEEE Intelligent Vehicle Symposium*, 2002

[5] F. Xu and K. Fujimura, “Pedestrian Detection and Tracking with Night Vision”, *IEEE Intelligent Vehicle Symposium*, 2002