Decision Fusion for Object Detection and Tracking Using Mobile Cameras

Luis David López Gutiérrez and Leopoldo Altamirano Robles

National Institute of Astrophysics Optics and Electronics, Luis Enrique Erro No 1,
Santa Maria Tonantzintla, Puebla, 72840 México
luis_david@ccc.inaoep.mx, robles@inaoep.mx

Abstract. In this paper an approach to the automatic target detection and tracking using multisensor image sequences with the presence of camera motion is presented. The approach consists of three parts. The first part uses a motion segmentation method for targets detection in the visible images sequence. The second part uses a background model for detecting objects presented in the infrared sequence, which is preprocessed to eliminate the camera motion. The third part combines the individual results of the detection systems; it extends the Joint Probabilistic Data Association (JPDA) algorithm to handle an arbitrary number of sensors. Our approach is tested using image sequences with high clutter on dynamic environments. Experimental results show that the system detects 99% of the targets in the scene, and the fusion module removes 90% of the false detections.

1 Introduction

The task of detecting and tracking regions of interest automatically is a fundamental problem of computer vision; these systems have a great importance in military and surveillance applications. A lot of work has already been carried out on the detection of multiple targets. However, detection and tracking of small, low contrast targets in a highly cluttered environment still remains a very difficult task.

The most critical factor of any system for automatic detection is its ability to find an acceptable compromise between the probability of detection and the number of false target detection. These types of errors can generate false alarms and false rejections. In a single sensor detection system, unfortunately, reducing one type of error comes at the price of increase the other type. One way to solve this problem is to use more than one sensor and to combine the data obtained by these different expert systems. In this paper we propose an approach to solve the automatic detection problem of objects using decision fusion, our principal contribution is improve the target detection and tracking results without specialization of the algorithms for a particular task; the approach was tested on a set of image sequences obtained from mobile cameras.

The paper is organized as follows. Section 2 introduces the models which are considered, and briefly they are described. Section 3 shows an overview of the approach. Sections 4 and 5 describe the algorithms used to detect objects of interest in visible and infrared image sequences respectively. Section 6 describes the method for combining the results obtained by the two algorithms. Several results that validate our approach are reported in section 7, and finally section 8 contains concluding remarks.

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 84–91, 2004.

2 Background

Parametric motion model: The parametric motion model w_θ represent the projection of the 3D motion field of the static background [1], where w_θ denotes the modeled velocity vector field and θ the set of model parameters. The parametric motion model is defined at pixel p = (*x,y*) as:

$$\vec{w}_\theta(p) = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix} = \begin{bmatrix} u(p) \\ v(p) \end{bmatrix} \tag{1}$$

Where $\theta = (a_i)$, i = 1..6, is the parameter vector to be estimated.

Motion estimation: To estimate a motion model θ_k we use a gradient-based multi-resolution robust estimation method described in [2]. To ensure the goal of robustness, we minimize an M-estimator criterion with a hard-redescending function [3]. The constraint is given by the usual assumption of brightness constancy of a projected surface element over its 2D trajectory [4]. The estimated parameter vector is defined as:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, E(\theta) = \underset{\theta}{\operatorname{argmin}} \sum_{p \in R(t)} \rho(DFD_\theta(p)) \tag{2}$$

Where $DFD_\theta(p) = I_{t+1}(p + \vec{w}_\theta(p)) - I_t(p)$, and p(x) is a function which is bounded for high values of *x*. The minimization takes advantage of a multiresolution framework and an incremental scheme based on the Gauss-Newton method. More precisely, at each incremental step k (at a given resolution level, or from a resolution level to a finer one), we have: $\theta = \hat{\theta}_k + \Delta\theta_k$. Then, a linearization of $DFD_\theta(p)$ around $\hat{\theta}_k$ is performed, leading to a residual quantity $r_{\Delta\theta k}(p)$ linear with respect to $\Delta\theta_k$:

$$r_{\Delta\theta k(p)} = \nabla I_t(p + \vec{w}_{\hat{\theta}k}(p)) \cdot \vec{w}_{\Delta\theta k}(p) + I_{t+1}(p + \vec{w}_{\hat{\theta}k}(p)) - I_t(p) \tag{3}$$

Where $\vec{\nabla} I_t(p)$ denotes the spatial gradient of the intensity function at location p and at time t. Finally, we substitute for the minimization of $E(\theta_k)$ in (2) the minimization of an approximate expression E_a, which is given by $E_a(\Delta\theta_k) = \sum \rho(r_{\Delta\theta k}(p))$. This error function is minimized using an Iterative-Reweighted-Least-Squares procedure, with 0 as an initial value for $\Delta\theta_k$ [1]. This estimation algorithm allows us to get a robust and accurate estimation of the dominant motion model between two images.

Mixture Gaussian background model: Mixture Models are a type of density model which comprise a number of component functions, usually Gaussian. These component functions are combined to provide a multimodal density [5]. The key idea of background model is to maintain an evolving statistical model of the background, and to provide a mechanism to adapt to changes in the scene. There are two types of background model:

Unimodal model: each pixel is modeled with a single statistical probability distribution (Gaussian distribution) η(X, μt, Σt,), where μt and Σt are the mean value and covariance matrix of the distribution at frame t respectively. Pixels where observed

colors are close enough to the background distribution are classified as background points, while those too far away as foreground points.

Multimodal model: a mixture of multiple independent distributions is necessary to model each pixel. Each distribution is assigned a weight representing its priority. A pixel is classified as a background point only if the color observed matches with one of the background distributions. A new distribution of the observation should be imported into the background model if none of the distributions matches it.

Joint Probabilistic Data Association: The Joint Probabilistic Data Association (JPDA) algorithm considers the problem of tracking T targets in clutter [6]. $x^t(k)$ ($1 \leq t \leq T$) denotes the state vectors of each target t at the time of the *k*th measurement. The target dynamics are determined by known matrices F^t and G^t and random noise vectors $w^t(k)$ as follows:

$$X^t(k+1) = F^t(k)x^t(k) + G^t(k)w^t(k) \tag{5}$$

where t = 1, . . . ,T. The noise vector $w^t(k)$ is stochastically independent Gaussian random variables with zero mean and known covariance matrices. Let m_k denotes the number of validated returns at time k. The measurements are determined by

$$z_l(k) = H(k)x^t(k) + v^t(k) \tag{6}$$

where t =1, . . . ,T, and l =1, . . .,m_k. The H(k) matrix is know, each $v^t(k)$ is a zero-mean Gaussian noise vector uncorrelated with all other noise vectors, and the covariance matrices of the noise vectors $v^t(k)$ are know.

The goal of JPDA is to associate the targets with the measurements, and to update those estimates. The actual association of targets being unknown, the conditional estimate is determined by taking a weighted average over all possible associations. An association for the kth observation is a mapping $a:\{1, \ldots, T\} \rightarrow \{0, \ldots, m_k\}$ that associates the target t with the detection a(t), or 0 if no return is associated with the *t*th target.

Let $\theta_a(k)$ denotes the event that "a" is the correct association for the *k*th observation. And $\hat{x}_l^t(k \mid k)$ denotes the estimate of $x^t(k)$ given by the Kalman filter on the basis of the previous estimate and the association of the *t*th target with the *l*th return. The conditional estimate $\hat{x}^t(k \mid k)$ for $x^t(k)$ given Z^k is

$$\hat{x}^t(k \mid k) = \sum_{l=0}^{m_k} \beta_l^t(k)\hat{x}_l^t(k \mid k) \tag{7}$$

where $\beta_l^t(k) = \sum_{a:a(t)=l} P(\Theta_a(k) \mid Z^k)$ is the conditional probability of the event $\theta^t_l(k)$ given Z^k. The set of probabilities $\beta_l^t(k)$ can be computed efficiently as the permanents of a set of sub-matrices.

Multi-sensor data fusion: The multi-sensor data fusion is defined as the process of integrating information from multiple sources to produce the most specific and comprehensive unified data about an entity, activity or even [7].

Fusion processes are often categorized as low, intermediate or high level fusion depending on the processing stage at which fusion takes place [7].

Low level fusion, also called data fusion, combines several sources of raw data to produce new raw data that is expected to be more informative and synthetic than the

original inputs. *Intermediate level fusion*, also called feature level fusion, combines various features. Those features may come from several raw data sources or from the same raw data. *High level fusion*, also called decision fusion combines decisions coming from several experts. Methods of decision fusion include voting methods, statistical methods, fuzzy logic based methods, and machine learning methods.

3 Overview of the Approach

Figure 1 shows an overview of the method. The proposed algorithm consists of three independent parts. The first part finds the camera motion, and detects the mobile targets in the visible image sequence. The second part detects the mobile target in the infrared image sequence. Each part of the algorithm behaves as an expert, indicating possible presence of mobile targets in the scene; Decision fusion is used to combine the outcomes from these experts.

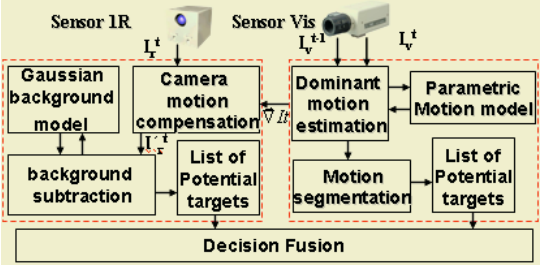


Fig. 1. Overview of the approach.

4 Targets Detection in Visible Images

Mobile objects in the visible image sequences are detected performing a thresholding on the motion estimation error, where the mobile objects are the regions whose true motion vector does not conform to the modeled flow vector.

In [8] is shown through the analysis of the results of different kinds of optical flow estimation algorithms, that $\| \vec{\nabla} \tilde{I}(p) \|^2$ is indeed a proper measure of the reliability of the estimation of the normal flow u_n, thus, the motion error is calculated using the following weighted average, which is proposed in [9]

$$Mes_{\hat{\Theta}t}(p) = \frac{\Sigma_{q \in F(p)} \left(\| \vec{\nabla} \tilde{I}(q) \|^2 \times | FD_t(q) | \right)}{Max(\Sigma_{q \in F(p)} \| \vec{\nabla} \tilde{I}(q) \|^2, n \times G_m^2)} \tag{8}$$

Where F(p) is a small neighborhood around p which contains n points, and G_m is a constant which accounts for noise in the uniform areas. An interesting property of this local measure is the following. Let us suppose that the pixel p and its neighborhood undergoes the same displacement of magnitude δ and direction $\vec{u}$. In [1] there were derived two bounds l(p) and L(p) such that, whatever the direction $\vec{u}$ might be, the following inequality holds:

$$0 \leq l(\mathrm{p}) \leq Mes_{\hat{\Theta}t}(p) \leq L(\mathrm{p}) \tag{9}$$

The bounds used in the experiments are given by:

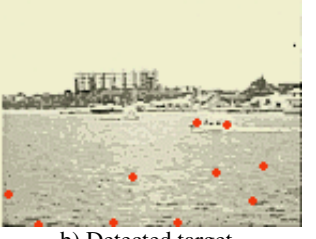
$$\begin{cases} I(p) = \eta\delta\sqrt{\lambda'_{\min}(1-\lambda'_{\min})} \\ L(p) = \delta\sqrt{1-\lambda'_{\min}} \end{cases} \text{ with } \eta = \frac{\Sigma_{q\in F(p)} \|\nabla\tilde{I}(q)\|^2}{Max(\Sigma_{q\in F(p)} \|\nabla\tilde{I}(q)\|^2, n\times G_m^2)} \text{ and } \lambda'_{\min} = \frac{\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}$$

Where λ_{min} and λ_{max} are respectively the smallest and highest eigenvalues of the following matrix (with $\tilde{I}(q)$ = Image at time q and $\nabla\tilde{I}(q) = (\tilde{I}_x(q), \tilde{I}_y(q))$:

$$M = \begin{pmatrix} \Sigma_{q\in F(p)}\tilde{I}_x(q)^2 & \Sigma_{q\in F(p)}\tilde{I}_x(q)\tilde{I}_y(q) \\ \Sigma_{q\in F(p)}\tilde{I}_x(q)\tilde{I}_y(q) & \Sigma_{q\in F(p)}\tilde{I}_y(q)^2 \end{pmatrix} \tag{10}$$

Figure 2 shows the results of the target detection method in the visible sequence in presence of one target.

(a) Visible image sequence.

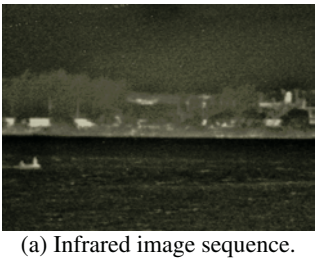


b) Detected target.

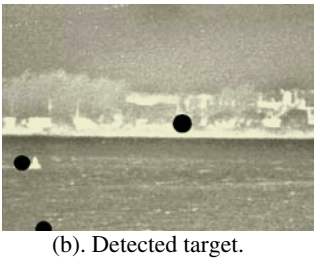
Fig. 2. Motion segmentation results in the Ship sequence.

5 Targets Detection in Infrared Images

Mobile objects in the infrared image sequences are detected determining the background in the image, and subtracting it to the original image, which has been preprocessed to eliminate the camera motion, this preprocessing step use information of the dominant motion calculated in the last module.



(a) Infrared image sequence.



(b). Detected target.

Fig. 3. Background model results in the boat sequence.

The background is obtained using a statistical model to classify pixels (see section 2). Each pixel is modeled as a mixture of 3 Gaussian models [10]. This process has three main stages:

1. Gaussian model initialization.
2. Background detection.
3. Update of the background estimation.

Figure 3 shows the results of the target detection method in the infrared sequence in presence of one target.

6 Decision Fusion

The first and second parts of the approach behave as experts indicating the possible position of mobile targets in the scene. The final decision is reached by fusing the results of these experts.

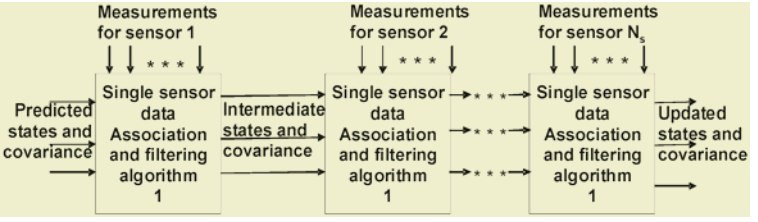


Fig. 4. Multi-sensor Fusion Architecture.

Figure 4 shows the sequential Multi-Sensor Data Fusion architecture [11] used to combine the individual target detecting results. The initial state of the tracking algorithms is obtained using a weighted "k out of N" voting rule. The combination of the measurements is done; making Ns (Number of sensors in the system) repetitions of the JPDA algorithm (see section 2).

The fusion algorithm works on the basis of the following equations.

Let m_{ki}, i = 1, 2, . . . ,Ns, the number of validated reports from each sensor i at time k. The measurements are determined by

$$z_l^i(k) = H_i(k)x^t(k) + v_i^t(k) \tag{11}$$

where t=1, ... ,T, i =1, ... ,Ns, and l =1, ..., m_{ki}. The measurement $z_l^i(k)$ is interpreted as the lth measurement from the ith sensor at time k. Generalizing from the single-sensor case, the $H_i(k)$ matrices are known, and $v_i^t(k)$ are stochastically independent zero-mean Gaussian noise vectors with known covariance matrices. The observation at time k is now

$$Z(k) = (z_1^1(k),\ldots,z_{mk1}^1(k),z_1^2(k),\ldots z_{mk1}^2(k),\ldots,z_1^{Ns}(k),\ldots,z_{mk1}^{Ns}(k)) \tag{11}$$

The conditional estimate of the fusion algorithm is given by:

$$\hat{x}^t(k \mid k) = \sum_L \beta_L^t(k)\hat{x}_L^t = \sum_L \prod_{i=1}^{Ns} \beta_L^t(k)\hat{x}_L^t \tag{13}$$

Where the sums are over all possible sets of associations L with target t.

7 Results

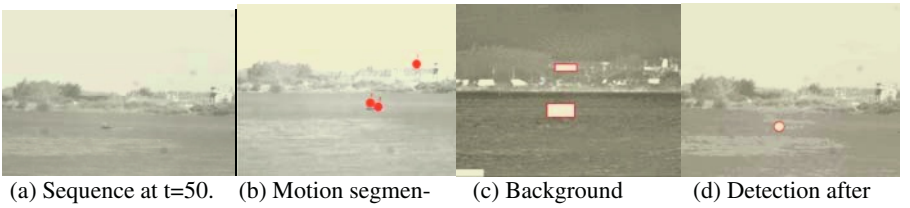
In this section, we will show the experimental results of our approach. The algorithm was tested with a database of two multi-spectral image sequences. The weather conditions were: winds of 30 to 70 km/hour, and variable lighting. The boat sequence was used to characterize the results of the motion segmentation algorithm.

Table 1 shows the principal features and results of the two first blocks. In the table, Pd is the probability of detection and NFt is the average number of false targets per image. The figures 5(a) and 6(a) show an image of the Boat and People sequence

respectively. By applying the algorithms described in section 4 and 5 the objects are detected in each frame, figures 5(b) and (c) show the target detection results using the Boat sequence, figure 6(b) and (c) show the target detection results using the People sequence.

Table 1. Results of different experts.

Sequence	Size	Frames	Targets	Sensor	Pd (%)	NFt
Boat	640 x 480	150	1	Visible	96	2.0
				Infrared	100	1.5
People	640 x 480	150	2	Visible	99	1.5
				Infrared	98	0.6
Ship	640 x 480	50	2	Visible	95	5.2



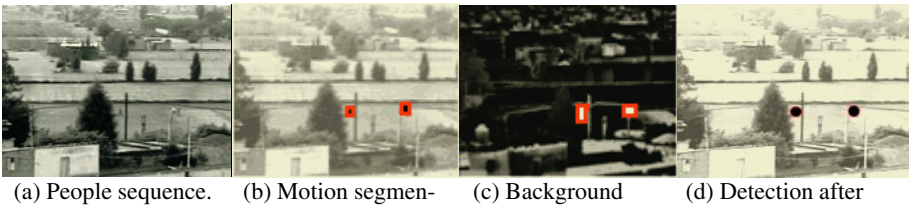
(a) Sequence at t=50. (b) Motion segmentation result. (c) Background model result. (d) Detection after fusion

Fig. 5. Target detection results in the boat sequence.

In table 2, results after the decision fusion are shown. In both sequences, the fusion improves results. The data association step in the fusion module reduces the number of false targets creating gating regions and considering just the measurements that fall in that region. The fusion module improves the target state estimation by processing sequentially the sensors detection, in this module if an target was not detected in a sensor, the information about it stays and the following sensor is processing, this way to combine the information improves the probability of detection, because the target must be loosed in all sensors to lose it in the fusion decision result. Figure 5(D) and 6(D) shows these results graphically.

Table 2. Results after fusion.

Sequence	Processing average time	Pd (%)	NFt
Boat	4.3 seg.	100	0.5
People	4.1 seg.	99	0.1



(a) People sequence. (b) Motion segmentation result. (c) Background model result. (d) Detection after fusion

Fig. 6. Target detection results in the people sequence.

8 Conclusions

In this paper an approach to improve target detection process using decision fusion is proposed. The approach was tested using multi-spectral image sequences from moving cameras. Experimental results show that targets detection algorithms detects in average 97% of the targets in the worse case, and in the better one detects 99.5%. The fusion module detects in the worst case 99% of the targets and 100% in the better one, while the 90% of the false targets are removed. This results show the advantages of this approach for automatic detection and tracking. It has been shown that this approach performs better that either tracker in isolation. Most importantly the tracking performance is improved without specialization of the tracking algorithms for a specific task; it remains to develop an algorithm to handle target occlusion and to reduce the processing time.

References

1. J. Odobez, P. Bouthemy. Direct incremental model-based image motion segmentation analysis for video analysis Signal Processing. Vol 66, pp 143-155, 1998
2. J. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. JVCIR, 6(4) pp 348-365, 1995.
3. P.J. Hubert. Robust statistics. Wiley, 1981.
4. Horn, Shunck. Determining optical flow. Artificial Intelligence, vol 17 pp 185-203, 1981
5. C. Stauffer, Adaptive background mixture models for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 246-252, 1999.
6. Bar-Shalom, T. Fortmann. Tracking and data association, Academic Press, San Diego, 1988.
7. E. Waltz and J. Llinas, Handbook of multisensor data fusion, CRC Press, 2001.
8. J. Barron, D Fleet, S. Bauchemin. Performance of optical flow techniques. International Journal of Computer Vision. 12(1) pp 43-77, 1994.
9. M. Irani, B. Rousso, S. Peleg. Computing occluding and transparent motion. Intern. J. Comput. Vis. 12(1) pp 5-16, 1994.
10. C. Stauffer, W. E. L. Grimson, Learning patterns of activity using real time tracking. IEEE trans. PAMI, val 22, no. 8, pp 747-757, Aug, 2000.
11. L. Pao, S. O'Neil. Multisensor Fusion algorithms for tracking. Proc. of American Control Conference. pp. 859–863, 1993.