

**Título** Concatenating multiple trajectories using Kalman filter for pedestrian tracking

---

**Tipo de Producto** Ponencias (texto completo)

---

**Autores** Negri, Pablo y Garayalde, Damián

---

**Código del Proyecto y Título del Proyecto**

---

D14T01 - Análisis del estado del arte en prótesis de miembro superior.  
(Tecnologías de músculos artificiales aplicadas en prótesis de miembro superior – Parte I)

---

**Responsable del Proyecto**

---

Garayalde, Damián

---

**Línea**

---

Automatización y Sistemas de Control

---

**Área Temática**

---

Dinámica Industrial /Automatización y Robótica

---

**Fecha**

---

Junio 2014

---

# Concatenating multiple trajectories using Kalman filter for pedestrian tracking

Pablo Negri<sup>†\*1</sup> and Damian Garayalde<sup>†</sup>

<sup>†</sup>*Instituto de Tecnologia - UADE, Buenos Aires, Argentina*

<sup>1</sup>*pnegri@uade.edu.ar*

<sup>\*</sup>*CONICET, Buenos Aires, Argentina*

**Abstract**—This work proposes a tracking-by-detection methodology for pedestrians following using a targlet framework. In this framework each pedestrian is considered as an autonomous agent, denominated targlets, and modeled with a state machine. The tracking procedure is initialized by a people detector computed on a Movement Feature Space. Detector outputs generate probabilistic fields employed for the tracking of each targlet, in order to obtain their individual trajectories along the sequence. The targlet framework is then analyzed off-line to: filter false positives, and concatenate broken trajectories of the same person using a Kalman filter approach. The system is tested on the public dataset PETS2009 S2.L1 obtaining good results, similar to the best methodologies in the state of the art.

## Resumen—

Este trabajo propone el seguimiento-por-detección aplicado a peatones usando un esquema basado en *targlets*. En este esquema, cada peaton se considera una agente autonomo, denominado targlet, y a su vez es modelado como una maquina de estados. El seguimiento comienza por el uso de un detector calculado en el Movement Feature Space. La salida de los detectores genera campos probabilísticos que son usados para el seguimiento de cada *targlet* construyendo su trayectoria individual. Un análisis off-line del sistema permite el filtrado de falsos positivos y la asociación de diferentes targlets pertenecientes a la misma persona usando el filtrado de Kalman. El sistema fue testeado en una base de imágenes pública, el PETS2009 S2.L1, obteniendo buenos resultados, similares a los mejores trabajos del estado del arte.

## I. INTRODUCTION

Modern pedestrian detection systems have adopted a methodology based on Tracking-by-Detection [1]–[6]. It involves the use of a pedestrian detector applied on individual frames in the first step. Then, those detections are employed as data source to construct trajectories across the frames of the sequence. This approach is robust to changing backgrounds, to moving cameras, and to the presence of others objects in movement. Tracking-by-Detection algorithms can be classified as off-line or on-line tracking.

Off-line tracking involves a first step of pedestrian location hypotheses generation evaluating the overall sequence. Then, further analysis are applied to construct the trajectories. In [7], they run two detectors, one based on SVM classification of histogram of oriented gradients (HOG) [8], and the other uses relative optical flow (HOF) [9]. Individual trajectories are represented in the continuous space as cubic B-splines. The off-line analysis of the trajectories includes an iterative algorithm which solves a multi-labeling problem minimizing a continuous energy function. In the work of Leibe et al. [1], they construct a first hypotheses of trajectories in the detection step with a procedure based on

the concept of event cones. Each event cone establishes the spacetime volume of influence for a detection hypotheses that is used to the association with the trajectories. An posterior iterative selection algorithm defines which trajectories explain best the obtained data.

On-line methodologies must run in real-time, minimizing false positives, and solving different tracking problems, i.e. lost pedestrians. Benfold and Reid [4] propose to solve the detection step combining a head and a body detector. The tracker employs the Lucas Kanade algorithm (LKT) [10] to estimate the motion in the frame, and an update similar to the Kalman filter is used to construct the trajectory. The work of Breitenstein et al. [3] combines an on-line training for each pedestrian with a particle filter in charge of the tracking. The particles are propagated combining different sources of information, as associated detections, intermediate outputs of the object detector, and the classifier.

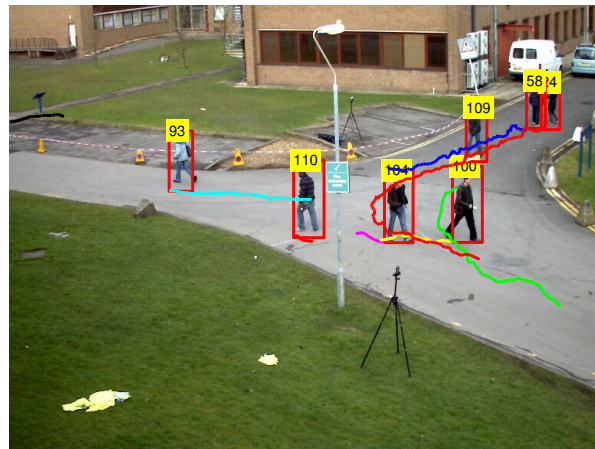


Fig. 1. Tracking sample on the PETS2009 sequence using our framework.

In this work, we construct on-line a *targlet* framework from detections obtained by a classifier running on the Movement Feature Space (MFS) [11]. The term *targlet* is inspired by the mixture of the words *target* and *aglet*. The last is an old programming entity defined as a Java-based autonomous software agent. The framework associates one targlet to one pedestrian since his first view on the scene until he goes out of sight. It keeps position and motion information of the previous frames, helps its association with new hypotheses computed by the detector, and gives an activity context to perform the tracking. In this way, each targlet is modeled as a state machine. Figure 1 shows a sample of the targlet framework on the PETS2009 public

dataset. The ID of each targlet is shown as the label on the top of the rectangle over the pedestrian, and some trajectories are also displayed. The information collected over the sequence will be analyzed off-line by our algorithm to filter false alarms, and associate targlets using a Kalman filter based methodology.

The main contributions of this works consist of the construction of the targlet framework, the development of the state machine for each targlet, and the trajectory concatenation procedure based on the Kalman filter.

The paper is organized as follow. Next section details the detection and tracking procedure on the MFS, and the state machine associated to each targlet. Section III develops the procedures pruning the targlet framework in order to improve the results. Test results are presented on section IV to finish with the conclusions of the paper.

## II. TARGLET FRAMEWORK

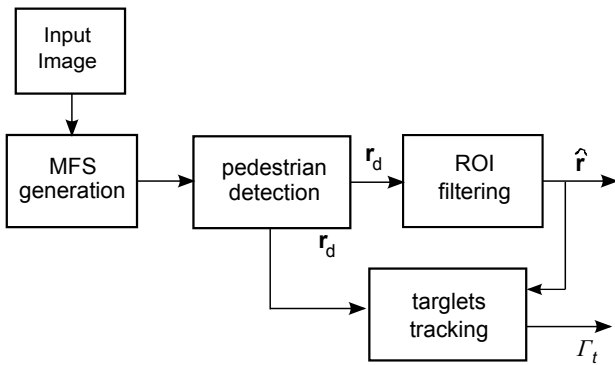


Fig. 2. Block diagram of the pedestrian tracking system and the Targlet Framework.

The methodology presented in this work is shown in Fig. 2. The input image is projected into the MFS where the pedestrian detector [11] is applied. Then, the list of detections are fed to the targlet tracking framework. The output of the system consists on the pedestrian hypotheses  $\hat{\mathbf{r}}$  and the set of targlets  $\Gamma_t$  at time  $t$ .

### A. Pedestrian detection on the MFS

The MFS is an adaptive motion extraction model. It uses level lines and their orientations as descriptors to generate a background model. The motion in the frame at time  $t$  is encoded in two arrays:  $S_t$  and  $O_t$ . The matrix  $S_t(p)$  counts the number of moving level lines that pass through the pixel  $p$ , and  $O_t(p)$  indicates the orientation of the level lines.

Fig. 3 presents an example of the pedestrian detector output. Hypothesis generation step (HG) uses a cascade of boosted classifiers and their validation (HV) is assumed by a SVM classifier. Both classifiers encode information of the MFS as histograms of oriented level lines (HO2L) (see [11] for more details).

The output of the HV step is a list of rectangles and their associated confidence score  $\mathbf{r}_d = \{(r_i, s_i)\}_{n_i=0, \dots, n-1}$ . Each region of interest (ROI or roi) is defined as  $r_i = [x_i^c, y_i^c, w_i, h_i]$ , where  $(x_i^c, y_i^c)$  is its central position and  $(w_i, h_i)$  are the width and height. A Non Maxima Suppression (NMS) filtering method is applied on  $\mathbf{r}_d$  to determine the estimated pedestrian positions  $\hat{\mathbf{r}}$ .

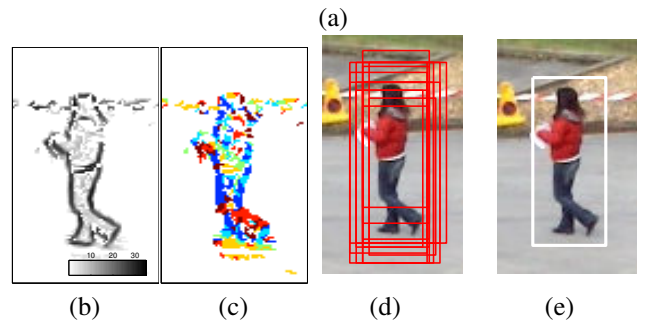
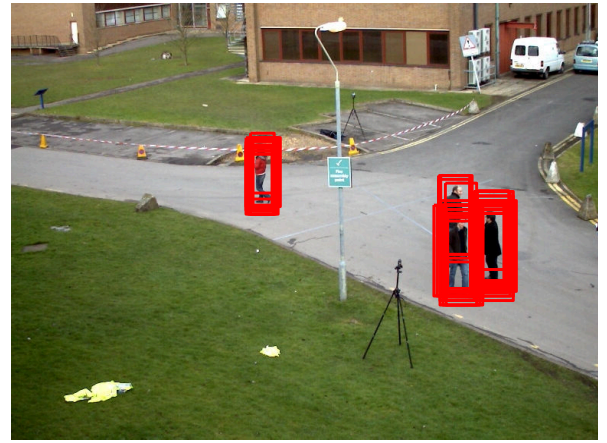


Fig. 3. Pedestrian detector, (a) shows the set  $\mathbf{r}_d$  of detected rectangles, (b) and (c) presents the MFS information of the  $S_t$  and the  $O_t$  arrays respectively, (d) are the detections over one person, and (e) is the estimated pedestrian position  $\hat{\mathbf{r}}$ , product of the filtering of  $\mathbf{r}_d$ .

### B. Detections and targlets association

The targlet framework  $\Gamma_t$  at time  $t$  consist of a set of targlets:  $\Gamma_t = \{T_1, \dots, T_n\}$ . Each targlet contains the following information:  $T_{i,t} = \{\mathbf{r}_t, id, e_t, \mathbf{m}_t\}$ , where  $\mathbf{r}_t$  is the roi associated  $\mathbf{r} = \{x, y, w, h\}$ ,  $id$  is a number identifying the targlet,  $e_t$  is the state of  $T_i$  at time  $t$ , and  $\mathbf{m}_t$  is the motion history composed of the last 5 motion vectors:  $\mathbf{m}_t = \{\mathbf{h}_{t-4}, \dots, \mathbf{h}_t\}$ .

The association of the estimated pedestrian positions  $\hat{\mathbf{r}}$  to the targlet framework  $\Gamma_t$  works as follows: first, for each pair  $(r, T)$ , with  $r$  a detection and  $T$  a targlet, the vector defined by the new detection  $r$  and the roi of  $T$  is computed. Let define  $d$  and  $\theta$  as the module and the angle of the vector. This angle is compared against the angle of the average vector  $\bar{m}$  calculated using the motion history of the targlet.

The distance  $d$  and the difference between  $\theta$  and  $\angle(\bar{m})$  are evaluated by specifics thresholds that depend on the state  $e_t$  of the targlet. The values of the thresholds are chosen by the assuming there are only slow changes in the dynamic of each pedestrian. The pairs that not fulfill these requirements are eliminated. The pair with the best match is retained and updates the description of the targlet dynamics. Detections without a targlet match generate a new targlet in the framework. Targlets without a detection match are considered as lost.

### C. Tracking

The tracking of a pedestrian consists of locating its position on the current frame  $F_t$ , starting from its initial position at the preceding frame  $F_{t-1}$ .

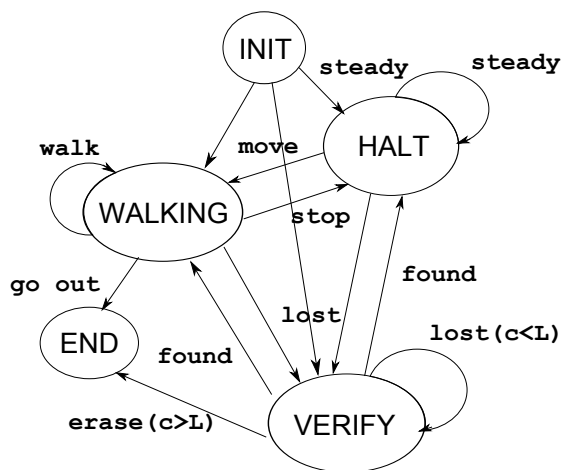


Fig. 4. Individual target state machine.

The tracking is carried on by two types of tracking fields: an object confidence field and an appearance field. The former uses the detector output scores to construct a continuous map of the likelihood to find a person in an image location. However, due to the real conditions of the working sequences, partial (or total) occlusions of the pedestrians usually happens, or the scene shows strong lightening changes, or the changing background camouflages the people. In these cases, the appearance field based on a corner extraction on the MFS encodes singular pedestrian features and is robust to track the target with partial information.

Using the probability fields, the tracking procedure follows a combination of the Mean Shift algorithm [12], and the pyramidal LKT approach proposed by Bouguet [13]<sup>1</sup>.

#### D. State machine

The Targlet Framework models each pedestrian as an autonomous agent with a state machine. Throughout the target lifetime, from the first capture that it appears in the sequence, until it exits the view, the tracking system collects information of this evolution generating events that trigger transitions between the states.

Figure 4 shows the states of a targlet and the events generating the transitions to the following states.

**INIT STATE:** a detection  $\text{roi } \hat{r}_t$  which can not be associated to an existing targlet in the framework generates a new targlet with the INIT state. In the next frame, the association stage searches the corresponding detection  $\hat{r}_{t+1}$ . If it exists, the detection field is employed in the tracking procedure to generate a motion vector  $\mathbf{h}$ . Two events are then associated: if the module of  $\mathbf{h}$  is lower than a threshold, the event *steady* triggers the transition to the HALT state. Otherwise, the event *move* is generated and the new state of the targlet will be WALKING. When there is no detection associated to the targlet, the *lost* event is generated and the new state will be VERIFY.

**HALT STATE:** in this state the event *steady* does not trigger a state transition. The event *move* shows that the targlet initiates a movement, and the targlet has a transition

to the WALKING state. A *lost* event triggers a transition to the VERIFY state.

**WALKING STATE:** in this state the target is on continuous movement, generating motion vectors  $\mathbf{h}$  with module higher than the threshold. The target  $T_1$  on fig. 5 correctly tracks the pedestrian  $p_1$ . To generate a *stop* event in order to trigger to HALT state, the average of the last three motion vectors is calculated:  $\mathbf{h}_{t-2}$ ,  $\mathbf{h}_{t-1}$  and  $\mathbf{h}_t$ . If the module of the average is lower than the threshold, then the *stop* event is generated. This procedure helps to filter some tracking errors and the transitions to the events are generated smoothly. If the target goes beyond the limits of the scene, a *go out* event is generated and the transition is to the state END.

**VERIFY STATE:** this state is triggered when it is not possible to associate a roi detection to the target and the *lost* event is generated. The transition to this state initializes a counter  $c$  of the times that the target is lost. In this state, the pursuit of the target is made of using a corner detector computed on the MFS being robust to partial occlusions or other kind of distracting sources for the pedestrian detector. If the target can be associated to a detection in the next frames, there is a transition to the other states depending on the motion vector. But if the counter  $c$  arrives to a  $L$  value, an *erase* event is generated and the state machine changes to the END state. Fig. 5 shows two examples of that case. The target  $T_3$  is an example of a lost pedestrian, and the target  $T_2$ , which was generated by a false alarm  $f$ , was closed immediately because  $f$  was the only detection associated.

**END STATE:** this state closes the target.

The advantage of the targlet framework is that its activity information can be exploited by the tracking system, i.e. in the data association stage it is very useful to know the dynamic of the target in order to reject some hypothesis. A future extension of this framework is the analysis of the events which can incorporate information about other objects in the scene, as other pedestrians, vehicles, traffic lights, etc. In this way it is possible to generate some kind of behavioral model.

### III. TARGLET FRAMEWORK PRUNING

This section develops the off-line evaluation of the Targlet Framework along the sequence. All the targlets are analyzed in order to filter false alarms, and to associate broken tracks.

#### A. False Positives filtering

False Positives (detections that are not pedestrians) can be easily detected if the only states of the corresponding targlet where INIT and VERIFY i.e. targlet  $T_2$  in fig. 5. It means that there were only one detection  $\hat{r}$  associated to this region of the image, possibly generated by the light conditions, shadows, etc. To filter that cases, the targlet is eliminated if:  $\#NV > \#NH + \#NW$ , where  $\#NV$ ,  $\#NH$ ,  $\#NW$  is the number of times the targlet has VERIFY, HALT and WALKING states respectively.

#### B. Targlet concatenation with Kalman filter

In general, a pedestrian is lost if: the detector misses him (i.e. targlet  $T_3$  in fig. 5), or the tracking algorithm diverges because of the presence of other pedestrians, noises or an

<sup>1</sup>The details of the methodology are beyond the purpose of this paper. However, for further information the reader can contact the authors.

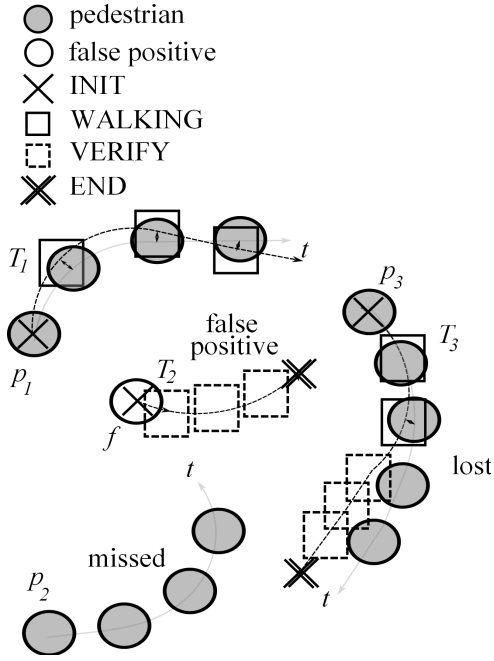


Fig. 5. This figure presents various examples on the tracking of pedestrians and the behavior of the Targetlet Framework. In this schema, real pedestrians  $p_i$  are represented by filled circles, false positives as empty circles, and the targetlets  $T_i$  are presented by their different states.

abrupt change in the movement direction. In both cases, the targetlet has a transition to the VERIFY state for some frames and it is finally closed when it is not associated to a pedestrian again. However, the detector continues his job and detects the pedestrian, in his new position, which is not necessarily in the vicinity of the targetlet in VERIFY state. As was explained before, a new targetlet is created and starts to follow the pedestrian.

The concatenation procedure will connect those targetlets and give them the same ID. A Kalman filter is employed to perform this task. It estimates a time-varying state vector  $X_t$  from the noisy measurements  $V_t$  consisting of the position of the targetlet at time  $t$ . The model of our application following the system dynamics and the measurements is:

$$X_{t+1} = \Phi_{t+1,t} X_t + w_t \quad (1)$$

$$V_t = H_t X_t + v_t \quad (2)$$

where  $w_t$  and  $v_t$  are assumed to be normally distributed with zero mean and covariances  $Q_t$  representing the model error, and  $R_t$  the covariance of the measurement error.  $\Phi_{t+1,t}$  is the evolution matrix, and  $H_t$  the selection matrix.

The filter computes the optimal estimate  $\hat{X}_t$  and the associated uncertainty  $P_t$  of the state vector  $X_t$  recursively from: the data  $V_0, V_1, \dots, V_t$ , and the initial estimation  $E(X_0)$  and  $Var(X_0)$  [14], [15].

The matrix used by the filter are the following:

$$V = \begin{pmatrix} x_m \\ y_m \end{pmatrix} \quad X = \begin{pmatrix} x_m \\ \dot{x}_m \\ y_m \\ \dot{y}_m \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \Phi = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In that way, the filter provides a reasonable estimate of a region where the pedestrian should be detected when the targetlet has a transition to the VERIFY state, using the covariance matrix  $P$ . If a targetlet is initiated inside this region, they can be concatenated.

To implement this methodology all the positions in the image are projected to real world coordinates using the calibration data provided with the sequence.

The state vector position of a targetlet is the position of the left inferior corner of their roi  $T_i.roi$  projected to the real world coordinates  $T_i.x = [x^w, y^w]$ , assuming that the evolution of the pedestrian is at  $Z=0$ . At the initiation of the targetlet, the state vector takes as initial value its first position  $X_0 = T_i.x_0$ , and the covariance matrix  $P_0$  takes high initial values. The Kalman filter evolution will follow the position of the targetlet throughout the sequence updating and saving their components.

If the targetlet changes its state to VERIFY at time  $t_v$  and their position in the scene is far away the exit points [16], the associated pedestrian was lost. The targetlet can find the pedestrian in the next  $L$  frames using the appearance tracking based on the corners. If the targetlet does not recovers the track, it is closed. In that case, the concatenation algorithm 'rewinds' the targetlet framework to the frame at time  $t_v$  and starts the matching procedure.

Let be  $T_0$  the tracked targetlet that is now in VERIFY state. Let  $T_{i=1,\dots,n}$  be all the other targetlets in the framework at time  $t_v$ . The matching procedure evaluates the set  $T_{i=1,\dots,n}$  to identify the possible position of the lost pedestrian. The research area is determined using the result of the Kalman filtering of  $T_0$  and the procedure is conducted using the Mahalanobis distance [14]:

$$d_i(T_0, T_i) = (X^{T_0} - V^{T_i})^T (S)^{-1} (X^{T_0} - V^{T_i})$$

the covariance  $S$  of the difference  $(X^{T_0} - V^{T_i})$  is the sum of the covariance matrix  $P^{T_0}$  and  $R$ . The distance has a  $\chi^2$  distribution with one degree of freedom. From the  $\chi^2$  distribution table, the distance  $d_i$  is thresholded by the value 3.84, in order to have the 95% probability to found the pedestrian.

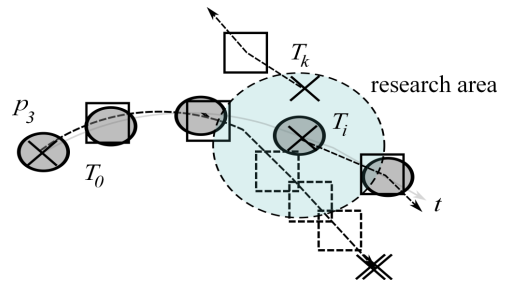


Fig. 6. The figure shows the research area around the pedestrian  $p_a$  position, the targetlet  $T_0$  which is closed, and the candidates targetlets  $T_i$  and  $T_k$  for the matching procedure.

Figure 6 presents the matching procedure. The research area defined by the Mahalanobis distance is designed by the circle. There are two targetlets candidates  $T_i$  and  $T_k$  initiating inside the area. In order to proceed with the concatenation, the other condition to fulfill is that the evolution of the targetlet should follow the displacement of the matched targetlet  $T_0$ . This can be easily computed using the displacement vector of  $T_0$  until time  $t_v$ , and the displacement vector of the targetlet candidate taking into account their position at time  $t_{v+1}$ . Targetlet  $T_k$  is not retained because it has a different evolution. Then, targetlet  $T_i$  is retained and concatenated with  $T_0$ , updating its ID ( $T_i.id = T_0.id$ ).

#### IV. RESULTS

##### A. Datasets

The system was evaluated on the PETS2009 dataset [17], from which it was used the task S2.L1 view 1. The sequence is composed of 795 frames with 4650 annotated rois positions corresponding to 19 pedestrians. Pedestrian does not follow exactly a natural movement because they change their trajectory many times, and sometimes abruptly. There is a small region on the center of the sequence where a signal board occludes the pedestrians.

##### B. Experiments

Figure 7 shows the result of the concatenation procedure of the targetlet 104. In column (a) the trajectories of targetlets 104, 128, 146 and 155 are depicted. Please note the presence of a kind of *tail* on their path before the closing. This corresponds to the positions where the targetlets have VERIFY state, and loose the pedestrian track. This tails were filtered on the Kalman concatenated trajectory, shown in the column (b) of fig. 7.

The concatenated targetlet 104 tracks the pedestrian until it is occluded behind the sign of the light. The concatenation procedure presents good results, tracking the pedestrian for 184 frames supporting abrupt changes in the movement, and some occlusions behind the sign and other pedestrians.

Performance test of the targetlet framework conducted on the datasets were calculated using CLEAR MOT metrics [18]. The scores are:

- The multiple object tracking precision (MOTP) measures the total error of the estimated position for matched pedestrian-targetlet pairs. For the precision MOTP, as in [3], it was considered the score of 50 % as significant for the tracking, the same as the Pascal VOC Challenge [19].
- The multiple object tracking accuracy (MOTA) accounts for the errors made by the tracker as false negative rate, false positive rate, and number of identity switches.
- Identity switches (SWIDs) identifies when a new targetlet is assigned to a pedestrian. In [18] this score is calculated by comparing the correspondence map between pedestrians and targetlets of time  $t$  with the map at time  $t - 1$ . In that way, if a pedestrian was lost on the previous frame, it cannot be found on the  $t - 1$  map, and a new targetlet assigned to this pedestrian is not considered as an ID switch. For our tests, the comparison of the correspondences were extended until

finding the previous targetlet ID that matched with the pedestrian. It is for that reason that our scores have higher values in this item.

- False Negatives (FN) scores the ratio of missed pedestrians. This score is closely related by the performance of the detector.
- False Positives (FP) measures the ratio of targetlets rois not matched to a pedestrian position.
- Number of Targetlets, counts the total number of targetlets on the framework along the sequence.

Table I reports the scores of the analysis in the targetlet framework. The first row, Raw Framework, corresponds to the output of the complete framework on the sequence. As it can be seen, the framework created 355 targetlets. Many of those targetlets corresponds to false positives, and shows a score of 45.7 %. The number of switches ID is very important.

Raw results of the Targetlet Framework are shared at <http://pablonegri.free.fr/Downloads/ARGENCON2014.htm>, if the reader wants to reproduce the results using our methodology.

In the second row, Filtered Raw Framework, the false positive filtering (sec. III-A) was applied to the framework. The number of meaningful targetlets is 128, and the false positive ratio shows a remarkable reduction to 18.8%.

Fourth row, Raw DKS Framework, presents the results of the framework on which it was applied the concatenation procedure using the Kalman filtering. The number of targetlet was reduced by 90, and the difference represents those targetlets that were concatenated. Besides, the improvement obtained by the concatenation procedure can be seen on the number of switches IDs, which consist in a reduction of the 50%.

Filtered Raw DKS Framework represents the result of applying the concatenation procedure followed by the false positive filtering. The number of total targetlets is now 80 and it has the best scores on MOTA, FP and SW ID. These are good results, and the analysis of the cases of SW IDs shows that the majority occurs when there are occlusions of the pedestrians. The concatenated procedure is limited on the number of frames that looks for the lost targetlet, and sometimes, the pedestrian remains occluded beyond this number and he is finally lost by the targetlet association algorithm.

The last rows in table I show results reported in the literature on the same dataset. Both works have better scores on the MOTA results, which is normal considering their complex analysis performed on the detected path [7], and the detection methodology based on on-line learning [3] which is robust against occlusions. This shows us some clues that can be explored to improve our performance in future works.

#### V. CONCLUSIONS

This article introduces a targetlet framework applied to pedestrian tracking on video sequences. Each targetlet consists in an autonomous agent modeled as a state machine that tracks a pedestrian hypothesis. It is proposed a Detection-by-Tracking procedure executed on the MFS that captures the information within the images and feeds the framework. An off-line analysis of the targetlet framework is presented

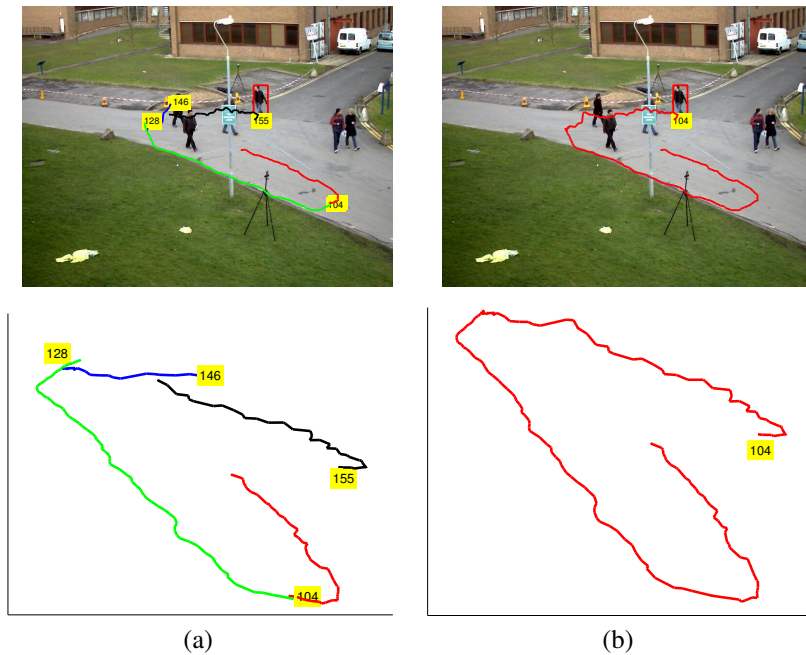


Fig. 7. The figure presents the result of the concatenation procedure on a set of targets. First row shows the path on the scene and second row show the path in real world coordinates. The labels indicate the ID number of the targets at the end of their path.

Target Framework	Nro. Targ	MOTP (%)	MOTA (%)	FN (%)	FP (%)	SW id
Raw Framework	355	79.8	43.6	7.3	45.7	150
Filtered Raw Framework	128	<b>73.0</b>	68.5	10.1	18.8	113
Raw DKS Framework	267	79.8	44.6	7.3	45.7	105
Filtered Raw DKS Framework	<b>80</b>	71.30	<b>71.0</b>	11.1	<b>16.2</b>	<b>71</b>
Breitenstein et al. [3]	-	53.6	79.7	-	-	-
Andriyenko et al. [7]	-	95.9	78.7	-	-	10*

TABLE I

EVALUATION SCORES RESULTS, PRECISION (MOTP), ACCURACY (MOTA), FALSE NEGATIVE RATE (FN), FALSE POSITIVE RATE (FP), AND THE NUMBER OF ID SWITCHES. (\*) WERE COMPUTED FOLLOWING [18].

allowing the pruning of the results in order to filter false positives and concatenate associated targetlets. The experiments conducted on a public dataset show promising results using this simple methodology, which are comparable to the best algorithm of the state of the art.

#### ACKNOWLEDGMENT

This work was supported by PICT-2283 of ANPCyT, ACyT R12T03 of UADE and CONICET (Argentina).

#### REFERENCES

- [1] B. Leibe, K. Schindler, N. Cornelis, and V. G. L., "Coupled object detection and tracking from static cameras and moving vehicles," *PAMI*, vol. 30, no. 10, pp. 1683–1698, 2008.
- [2] S. Stalder, H. Grabner, and L. Gool, "Cascaded confidence filtering for improved tracking-by-detection," in *ECCV*, 2010, vol. 6311, pp. 369–382.
- [3] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *PAMI*, vol. 33, no. 9, pp. 1820–1833, 2011.
- [4] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *CVPR*, 2011, pp. 3457–3464.
- [5] B. Yang and R. Nevatia, "Online learned discriminative part-based appearance models for multi-human tracking," in *ECCV*, 2012, vol. 7572, pp. 484–498.
- [6] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *CVPR*, June 2012, pp. 1815–1821.
- [7] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *CVPR*, 2012, pp. 1926–1933.
- [8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, 2005, pp. 886–893.
- [9] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *CVPR*, 2010, pp. 1030–1037.
- [10] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, August 1981, pp. 674–679.
- [11] P. Negri, N. Goussies, and P. Lotito, "Detecting pedestrians on a movement feature space," *Pattern Recognition*, vol. 47, no. 1, pp. 56–71, 2014.
- [12] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *CVPR*, vol. 2, 2000, pp. 142–149.
- [13] J. Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Intel Corporation, Tech. Rep. 5, 2001.
- [14] R. Deriche and O. D. Faugeras, "Tracking line segments," in *ECCV*, 1990, pp. 259–268.
- [15] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.
- [16] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *CVPR*, June 2012, pp. 1918–1925.
- [17] (2009, June) Pets2009. [Online]. Available: <http://www.cvg.rdg.ac.uk/PETS2009/a.html>
- [18] K. Bernardin and R. Stiefelwagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, 2008.
- [19] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 8, no. 2, pp. 303–338, 2010.