INCREMENTAL DETECTION AND TRACKING OF MOVING OBJECTS BY OPTICAL FLOW AND A CONTRARIO METHOD

Dora Luz Almanza-Ojeda, Michel Devy and Ariane Herbulot
CNRS, LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS-CNRS, F-31077 Toulouse, France

Keywords: Moving obstacles, Detection, Tracking, Clustering, Monocular vision.

Abstract: This paper concerns moving objects detection and tracking based on the a contrario theory and on a Kalman filtering process. Only visual information is acquired from a B&W camera embedded on a mobile robot. KLT and a contrario theory are used to initially detect and cluster moving points. Then, each detected group of moving points is tracked as a moving object using Kalman Filter. The process detection-clustering-tracking is executed in an iterative way to deal with some challenges for real robot navigation. Furthermore, the area in which a moving obstacle is detected, is enlarged in the time until its real limits: clusters are fused with already detected objects considering similarities about their respective velocities and positions. Experimental results on real dynamic images acquired from a camera mounted on a moving robot, are presented and discussed.

1 INTRODUCTION

One key function required for autonomous robot navigation, must cope with the detection of objects close to the robot trajectory, and the estimation of their states. This function has been studied by the robotic and the Intelligent Transportation Systems communities, from different sensory data. For driver assistance, many contributions concern laser-based obstacle detection and tracking (Vu and Aycard, 2009). Some works have made more robust the approach from the fusion with monovision (Gate et al., 2009). But in spite of numerous contributions, this function still remains a challenge when it is based only on vision. So, this work concerns the detection of mobile objects from images acquired from a robot moving in an outdoor environment . It is proposed to reach this objective, using only a monocamera system: as it has been proved in numerous works (Davison, 2003), 2-D information is sufficient in order to estimate the camera motion using a SLAM algorithm, based on static points. The proposed strategy consists in detecting these static points, and moreover detecting and clustering the moving ones in order to track mobile objects: it is the first step towards the full integration of a Visual SLAMMOT approach.

The KLT tracker (Shi and Tomasi, 1994) based on sparse optical flow, is widely used for robotics applications, because of its simplicity and low computational cost. Our own method is based also on the KLT tracker as a valid and confirmed procedure, that can be applied in a real time context during navigation. Next, in order to identify which of the tracked interest points belong to a moving object, we use a clustering based on the a contrario theory (Desolneux et al., 2008). (Veit et al., 2007) have validated this clustering algorithm which does not need any parameter tuning for finding clusters of dynamic features in an image sequence. (Poon et al., 2009) have also adapted this approach for the detection of moving objects in short sequences; additionally, the authors obtain 3D components of feature points to improve the correspondence between the points and the moving objects. The authors present experimental results on real images, acquired from a fixed camera; essential issues of autonomous navigation are not considered.

Finally, the tracking of the detected moving object is performed by Kalman Filtering. This procedure is tested on a long sequence of images acquired in an outdoor open environment during a robot navigation task. Moving object region is incrementally increased thanks to statistical evaluations.

2 OVERALL STRATEGY

Figure 1 presents the algorithm performed for every image acquired at time t with a given period Δt. Ini-

Luz Almanza-Ojeda D., Devy M. and Herbulot A. (2010).
INCREMENTAL DETECTION AND TRACKING OF MOVING OBJECTS BY OPTICAL FLOW AND A CONTRARIO METHOD.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 480-483
DOI: 10.5220/0002849404800483

tially objects tracking (doted rectangle 3) is not activated, because no moving object has been detected. The first step (doted rectangle 1) detects a given number $Npts$ of interest points using the KLT detector. The KLT tracker is then performed on the Nim next images to build a trail for every tracked point. We will call $Nim\Delta t$, the "time of trail" because it represents the number of images used to accumulate positions and velocities of tracked KLT features. Specifically four images are considered as enough to estimate the apparent motion of a point. Before looking for new trails in this process 1, new feature points are selected. KLT process is executed continuously while the robot navigates in order to provide new visual information of the environment at each time of trail.

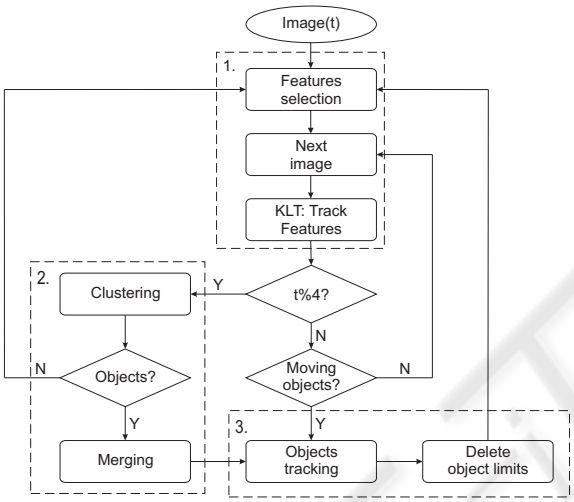


Figure 1: Algorithm to detect and track multiple moving objects.

The second step is performed on trails provided at each time of trail, for only moving features (trails longer than 1 pixel). These features will be grouped thanks to the a contrario theory (doted rectangle 2). Resulting sets of points, i.e. clusters, represent moving objects in the scene. If no object is present then the control process activates again the first step. Otherwise, a merging evaluation is carried out based on similar velocity and close position among already detected objects and new ones. The third step performs independently initialization and tracking by a Kalman filter of clusters detected as moving objects (doted rectangle 3). Finally, object current positions are kept in an occupation grid, in order to avoid several detections of the same object by our procedure. This task is entitled as "Delete object limits" block.

3 OPTICAL FLOW FIELD

We use a sparse optical flow because we must distribute the processing time among some other main tasks. $Npts$ initial interest points in input image t_0 are detected by analyzing of spatial image gradients in two orthogonal directions (typically $N = 150$). Locations of these initial interest points, in next image, are obtained by maximizing a correlation measure over a small window. The iterative process is accelerated by constructing a pyramid with scaled versions of the input image. Furthermore, rotation, scaling and shearing of each point are pertinently handled by calculating their corresponding linear spatial transformation parameters during the iterative process. Once displacement vectors are obtained for all initial features, their velocity is estimated based on their displacement vector.

When moving objects are detected, the corresponding points are subtracted from the $Npts$ initial points. Thus in following iterations, KLT process will search less than $Npts$ new points under the constraint that these points must not be located close to object features. This operation allows us to maintain a fix number of interest points between the KLT and the tracker process; this rigorous control in the number of points is important for our performance because long image sequences will be evaluated.

4 MOVING OBJECT DETECTION AND TRACKING

Given an input vector $V(x,y,v,\theta)$ in R^4 (trails obtained by KLT during a time of trail), the method evaluates which elements in V have a particular distribution contrary to the established random distribution p of the background model. So, a binary tree with V elements is constructed using a single linkage method. Each node in the tree represents a candidate group G that will be evaluated in a set of given regions represented by $\mathcal{H}$. Each region $H \in \mathcal{H}$ is centered at each element $X \in G$ until finding the region H_X that contains all elements in G; at the same time this region has to minimize the probability of the background model distribution. The final measure of meaningfulness (called Number of False Alarms NFA) is given by Eq. (1).

$$NFA(G) = N^2 \left|\mathcal{H}\right| \min_{\substack{X \in G, \\ H \in \mathcal{H}, \\ G \subset H_X}} B(N-1, n-1, p(H_X)) \tag{1}$$

In this equation N represents the number of trails in V, so the number of tracked points from the $Npts$ selected features, $|\mathcal{H}|$ is the cardinality of regions and n is the elements number in the group G. The term which appears in the minimum function is the accumulated binomial law. Distribution p consists of four independent distributions, one for each dimension data. A group G is said to be meaningful if $NFA(G) \leq 1$.

Furthermore two sibling meaningful groups in the binary tree could belong to the same moving object, then a second evaluation for all the meaningful groups is calculated by Eq. (2). To obtain this new measure, we use region group information (dimensions and probability) and a new region that contains both test groups G_1 and G_2 is computed. New terms are $N' = N - 2$, number of elements in G_1 and G_2, respectively $n_2' = n_1 - 1$ and $n_2' = n_2 - 1$, and term $\mathcal{T}$ which represents the accumulated trinomial law.

$$NFA_G(G_1, G_2) = N^4 \cdot |\mathcal{H}|^2 \, \mathcal{T}\left(N', n_1', n_2', p_1, p_2\right) \quad (2)$$

Both measures defined in Eq. (1) and Eq. (2) represent the significance of groups of the binary tree. Final clusters are found by exploring all the binary tree, comparing if it is more significant to have two moving objects G_1 and G_2 or to fusion it in a single group G. Mathematically, $NFA(G) < NFA_G(G_1, G_2)$ where $G_1 \cup G_2 \subset G$.

4.1 Merging Groups

This function is executed when moving objects have been detected from previous times of trail. Let us suppose that new ones are detected by the clustering method. O is a set of M objects given by $O = O_T \cup O_C$ where O_T consists of $(1, 2, ..., k)$ moving objects tracked by Kalman filter, and O_C consists of $(1, 2, ..., l)$ new moving clusters, that could be interpreted either as new moving objects, or part of existing ones. For each object in O, the velocity vector is modeled by the mean of their velocity components in X and Y, respectively represented by μ_{v_X} and μ_{v_Y}. Eq. (3) gives a decision measure for merging regions.

$$\min_{\substack{i,j \in M, \\ i \neq j, \\ O_i, O_j \subset O}} \left(\begin{bmatrix} s(\mu_{v_X}(O_i), \mu_{v_X}(O_j)) \\ s(\mu_{v_Y}(O_i), \mu_{v_Y}(O_j)) \end{bmatrix} \right) < \begin{bmatrix} d_{v_X} \\ d_{v_Y} \end{bmatrix} \quad (3)$$

We evaluate the similarity measure s which performs the subtraction among velocity models for each object in O. Parameters d_{v_X} and d_{v_Y} are constant values set to one pixel. This evaluation is carried out in a linked way, where merged groups are removed from O and added as a new object at the end of the list with, obviously, a new corresponding velocity model. This strategy enriches the decision process for regions merging.

4.2 Moving Objects Tracking

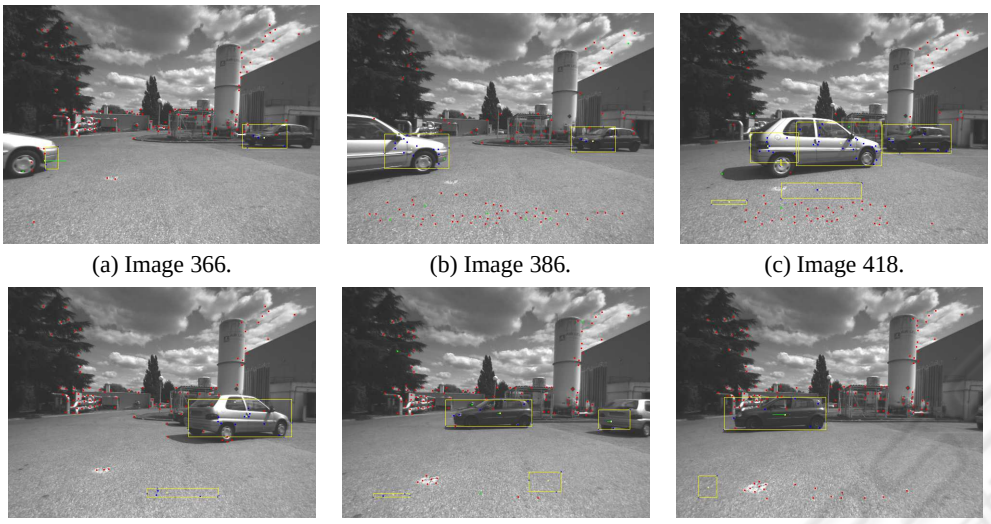
Every new object, defined as a cluster in O_C, is copied in O_T as (1) a list of points and the including bounding box extracted from the last image of the time of trail, and (2) a state vector with the barycenter and the mean velocity, i.e. X, Y, μ_{v_X} and μ_{v_Y} values, respectively. Then, as shown in Figure 1, a Kalman filter tracker, with a constant velocity model, is applied to find the next object position in next images, using the KLT tracker results. A feature point could be removed from the model object when it is not tracked or when the result given by the KLT tracker is not inside the object bounding box or is too far of the mean object points motion. When an object is out of image bounds or occluded in the scene, it is removed from the tracking process.

Finally, a temporal occupation grid is managed in order to select new KLT features, so that the KLT tracker is always applied to $Npts$ points: new points are selected in order to increase the points density inside or around moving objects, or in order to monitor image areas classified as static for a long time.

5 EXPERIMENTAL RESULTS

Robot navigation was performed in a parking with a camera mounted on our robot; 640×480 images are processed off line at $10Hz$ by a C++ implementation of our algorithm. By now, it is not integrated with the robot localization, therefore, we carefully control robot speed. Figure 2 presents images with main situations about object detection during the robot motion.

Figure 2a shows the bounding box of two moving objects, that we labeled as O_1 and O_2 for the right and left side car, respectively. Object region growing could be possible at each time of trail when new clusters are detected, as depicted in Figure 2b for O_1 while Kalman Filter tracks both objects at each image time. Until Figure 2b, O_1 always shows a fronto-parallel motion. Caused by a diagonal motion of the car O_2, our method detects some regions in the same object that have different displacements and consequently different velocities (Figure 2c). To solve this problem, we initialize and track all objects independently and some times of trail later, merging is possible as Figure 2d illustrates it. In the same image, O_1 is hidden



(a) Image 366. (b) Image 386. (c) Image 418.

(d) Image 446. (e) Image 472. (f) Image 488.

Figure 2: Detection clustering and tracking of moving objects during robot navigation. Top row: Detection and tracking of right side car O_1 and left side car O_2. Second row: O_1 and O_2 cross front visual field of robot until O_2 is out of image.

and is removed from the filter process. Detected objects in the ground are caused by camera movement, however they fall quickly out of image bounds. Figures 2e and 2f shows that O_1 is totally detected and tracked again while O_2 region becomes smaller until it disappears.

6 CONCLUSIONS AND FUTURE WORK

The global algorithm works fast, so that it could be embedded on the robot and executed on line. To guarantee the highest performance in overall strategy, the number of feature points processed by the KLT tracker and by the clustering method must be under 150. Two future works are considered; at first, a new strategy is evaluated for reducing the latency time between the arrival of a moving object in the camera view field and its detection by our algorithm. It requires to build several trails and to apply the clustering algorithm in parallel. Moreover, a general strategy to estimate robot motion based on monocamera SLAM approach using static points will be applied to compensate the points motion caused by the camera motion, while dynamic points will be considered in a MOT process.

ACKNOWLEDGEMENTS

This work has been supported by the scholarship 183739 of the Consejo Nacional de Ciencia y Tecnología (CONACYT), the Secretaría de Educación Pública and by the mexican government.

REFERENCES

Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Int. Conf. on Computer Vision*, pages 1403–1410.

Desolneux, A., Moisan, L., and Morel, J.-M. (2008). *From Gestalt Theory to Image Analysis A Probabilistic Approach*, volume 34. Springer Berlin / Heidelberg.

Gate, G., Breheret, A., and Nashashibi, F. (2009). Centralised fusion for fast people detection in dense environments. In *ICRA'09, IEEE Int. Conf. on Robotics Automation, Kobe, Japan*.

Poon, H. S., Mai, F., Hung, Y. S., and Chesi, G. (2009). Robust detection and tracking of multiple moving objects with 3d features by an uncalibrated monocular camera. In *4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, pages 140–149, Berlin, Heidelberg. Springer-Verlag.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1994.*, pages 593–600.

Veit, T., Cao, F., and Bouthemy, P. (2007). Space-time a contrario clustering for detecting coherent motion. In *ICRA'07, IEEE Int. Conf. on Robotics and Automation*, pages 33–39, Roma, Italy.

Vu, T. V. and Aycard, O. (2009). Laser-based detection and tracking moving objects using data-driven markov chain monte carlo. In *ICRA'09,IEEE Int. Conf. on Robotics Automation, Kobe, Japan*.