VISUAL-BASED DETECTION AND TRACKING OF DYNAMIC OBSTACLES FROM A MOBILE ROBOT

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 presents a methodology for detecting and tracking moving objects during mobile robot navigation in unknown environments using only visual information. An initial set of interest points is detected and then tracked by the Kanade-Lucas tracker (KLT). Along few images, point positions and velocities are accumulated and a spatio-temporal analysis, based on the a contrario theory, is performed for the clustering process of these points. All dynamical sets of points found by the clustering are directly initialized and tracked as moving objects using Kalman Filter. At each image, a probability map saves temporally the previous interesting point positions with a certain probability value. New features will be added in the most likely zones based on this probability map. The process detection-clustering-tracking is executed in an iterative way to guarantee the detection of new moving objects or to incrementally enlarge already detected objects until their real limits. Experimental results on real dynamic images acquired during robot outdoor and indoor navigation task are presented. Furthermore, rigid and non rigid moving object tracking results are compared and discussed.

1 INTRODUCTION

A 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 ITS (Intelligent Transportation Systems) communities, from different sensory data such as laser, radar, vision, among others. For driver assistance, many contributions concern laser-based obstacle detection and tracking (Vu and Aycard, 2009). In order to track obstacles, it is required to estimate from the same sensory data, the egomotion of the vehicle: a global probabilistic framework known as SLAMMOT (Simultaneous Localization and Mapping with Mobile Object Tracking) has been proposed by Wang et al. (Wang et al., 2007). But in spite of numerous contributions, this function still remains a challenge when it is based only on vision. So this paper proposes a strategy to detect static points, and moreover to detect and cluster the moving ones in order to track mobile objects: it is the first step towards the full integration of a Visual SLAMMOT approach. It is proposed to reach this objective, using only a monocamera system.

There are various methods for feature detection and tracking using a single mobile camera. A. J. Davison (Davison, 2003) has proposed a spatio-temporal approach, in order to detect and track 2D points from an image sequence and then to reconstruct corresponding 3D points used to locate the camera. The Harris detector allows to initialize points to be searched in successive images from an active strategy, i.e. selecting the best points to be tracked. Each point is represented by its appearance, i.e. a $n \times n$ template around its first position. The system state at instant time t, is given by the 3D position and the speed of every tracked point, as well as the position and the speed of the camera. From this state, every point position is predicted in the image acquired at time $t+1$, forming an elliptic zone where the point must be found if the global state is consistent. Then, each interest point is searched in its predicted zone by a similarity measurement based on correlation score using its template. A clever strategy must be used to update the point template in order to better handle luminance variations and partial object occlusions.

A method widely used for robotics applications is based on the optical flow. Dense optical flow is a hard computing procedure that makes it not so interesting for real time applications. However, if optical flow is only extracted for interest features, i.e. for a very small part of total image points, a tracking pro-

Luz Almanza-Ojeda D., Devy M. and Herbulot A. (2010).
VISUAL-BASED DETECTION AND TRACKING OF DYNAMIC OBSTACLES FROM A MOBILE ROBOT.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 98-105
DOI: 10.5220/0002954800980105
Copyright © SciTePress

cedure of many objects characterized by some points, could be applied in real time. Such a sparse solution has been proposed by Shi-Tomasi (Shi and Tomasi, 1994) and it is commonly used in computer vision because of its simplicity and its low computational cost. Our own method is based also on this method as a valid and confirmed procedure, that can be applied in a real time context during navigation. Among others, let us cite the work presented in (Lookingbill et al., 2007) where authors have used the optical flow field to leverage the difference in appearance between objects at close range and the same objects at more distant locations. This information allows them to interpret monocular video streams during off-road autonomous navigation and to propose an adaptive road following in unstructured environments. This method has been evaluated for the navigation of an intelligent vehicle in a desertic terrain.

However, once interest points and optical flow are extracted and tracked from an image sequence, it is so important to distinguish which of those tracked points represent moving objects. Clustering techniques are the first basic solution to this question, but unfortunately most of them require initial information about the scene as the number of clusters to find. The success of these methods highly depends on these parameters. T. Veit et. al (Veit et al., 2007) coped with the same issue for the analysis of short video sequences. They validate a clustering algorithm based on the *a contrario* method (Desolneux et al., 2008) which does not need parameter tuning or initial scene information for finding clusters of mobile features. This approach has been also used in (Poon et al., 2009) to detect moving objects in short sequences; additionally, authors obtain 3D components of feature points to better detect the correspondence between points and moving objects on which they have been extracted. This work presents experimental results on real images, acquired from fixed cameras, so that essential issues of autonomous navigation are not considered.

This paper proposes moving object detection and tracking on a robot navigation context based on KLT tracker and the a contrario clustering. Then, the resulted clusters are initialized as moving objects and tracked by a Kalman Filter. At each iteration, image locations are ponderated by a bi-variate pdf function when a point is tracked. This value represents the probability that the location contains a dynamical point, being 0 the most interesting location where to find an interesting point.

Next section explains the proposed strategy. KLT procedure used to detect and track interest points is briefly described in section 3. Section 4 describes the main concepts of the *a contrario* theory. The global detection-clustering-tracking approach and the use of probability map in a long sequence of images are presented in section 5. Experimental results on real images are presented and discussed in section 6. Finally last section concludes and explains future works.

2 OVERALL DETECT, CLUSTER AND TRACK STRATEGY

In this work only visual information in grey-scale acquired from a mobile robot is used for scene analysis. General block diagram in Figure 1 describes the procedure carried out to detect and track multiple moving objects.

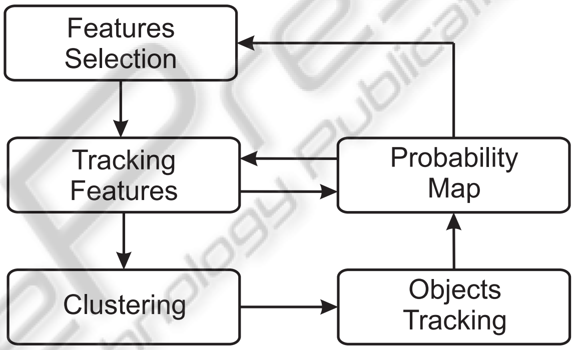


Figure 1: Algorithm to detect and track multiple objects.

N initial feature points are selected in the input image(t) using the Shi-Tomasi approach. Feature locations in next image are searched by the KLT tracker, based on correlation and optimization processes. Let us suppose first that no object is tracked; so the process loops on N_{im} successive images executing only two tasks, feature tracking and update of the probability map. We will call *time of trail*, this set of N_{im} processed images, i.e. the number of images used to accumulate positions and apparent velocities of tracked features. In order to select new features, we seek into the probability map the most interesting zones in the image for searching new points, i.e. points which have the higher probability to be on mobile objects. KLT process is executed continuously in this way, while the robot navigates in order to provide new visual information at each time of trail. Details of the KLT theory are explained in next section.

At the end of each time of trail, only moving KLT features are selected for being grouped by the a contrario clustering method. Here, moving KLT features are characterized by a velocity vector higher than a threshold set to 1 pixel. Resulted clusters represent moving objects in the scene; every one is directly initialized as a moving object and then tracked with a position and an apparent velocity estimated by a Kalman Filter. The Object tracking task exploits the KLT tracker: it verifies the consistency between the point tracking in the current image and the Object characteristics. At every iteration, this object could be merged with another detected object based on similar velocity and close position. Implementation details of both clustering and merging process are presented in section 4. Finally, object current positions are stored in the probability map with the highest probability of occupation, so this zone of the image is not interesting to look for new features.

3 FEATURE SELECTION AND TRACKING

Optical flow procedure used in this work is based on the initial technique proposed in (Lucas and Kanade, 1981) and on the well-known *Select Good Features to Track* algorithm (Shi and Tomasi, 1994). This technique is largely used in the robotics community because it proposes to match the more salient points, minimizing the processing time.

Detection of Moving Points. N distinctive feature points are initially extracted from image t_0 by the analysis of spatial image gradient in two orthogonal directions. Locations of these N points in next image are obtained by maximizing a correlation measurement over a small window. Iterative process is accelerated by constructing a pyramid with scaled versions of the input image. Furthermore, rotation, scaling and shearing are applied on each correlation window by optimizing a linear spatial transformation parameters during iterative process. Once displacement vectors are obtained for all initial features, apparent velocities are estimated based on displacement vectors.

This KLT optical flow method is sparse because only few points are initially selected to describe image content. This sparse method is used in order to save processing time, because some other essential tasks must be performed. Figure 2 (frames 54 to 58 extracted from the sequence shown in second row of Figure 5) depicts feature points detected and tracked during a time of trail from $N = 150$ initial selected points. From these accumulated locations, blue points represent points with long optical flow displacements. For these points the assumption that they belong to a moving object has a high probability.

Perception on Moving Objects. When moving objects are detected, the number of dynamic points is subtracted from the N points tracked permanently by KLT. Thus in following iterations, KLT will select less than N new points. This strategy allows to track only a fix number of N points between KLT and the Object tracking process. This rigorous control on the number of points is important in our methodology because long image sequences will be evaluated and the performance directly depends on the number of processed points.

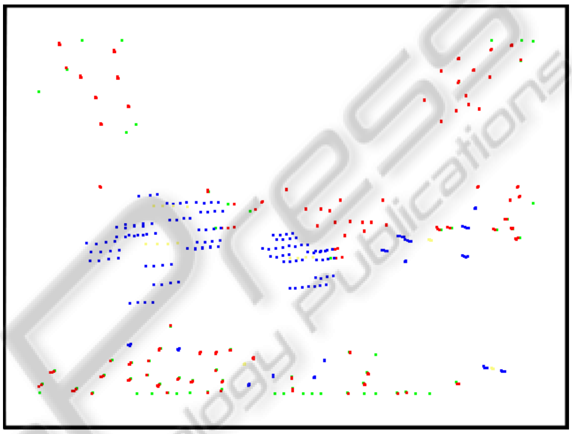


Figure 2: Accumulated Optical flow in 5 successive images from the sequence shown in the second row of Figure 5.

Figure 3 shows N feature points with locations tracked by KLT process at three different times of trail (8, 18 and 30). After 18 images and without any prior knowledge of the environment, we perceive consistent points motions, caused by the camera movement. By analyzing optical flow extracted from images acquired from a moving camera in dynamic environments, we deduce that larger is the time of trail, better will be the perception of moving objects, but also of the camera motion. In order to minimize the egomotion impact, we propose to reduce this time to 5 images which covers up egomotion and also, points out independent movements. This can be verified in Figure 2; a constant movement in the cloud of points can be seen in the middle of the image; egomotion is less remarkable. Furthermore, the best advantage of this choice $N_{im} = 5$ is the reduction of the waiting time before performing the a contrario clustering task.

4 A CONTRARIO METHOD AND MERGING PROCESS

Visual perception is a complex function that requires the presence of ”salient” or ”evident” patterns to iden-

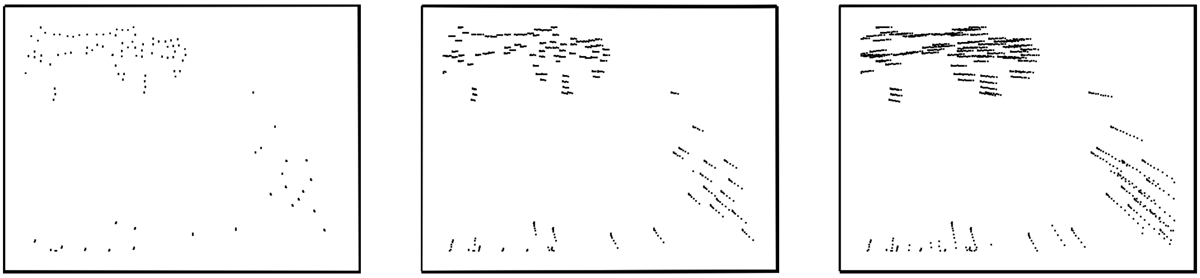


Figure 3: From $N = 150$ selected points in the first image, tracking results on a sequence acquired when the robot turns left: accumulated Optical flow along 8 (left image), 18 (middle) and 30 (right) images.

tify something that ”breaks” the continuous motion due to the camera egomotion. This ”salient pattern” corresponds to ”meaningful event” detected by the a contrario method (Cao et al., 2007). Basic concepts of the a contrario clustering inspired by the Gestalt theory, are exposed in (Desolneux et al., 2003) and deeply in (Desolneux et al., 2008). In general, Gestalt theory establishes that groups could be formed based on one or several common characteristics of their elements. In accord to this statement, an a contrario clustering technique (proposed by Veit, et. al. (Veit et al., 2007)) identifies one group as meaningful if all their elements show a different distribution than an established background random model. Contrary to most of clustering techniques, neither initial number of clusters is required nor parameter has to be tuned. In our context, i.e. unknown environment, approximated egomotion..., these characteristics are very favorable.

4.1 Evaluation of a Background Model

We use the background model proposed in (Veit et al., 2007) which establishes a random organization of the observations. Hence, background model elements are independent identically distributed (iid) and follow a distribution p. The iid nature of random model components propose an organization with not coherent motion present.

Next, given an input vector $V(x, y, v, \theta)$ in R^4, first objective is to evaluate which elements in V show a particular distribution contrary to the established distribution p of the background model (that explains ”a contrario” name). To avoid element by element evaluation, first, 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 evaluate in a set of given regions designed by $\mathcal{H}$. This set of regions is formed by different size of hyper-rectangles that will be used to test the distribution of several data groups. Each region H is centered at each element X in the group until finding the region H_X that contains all the group and at the same time makes minimal the probability of the background model distribution. Different size of hyper-rectangles are used in function of data range, in our experiments we use 20 different sizes by dimension. Final measure of meaningfulness (called Number of False Alarms NFA in referenced work) is given by eq 1.

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

In this equation N represents the number of elements in vector V, $\left|\mathcal{H}\right|$ is the carnality of regions and n is the elements in group test G. The term which appears in the minimum function is the accumulated binomial law, this represents the probability that at least n points including X are inside the region test centered in X (H_X). Distribution p consists of four independent distributions, one for each dimension data. Point positions and velocity orientation follow a uniform distribution because object moving position and direction are arbitrary. In other hand, velocity magnitude distribution is obtained directly of the empirically histogram of the observed data. So that, joint distribution p will be the product of this four distributions. 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 reuse region group information (dimensions and probability) and just a new region that contains both test groups G_1 and G_2 is calculated. 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.

$$NFA_G(G_1, G_2) = N^4 \cdot \left|\mathcal{H}\right|^2 \mathcal{T}\left(N', n'_1, n'_2, p_1, p_2\right) \quad (2)$$

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

4.2 Merging Groups

This function is executed only if moving objects have already been detected. O is a set of M objects that will contain all candidate objects for merging evaluation. That is, $O = O_T \cup O_C$ where O_T consists of $(1, 2, ..., k)$ moving objects tracked by Kalman filter, and O_C consist of $(1, 2, ..., l)$ new moving clusters, 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_{vX} and μ_{vY} . We use these models to evaluate eq.3 that let establish a decision constraint for merging.

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

We evaluate the similarity measure s which performs the subtraction among velocity models for each object in O. Parameters d_{vX} and d_{vY} are constant values set to one pixel in accord with the previous established threshold for detecting moving features in KLT process. This value is chosen in order to conserve the best trade off between the threshold of moving points in KLT module and the expected bias among object velocities in the scene. 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 allows the merging of the same objects previously detected with a more enriched model. Finally, even when both O_T and O_C object velocities are not resulted from the same process, they could be compared because both are based on pixel displacement in the scene (their optical flow).

5 MOVING OBJECTS TRACKED BY A KALMAN FILTER

Groups found by clustering technique are composed of point locations at different processing time. Hence, to confirm that moving object is still in the scene, only points present in last processed image are taken to initialize each cluster as a moving object in O_C. Therefore only the points located on the current image will be used to model the moving object.

To track moving objects in O_T along the image sequence, Kalman filter prediction evaluates a constant velocity model. To initialize this model a vector state is defined for each moving object detected. Vector state consists of the barycenter of object in X and Y and its velocities are set to μ_{vX} and μ_{vY} values, respectively. We assign the estimated position and velocity calculated by Kalman filter to baricenter of the object. This estimated position is used as the center of a window that will be extracted in the next image in order to search the object points. We called this window the zone of object and its size is a function of previous object limits and a security margin. Once object region is extracted in next image, we carried out a correlation process to find new object location.

5.1 Model Object Detection

The concept of object developed here covers the management of several points which have been evaluated as a part of a moving object. In this work, the task of tracking consists in following each object element with its particular appearance and the only relationship among them is their velocity and position. Thus, model object initialization consists in extracting a window patch (a template) around each point in the image where object was detected. The same number of templates are extracted around the estimated feature location in the next image. Appearance of initial templates in the current image is updated by an affine model. Feature points could be removed from the model if one or more of the following cases happen:

- Feature location is not found by correlation
- Location found is not inside the bounding box
- Displacement and velocity of the points found are not inside of the normally distribution of their respective mean data.

5.2 Occupancy Map

In our algorithm, it is important to add new points that complete the model of the detected moving object and at the same time detect new incoming objects. In a context of unknown environment, feature points should be initially chosen in all the image without highlighting some locations. To overcome this uniform point selection along robot navigation process, a space-time occupancy map is constructed by cells centered on tracked points indicating favorable locations for new feature detection. This map develops the idea of an occupation grid in the sense of higher probabilities represent the locations in the image that are no important to seek for a new point (like occupied locations). So, mainly locations in the image with lower probability values in the map will be first used to look for new points.

First probability map locations are initialized to $p_0 = 0.5$ value that represents the initial fair selection of feature points, that is, $p(u,v,t=0) = p_0$. When an interesting point is detected in (u,v) at $t > 0$, it becomes the center of an occupied cell and its corresponding value in the map is given by eq. 4 with $\alpha_t = 0$. A cell becomes empty in function of interest point displacements at every iteration : if a tracked point leaves the cell it was using at time $t-1$ its probability is updated and then this cell is labeled as empty in the current image t. The new cell in which tracked point is located at time t becomes now an occupied cell. The Figure 4 depicts simulated results of this map construction along five successive images for a tracked point. A square cell of size 11x11 centered on (u,v) point location is used. Cell value is always given by a 2D Gaussian pdf in combination with the rules established by eq. 4 and 5. The inverse effect of function $\alpha(u,v)$ applied to previous point positions is clearly seen in this image. With this probability map we enhance the assumption of new points of incoming object will appear behind its current detected point positions.

$$p(u,v,t) = \sum_{u,v \in Cell} (\alpha(u,v)_t + \mathcal{G}(\mu_u, \mu_v, \sigma_u, \sigma_v)) \quad (4)$$

where $\alpha(u,v)$ function describes the previous probability in the cell according to the previous state of location (u,v), that is:

$$\alpha(u,v)_t = \begin{cases} p_m - p(u,v)_{t-1} & \text{if } Cell_{u,v,t-1} \text{ occupied} \\ p_0 & \text{if } Cell_{u,v,t-1} \text{ empty} \\ 0 & \text{if } (u,v,)_{t-1} \text{ fair} \end{cases} \quad (5)$$

where p_m is the maximal probability value, set to 1 in our procedure. In order to avoid that this map stores for long time the same probability values in some locations, the map is reset each 2 times of trail process, except for the current object tracking locations.

6 EXPERIMENTAL RESULTS AND DISCUSSIONS

Proposed algorithms have been implemented in C, C++ and TCL and included as module into a framework for developping algorithms in robotics. Robot navigation was performed in indoor and outdoor context with a camera mounted on the robot (640×480 at $10Hz$). The number N of tracked points by KLT is set to 150. At this first stage of the experiments, any information about the odometry of the robot is available, therefore, we carefully control robot speed navigation.

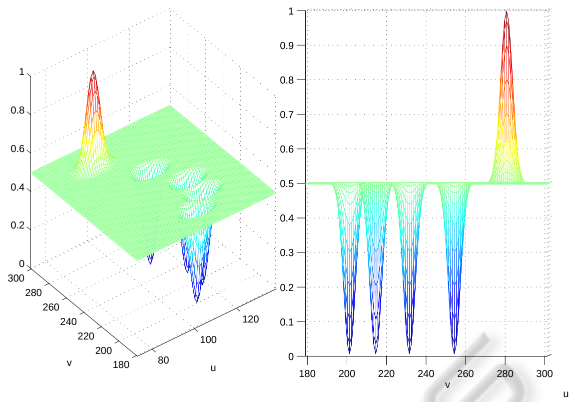
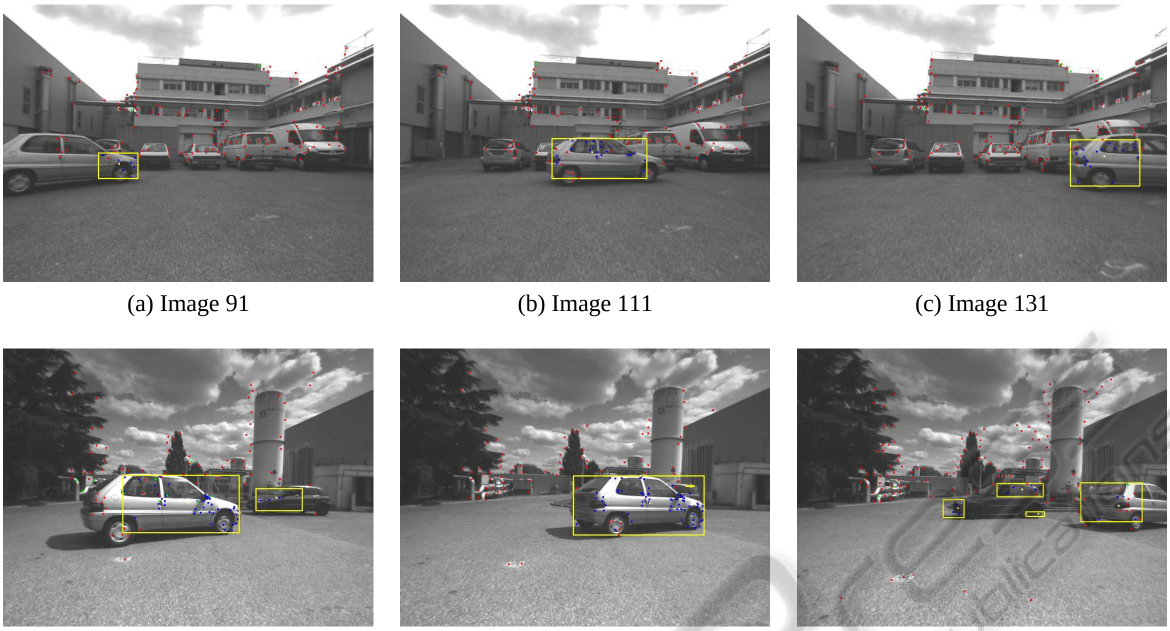


Figure 4: Simulated results of probability map for a tracked point in 5 successive images. Inverse shapes represent the probability in the four previous images, higher shape depicts current point location.

First test was carried out on a sequence of 300 images. The first row in Figure 5 shows three images selected from this sequence, at the time of a moving object appears in the field of vision of the robot. Image 5a shows the bounding box of moving object that enters from the left side of the image, all detected points inside of rectangle are used to initialize object model. After 20 processed images bounding box of initial object is enlarged (see image 5b). Even when in image 5c half of the moving object is out of the scene, the object is still tracked by the Kalman Filter. This is a specially difficult sequence because some images show drastic changes due to vibrations. So most of the points enters to the clustering process, but their movement is classified as random. The second row on Figure 5 shows the algorithm performance with object occlusions. In this case the car which is closer to the robot is well detected and tracked. Some problems to perfectly detect the second car after the occlusion occur and image 5d shows that the algorithm divides it in 3 different moving objects.

6.1 Non Rigid Moving Objects Tracking

An extension of tests to indoor environments is carried out for evaluating the development of our algorithm to track non rigid moving objects. Indoor environment induces more stable illumination conditions and normally a less charged background where non rigid objects could be better handle. The Figure 6



(a) Image 91 (b) Image 111 (c) Image 131

(d) Image 58 (e) Image 66 (f) Image 86

Figure 5: Experimental results in outdoor environment. First row shows moving object detection and tracking: (a) initial detection, then total detection in image (b) by merging function, finally tracking just the latest part of the car in (c). Second row : moving object occlusion is successfully handle by our proposed method.

(a) Image 193 (b) Image 206 (c) Image 229

Figure 6: Indoor experimental results show the non-rigid objects detection and tracking. A good but incomplete detection is done for a person who crosses the room image (a). Then a more enhanced detection is done in images (b) and (c) however random natural turns during person walking perturb the successive detection of object points.

shows a person that crosses the field of view of the mobile robot during its navigation in an indoor context. As long as the person moves away of the robot, almost a complete detection is achieved. However, with a considerable delay, initially only the head of the person is detected. The leg movements are large and in a forward-backward direction and it disturbs the detection of a non random movement. To overcome this restriction in our algorithm, a strategy that involves the use of 3D camera and several point positions given by a SLAM process is being developed. So, an additional but fast analysis comparison between the depths of moving points and detected moving object will provide enough information to associate these points to moving object. The 2D dimensional context of our proposed method is preserved by transforming 3D points into 2D points before entering to clustering process and at the end taking into account their third coordinate position only for achieving a faster and more complete non rigid detection.

7 CONCLUSIONS

Experimental results show that even with few images, it is possible to detect a rigid (and partially non-rigid) moving object by a spatio-temporal analysis of features. Object model is enlarged thanks to prior knowledge managed by the proposed probability map. This map is successfully used during the active search of feature points because it mainly highlights zones that certainly contain new moving interesting points. Our tests are performed off line on a recorded sequence; however, the global algorithm works fast and could process images at 10Hz. The clustering method is the highest time consuming in the global process; for that reason, the number of trails to be grouped by the clustering method, should be no more to 150 points. Thus, the trade-off between image size and that number of points guarantees the highest performance in overall strategy.

It has been assumed that all pixels whose displacements are less than one pixel could be considered as noise or as points displaced by little vibrations of the camera. However the most sensible part of our algorithm resides in robot motion. Under not controlled conditions of velocity, most significant displacements are concentrated in both left and right image sides, mainly caused by egomotion. A general strategy to avoid egomotion detection and non rigid moving objects is being integrated based on monocamera SLAM approach. An interchange of 3D and 2D points information between SLAM and our MOT process will be continuously carried out giving a cooperative sense to our new proposed strategy. That is, detected static points will be sent to SLAM, these points are candidates to be included as a new landmark in the stochastic map used to update camera pose estimation. Then, this camera pose will be received by our MOT process to estimate the camera motion and calculate real detected point displacements.

ACKNOWLEDGEMENTS

This work has been performed in the context of the RINAVEC project funded by ANR, the french *Association Nationale de la Recherche*. It 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

Cao, F., Delon, J., Desolneux, A., Musé, P., and Sur, F. (2007). A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2):91–119.

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. (2003). A grouping principle and four applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):508–513.

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

Lookingbill, A., Lieb, D., and Thrun, S. (2007). *Autonomous Navigation in Dynamic Environments*, volume 35 of *Springer Tracts in Advanced Robotics*, pages 29–44. Springer Berlin / Heidelberg.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision (darpa). In *Proc. 1981 DARPA Image Understanding Workshop*, pages 121–130.

Poon, H. S., Mai, F., Hung, Y. S., and Chesi, G. (2009). Robust detection and tracking of multiple moving objects with 3d featu res by an uncalibrated monocular camera. In *Proc. 4th Int. Conf. on Com puter 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 *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, 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 *IEEE Int. Conf. on Robotics Automation (ICRA), Kobe, Japan*.

Wang, C., Thorpe, C., Thrun, S., Hebert, M., and Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *Int. Journal of Robotics Research*.