Moving Objects Tracking in Video by Graph Cuts and Parameter Motion Model

Khalid Housni, Driss Mammass
IRF – SIC laboratory, Faculty of sciences
Agadir -Morocco

Youssef Chahir
GREYC-UMR CNRS 6072, University of Caen
Caen – France

ABSTRACT
The tracking of moving objects in a video sequence is an important task in different domains such as video compression, video surveillance and object recognition. In this paper, we propose an approach for integrated tracking and segmentation of moving objects from image sequences where the camera is in movement. This approach is based on the calculation of minimal cost of a cut in a graph "Graph Cuts" and the 2D parametric motion models estimated between successive images. The algorithm takes advantage of smooth optical flow which is modeled by affine motion and graph cuts in order to reach maximum precision and overcome inherent problems of conventional optical flow algorithms. Our method is simple to implement and effective. Experimental results show the good performance and robustness of the proposed approach.

General Terms
Tracking, Video Analysis, Computer Graphic.

Keywords
Graph Cuts, Motion Models, Tracking, Video Analysis, Overlapping Objects.

1. INTRODUCTION
Tracking of moving objects is one of the major areas of research, several approaches have been developed. We can classify these approaches according to the principle used in two great classes: *Model-based* approach, these approaches use the result obtained in the previous frame as a prediction process initialization in the current frame [1, 2, 3, 4, 5, 6]. *Motion-based* approach (also called *intensity based*) [7, 8, 9, 10], the prediction of the tracked object is done by the motion model estimated between the previous and current frame.

Whatever the method used, model-based or motion-based, we can classify these approaches according to the object shape representations commonly employed for tracking into three categories (the reader is referred to [11] for details):

- Point: The object is represented by a point, that is, the centroid [12] or by a set of points [13, 1].
- Primitive geometric shapes: Object shape is represented by a rectangle, ellipse, etc [14].
- Object silhouette and contour: the object is represented by a contour [5] or by silhouette (region inside the contour) [6, 4].

The methods of the last category are often based on dynamic segmentation technique. These methods are used for tracking complex non-rigid shapes and when we want to extract the silhouette of the object at any time, without *prior* knowledge of its new shape.

To summarize, a robust approach to tracking moving objects must solve three major classes of problems.

- Problem of large displacements: the methods based on the modification of contour (like [5]) or shape (like [4]) predicted in the previous step cannot support the great movement of objects. The simplest way to solve this problem is to use the motion model of the object to track in the step of predicting.
- Problem of topology change: Several approaches in the literature namely [1, 2, 3] cannot handle topology change due to contour or silhouette used in the initialization or due to dominant motion model used that doesn't always cover all pixels of the object to track. This is the case for complex non-rigid shapes (Namely the movement of the feet).
- Problem of overlapping objects: the classification of pixels of overlapping objects is the most difficult problem and especially when we want to extract the shape of objects at any time.

In this paper, to manage the objects displacement, we used a prediction based on motion model computed between the current and previous frame. That will take into account the large displacement. And we have used a dynamic segmentation step based on optimal boundary diction by graph cuts to solve the problem of topological change and the movement with a complex model (such as when parts of an object have different models). This stage of refinement by graph cuts cannot classify the pixels of objects that are overlapped. We overcome this limitation through the use of a similarity measure based on the observed intensities.

This paper is organized as follows: In the next section we present the basic definition and terminology of graph cuts and how they are used for image segmentation. In sections 3 and 4, we introduce the algorithm of object extraction under hard constraints and we explain how to use a graph cuts algorithm in order to refine a border of predicted objects, to obtain rich objects description and to overcome inherent problems of conventional optical flow algorithms, and we give a solution for case of overlapping objects. In section 5, we give the results obtained. Lastly, we give a conclusion and future works.

2. GRAPH CUTS METHODS IN VISION
Graph cuts have been used to efficiently solve a wide variety of vision problems like stereo [15] image segmentation [16, 17], image restoration [15, 18], texture synthesis [19] and many others that can be formulated in terms of energy minimization. This energy is represented in this form:

$$E(L) = E_{data} + E_{smooth}$$

$$= \sum_{p \in P} D_p(L_p) + \sum_{\substack{p=1..n \\ q \in N_p}} V_{pq}\ (L_p, L_q) \quad (1)$$

Where E_{smooth} is a piecewise smoothness term, E_{data} is a data term, P is the set of pixels in the image, $V_{pq}\ (L_p, L_q)$ is a smoothness penalty function to impose a spatial consistency, $D(l_P)$ is a data penalty function and $(L_p)_{p=1...\ n}$ is a vector of variables in finite space L $L_p \in \{s,t\}$ (L for label). Every pixel is associated with a variable L_p. N_p designates the set of the neighbors of the pixel p.

The main idea of a graph cuts is to bring back the problem of minimization of energy to a problem of a minimal cut in a graph. Greig and al. have shown that the minimization (such as estimating the maximum a posterior Markov random field) can be achieved by the minimum cut of a graph with two specific nodes "source" and "sink" for the restoration of binary images [18]. Later, the approach has been extended to non-binary problems known as "S-T Graph Cuts" [15].V.Kolmogorov [20] have given a necessary and sufficient condition, known as sub-modularity condition for a function to be minimized by graph cuts. In the case of energy (1), $V_{pq}(L_p, L_q)$ is a term of vicinity, which makes it possible to check the condition, and the construction of graph is done as follows:

As shown in figure 1, each pixel P_{ij} in the image corresponds to a node (vertices) v_{ij} in the graph. Two additional nodes are the *source* and *sink*, respectively the object and the background. Each node is linked to its neighbors by edges *n-link*, with connectivity chosen (in our implementation we used 8-connected lattice as our neighborhood structure), and whose capacities depend on differences in intensity, this allows to encourage the segmentation (cut) which passes through regions where the image gradient is strong enough. Each node is also connected by edges, *t-link* to the terminals (source and sinks).

A cut *C* of graph *G (V, E)* is a partition of the vertices V into two sets V_1 and V_2, such that :

$$V_1 \cup V_2 = V$$
$$V_1 \cap V_2 = \phi$$
$$source \in V_1 \quad and \quad sink \in V_2$$

In graph theory, the cost of a cut is usually defined as the sum of the weights corresponding to the edges it serves as is formulated in (2).

$$|C| = \sum_{\substack{p \in S \\ q \in T \\ (p,q) \in C}} w(p,q) \quad (2)$$

The value of *E (x)* is equal to same constant ε plus the cost of the "ST minimum cut" *C*. We say that *E* is exactly represented by *G* if the constant $\varepsilon = 0$.

To calculate the minimum cut, several algorithms have been developed: increasing path [21], Pushing-relabel [22] and the new algorithm of Boykov - Kolmogrov [23]. In our application, we used the algorithm of Boykov-Kolmogrov for its speed.

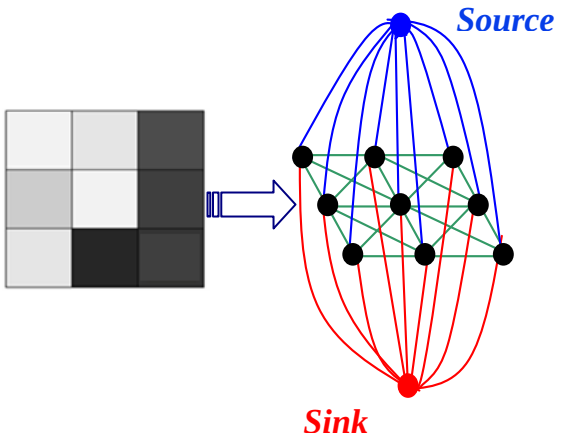


Fig 1: Construction of graph for a 3x3 image with the connectivity N8. Each pixel corresponds to a node, and all pixels are connected to the source and the sink.

3. OUR APPROACH

The simple way for tracking an object in movement is to detect motion by the subtraction of each two successive frames, but this necessitates that the background is stationary (fixed camera). In this paper, we present an approach of tracking moving objects in a dynamic frame (camera in movement). In the first step, we use the graph cuts to select the objects to be tracked (section 4 -1). The second step consists of calculating the motion model for each object (section 4 -2), which corresponds to the selected objects (or predicted objects), between the current frame and the following one. These models of movement will be used to predict each object in the next frame. Thereafter, we will use the graph cuts to obtain precise contours (section 4 -3). Figure 2 gives the principal steps of our algorithm. To classify the pixels of overlapping objects we used the multi may cuts algorithm (section 4 -4).

4. ALGORITHM IMPLEMENTATIONS

In this section, we discuss the implementation details of the proposed algorithm depicted in figure 2.The algorithm consists of three major steps : Selection of objects to track by Graph cuts (step of initialization), Prediction step by visual motion model, and step of borders objects refinement.

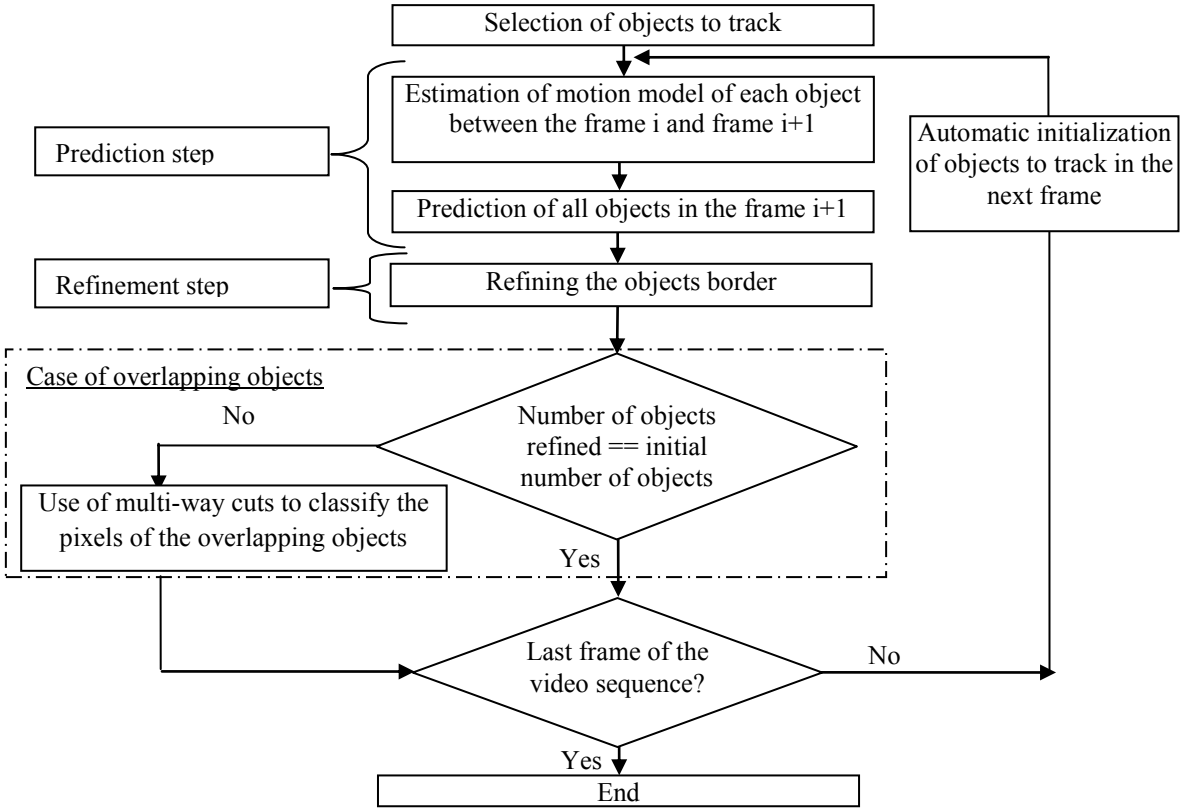


Fig 2: General outline of our approach.

4.1 Selection of Objects to Track

To select the objects to track, we used the interactive segmentation method based on detection of the boundary by Graph Cuts [16]. The extraction is done by the exploitation of the *prior* information entered by the user (seed of the object and background) (see figure 4 -a). These pixels selected by the user are then connected to the terminals (source and sink) by the edges with infinite weight (4) (5). This will satisfy the constraints (3), while all other pixels (nodes) will have no links to the terminals (6)

It is about using some topological constraints, known as "hard constraints" which can indicate that certain seeds of image *a prior* identified as a part of the Object or the Background

All neighboring pixels are connected by edges, n-links, whose capacities depend on intensity differences |Ip −Iq | (7). (see figure 3).

$$\begin{cases} \forall p \in O : L_p = 1 \\ \forall p \in B : L_p = 0 \end{cases} \quad (3)$$

The cost of the edges is given by:

$$\begin{cases} W(p,S) = \infty \\ W(p,T) = 0 \end{cases} \quad if \quad p \in O \quad (4)$$

$$\begin{cases} W(p,S) = 0 \\ W(p,T) = \infty \end{cases} \quad if \quad p \in B \quad (5)$$

$$W(p,S) = W(p,T) = 0 \quad if \quad p \notin \{O \cup B\} \quad (6)$$

$$W(p,q) = \exp\left(-\frac{\|I_p - I_q\|^2}{2*\sigma^2}\right)\frac{1}{dist(p,q)} \quad (7)$$

σ: Standard deviation of the intensities of the seeds of object
S: source, T: sink, O: seeds of object, B: seeds of background.

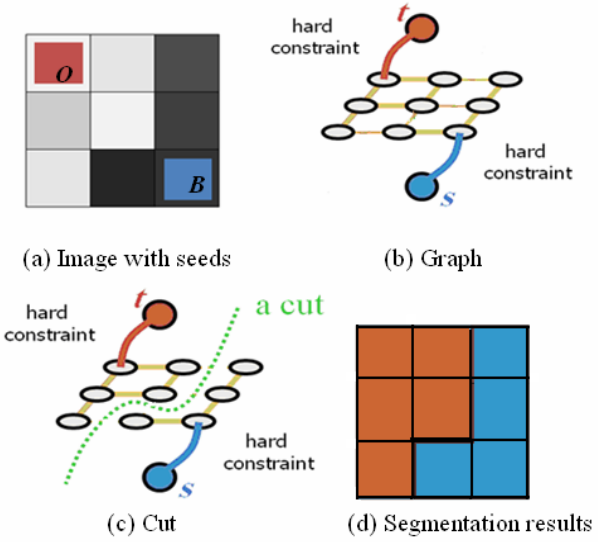
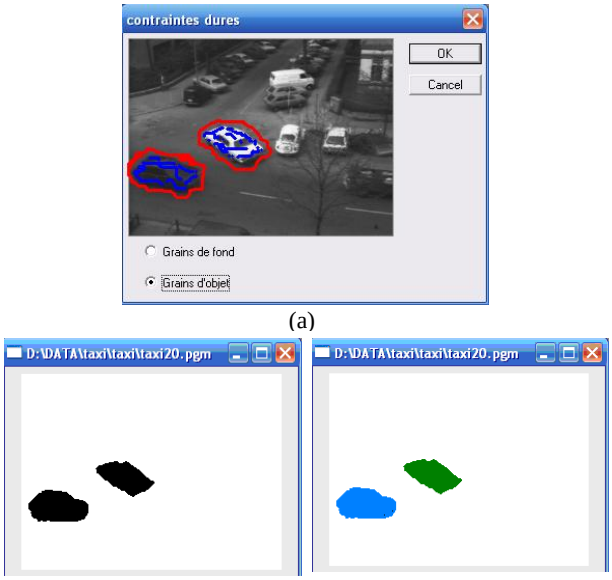


Fig 3: A simple 2D optimal boundary for 3x3 image. Boundary conditions are given by object seeds O and background seeds B provided by the user.



(a)

(b) (c)

Fig 4: (a) First frame with hard constraints, (b) Objects selected by graph cuts- to be tracked, (c) identification of selected objects by algorithm of detection of connected component.

4.2 Visual Motion Model

The whole key of the moving objects tracking, using motion based approach, resides in the choice of the motion model for each object because this one must represent the movement of all pixels in the object.

Our aim is to control the displacement of the objects to track with *a prior* knowledge of the objects and the background in the previous frame. As we have already introduced, a motion model is used to approximate the objects in the courant frame.

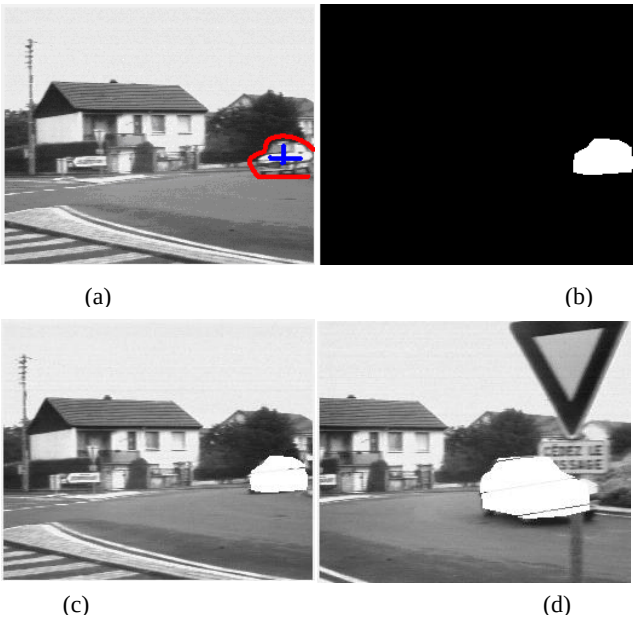
In the literature, several models of motion have been presented like rigid, homography, quadratic, etc. Affine Model (8) is a reasonable compromise between the low complexity and fairly good approximation of the non-complex motions of objects at about the same depth.

$$\vec{\omega}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} * \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (8)$$

In other words, affine motion model is a parametric modeling to six parameters (c_1 ,c_2, a_1, a_2, a_3, a_4) of the movement defining the transformation T: $\Omega_t \rightarrow \Omega_{t+1}$ which permits to pass the pixel x to the pixel x'.

Several methods have been proposed to estimate a parameter motion model. The most famous method in this field could be using multi-resolution approach [24]. This method allows calculating a parameter motion model in real times between the two input frames. In our application, we used this algorithm for its speed.

An implementation of this algorithm between two successive images can be found in Motion2D software, developed at Irisa/INRIA Rennes [25].



(a) (b)

(c) (d)

Fig 5: (a) First frame with hard constraints, (b) Object selected by graph cuts -to be tracked, (c), (d) The object predicted in the frames 2, 23 of the rond-point sequence using motion model.

4.3 Refinement of Border Objects

We can clearly see that in the figure 5, that a part of the road "background" at summer covered by the predicted object and a part of the car "object" is not covered. Figure 6 & 8 shows this.

Fig 6: This figure illustrates that a part of the rode at summer covered by the predicted object and a part of the car is not covered.

To obtain a rich object description with the aim of accomplishing advanced scene interpretation tasks we used the graph cuts approach described in section 3.1. therefore, as we have already introduced, graph cuts method for segmentation requires the user to mark a few pixels as object (seeds of object) and few others as background (seeds of background) and to generate automatically this information, we used the mathematical morphology (see figure7):

- Seeds of object: Four-time erosion of object detected using motion model (figure 7-b).
- Seeds of background: Seven-time dilatation of the object detected minus five-time dilatation of the object detected (figure 7-c).

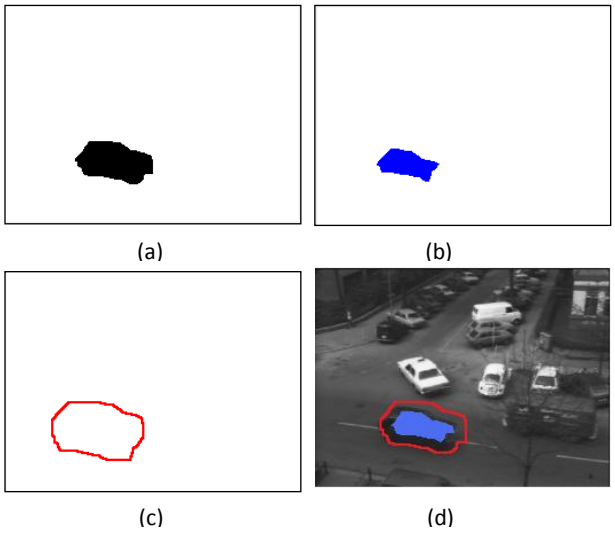


Fig 7: (a) object predicted using motion model (b), (c) hard constraint (seeds of object and background). The area between the blue (seeds of object) and the red line (seeds of background) is a band of incertitude around the borders. A graph is then built on this band to optimize the contours of the object.

This optimizes the borders object with a precision of ± 9 pixels, thus the number of pixels an object approaching the camera will increase and the reverse for the object away from camera.

Fig 8: First image gives the result obtained using only motion model, second image give the result obtained after the refinement by graph cuts.

4.4 Case of Overlapping Objects

Before calculating the motion model for each object, we use the algorithm of detection of a connected component to identify the selected objects (labeling).

This algorithm will classify the image pixels according to their belonging to *object_1*, *object_2… object_n* or to *background*. However, the image pixels classification is reliable when the objects to track are separated and consequently, multi-objects tracking may be ambiguous in some cases, like in the case of overlapping objects.

We overcome this limitation through the introduction of a visual appearance constraint, where a classification according to the observed intensities is to be considered. To this end, we will use the multi-way cuts algorithm [15] to classify the pixels of overlapping objects.

We define the cost functions for multi-way cuts as follows:

$$if \begin{cases} I_p \notin object_{i(i=1,2...n)} \ and \\ I'_{\omega_{Ai}(p)} \in object_{i(i=1,2...n)} \end{cases} \begin{cases} W(\omega_{Ai}(p), L_{i(i=1,2..n)}) = 0 \\ W(\omega_{Ai}(p), B) = 0 \end{cases} \quad (9)$$

$$if \begin{cases} I_p \in object_{i(i=1,2...n)} \ and \\ I'_{\omega_{Ai}(p)} \in object_{i(i=1,2...n)} \end{cases} \begin{cases} W(\omega_{Ai}(p), L_{i(i=1,2...n)}) = \exp\left(-\left\| I_p - I_{\omega_{Ai}(p)} \right\|\right) \\ W(\omega_{Ai}(p), B) = 0 \end{cases} \quad (10)$$

$$if\ I'_p \in Background \begin{cases} W(p, L_{i(i=1,2...n)}) = 0 \\ W(p, B) = \infty \end{cases} \quad (11)$$

Where $object_{i(i=1,2...n)}$ is one of the objects to track, I_p is the image pixel p in the frame I, ω_{Ai} is affine motion model of object i, $I'_{\omega Ai(p)}$ is the match of pixel p in the frame I' using motion model, $L_{i(i=1...n)}$ is the terminal i (corresponds to the object i), B is the terminal corresponds to the background. (9), (10) and (11) are used to calculate the costs of the edges connecting vertices (pixels) with terminals. All neighboring vertices are connected by edges, *n-links*, whose capacities is calculated by (7).

5. EXPERIMENTATIONS AND RESULTS

5.1 Data

The method has been tested on five sequences (*taxit*, *rond-point*, *walk*, *croisement* and *Coastguarder*). The first sequence has 41 frames of 256 x 191. The next sequence has 33 frames of 256 x 224. The third sequence has 127 frames of 300 x 200. The fourth sequence has 27 frames of 300 x 200.The last sequence has 80 frames of 300 x 240.

In *rond-point*, *croisement* and *coastguarder* sequence, the camera is in movement, and it is fixed in two other sequences.

5.2 Results and Complexity

The graph cuts approach is often used to solve a labeling problem. Knowing that, the complexity to solve such a problem depends on the number of pixels being matched. And in order to reduce a complexity of graph cuts labeling, we only used the rectangle which contains the object to treat instead of using the entire image, this method allows us to reduce the size of the graph. By this reduction in data size, and by using the fast algorithm described in [23], the graph cut is made in a real-time.

On figure 9, we can on the one hand see that the curve representing the number of pixels detected in the prediction step is below the curve represents the number of pixels of object refinement. On the other hand, the curve representing the number of pixels rejected by the refinement step never touches the x-axis, which shows that the motion model (called dominant motion) does not always cover all the pixels of the tracked object. Then the object predicted in the first step, prediction step, is only used as initialization to the next step (refinement step).

Figure 10 shows the change in the size of the object between the first and the last frame. This change in size may be due to rapprochement or being away of the object from the camera or due to the change in the topology. We have overcome these problems, the problem of the change in the size and the problem of the motion model that does not cover all the pixels of the object, by the stage of refinement.

We have implemented our approach in C++. And for all tests that we did, we used 2x2 GHz Intel Core 2 Duo with 2GB in memory.

Table 1: means execution time per frame (note that for reducing the complexity of graph cuts labeling, we only used the rectangle which contains the object to refine instead of using the entire image that reduces the size of graph).

Computation of motion model	Build a graph	Solve a graph cut
0.013 s	0.005 s	0.021 s

In case of the objects tracking that are not overlapping, total time per frame is 0.039 s. In this case, the tracking can be obtained in real time. In the other case, to this cost we must add the computation time required to classify the pixels of overlapping objects (=number of objects * time to build and solve a graph cuts) for example, for croisement sequence (two objects overlap) means execution time per frame is 0.094 s.

At first sight, we note that we have a great difference between the results (figure 8); the tracking resulting from combining Motion-model with Graph cuts is visually more satisfactory than those obtained by using only a parameter motion model modeled by affine motion. Figure 11 and 12 gives some examples of the experimental results.

We can see in the last frame of *croisement* sequence (figure 12) the reappearance of the object (black car) while the program could not detect it. This is normal because the object has disappeared in the previous frame and in the algorithm which we proposed the first step uses the object selected by the user or the object detected in the previous step.

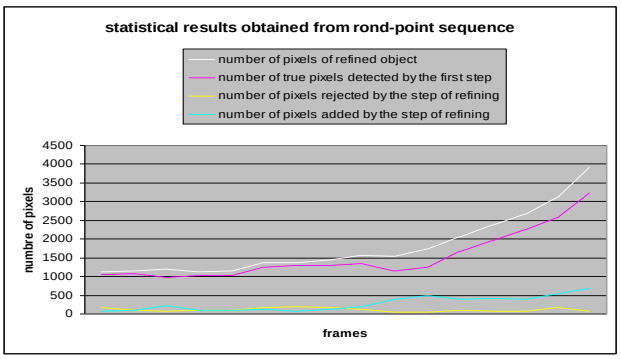


Figure 9: statistical results obtained shows the number of true pixels detected in the first step (tracking using only motion model), number of pixels added and rejected by step of refining the objects border.

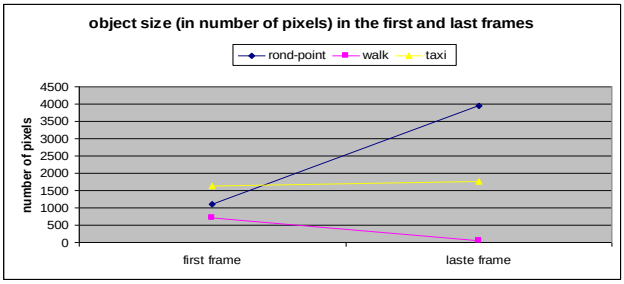


Fig 10: This figure shows the change in the size of the tracked objects. This change is taken into account in the step of refinement of objects.

6. CONCLUSION AND FUTURE WORK

In this paper we presented how we can track moving objects in video using the graph cuts approach and optical flow modeled by affine motion. We showed how to refine the border of predicted objects. The combined approaches produce dense and accurate optical flow fields. It provides a powerful and closed representation of the local brightness structure. Globally, the evaluation showed that the proposed approach gives good results according to our test. Further research will concentrate on improving the algorithm by defining criteria for identifying what properties characterize objects and distinguish them from other objects and from the background that will allow us to find the disappeared object. We will also try using more sophisticated initialization of seeds (automatic initialization).

Fig 11: Result obtained in frames 7, 13, 18 and 22 from *rond-point* sequence

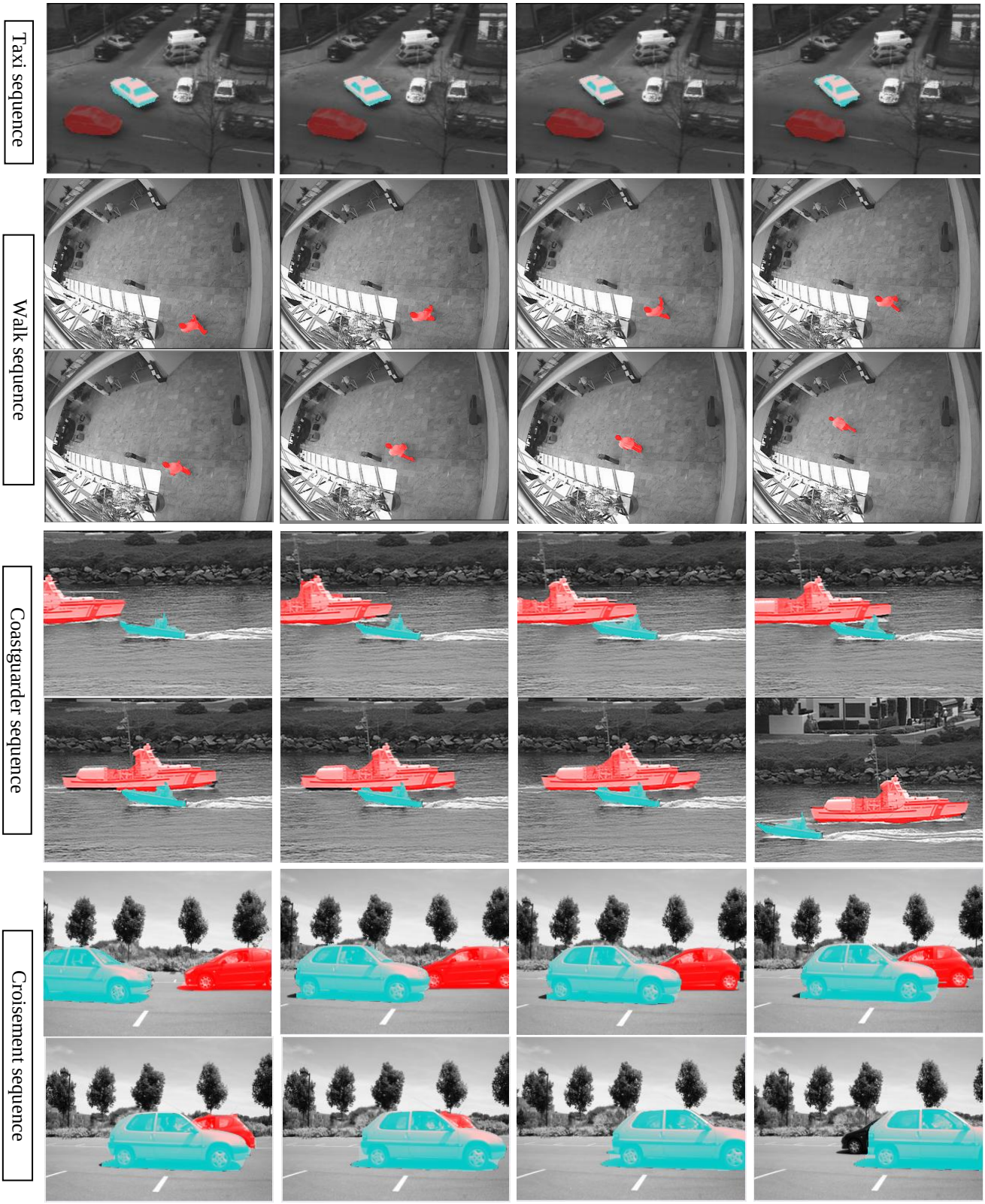


Fig 12: Experimental results. We can see in the last frame of *croisement* sequence the reappearance of the object (black car) while the program could not detect it. This is normal because the object is disappeared in the previous frame and in the algorithm which we proposed the first step uses the object selected by the user or the object detected in the previous step

7. REFERENCES

[1] D. Terzopoulos, R. Szeliski. 1993. Tracking with kalman snakes. Active vision, pp.3–20.

[2] M. Isard, A. Blake. 1998. Condensation – conditional density propagation for visual tracking. Int. J. Computer Vision, 29(1) :5–28.

[3] J. MacCormick, A. Blake. 2000. A probabilistic exclusion principle for tracking multiple objects. Int. J. Computer Vision, 39(1) :57–71.

[4] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland. 1997. Pfinder: real-time tracking of the human body. IEEE Trans. Pattern Anal.Mach. Intell. 19(7), 780–785.

[5] N. Xu, N. Ahuja. 2002. Object contour tracking using graph cuts based active contours. Proc. Int. Conf. Image Processing.

[6] B. Rosenhahn, U. Kersting, S. Andrew, T. Brox, R. Klette and H-P. Seidel. 2005. A silhouette based human motion tracking system Technical Report (The University of Auckland, New Zealand: CITR) ISSN: 1178-3581.

[7] B. K..P. Horn, B. G. Schunck. 1981. Determining Optical Flow. AI 17, pp. 185—203.

[8] J. Barron, D. Fleet, S. Beauchemin. 1994. Performance of optical flow techniques. Int. J. of Comput. Vis. 12(1), 43–77.

[9] A.R. Mansouri, J. Konrad. 2003. Multiple motion segmentation with level sets. IEEE Trans. Image Process. 12(2), 201–220.

[10] G. Adiv. 1985. Determining 3D motion and structure from optical flow generated by several moving objects. IEEE Trans. Pattern Anal. Mach. Intell. 7(4), 384–401.

[11] A. Yilmaz, O. Javed, M. Shah. 2006. Object tracking: A survey. ACM Comput.Surv. 38, 13.

[12] C. Veenman, M. Reinders, and E. Backer. 2001. Resolving motion correspondence for densely moving points. IEEE Trans. Patt. Analy. Mach. Intell. 23, 1, 54–72.

[13] D. Serby, S. Koller-Meier, and L. V. Gool. 2004. Probabilistic object tracking using multiple features. In IEEE International Conference of Pattern Recognition (ICPR). 184–187.

[14] D. Comaniciu, V. Ramesh and P. Meer. 2003. Kernel-based object tracking. IEEE Trans. Patt. Analy.Mach. Intell. 25, 564–575.

[15] Y. Boykov, O. Veksler, and R. Zabih. 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239.

[16] Y. Boykov, M.P. July. 2001. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In: proceedings of the International Conference on Computer Vision, Vancouver Canada.

[17] K. Housni, D. Mammass and Y. Chahir. 2009. Interactive ROI Segmentation using Graph Cuts, GVIP-ICGST Journal,Volume 9, Issue 6, pp 1—6..

[18] D. Greig, B. Porteous, and A. Seheult. 1989. Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society, Series B, 51(2):271–279.

[19] V. Kwatra, A. Schodl, I. Essa, and A. Bobick. 2003. Graphcut textures: image and video synthesis using graph cuts. In Proceedings of SIGGRAPH.

[20] V. Kolmogorov. 2004. What Energy Functions Can Be Minimized via Graph Cuts?. IEEE transactions on pattern analysis and machine intelligence, vol. 26, NO. 2.

[21] L. Ford and D. Fulkerson. 1962. Flows in Networks. Princeton University Press.

[22] A. Goldberg and R. Tarjan October. 1988. A new approach to the maximum flow problem. Journal of the Association for Computing Machinery, 35(4):921–940.

[23] Y. Boykov and V. Kolmogorov. 2000. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. 3rd. International Workshop on EMMCVPR. Springer-Verla, September.

[24] J.-M. Odobez, P. Bouthemy. 1995. Robust multiresolution estimation of parametric motion models. Journal of Visual Communication and Image Representation, 6(4):348-365.

[25] Vista team, http://www.irisa.fr/vista