IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing

A Parallel Method For Object Tracking

Cecilia Maria Buarque Fredrich, Raul Queiroz Feitosa and Marco Antonio Meggiolaro
Department of Electrical Engineering and Department of Mechanical Engineering
PUC-Rio
Rio de Janeiro, Brazil
buarque@ele.puc-rio.br, raul@ele.puc-rio.br and meggi@puc-rio.br

***Abstract*—Computer Vision-based techniques are powerful tools for designing efficient control systems. Cameras provide information, through which an industrial manipulator can, for instance, have its trajectory corrected with respect to the target object. This paper proposes a real-time object tracking method that is both robust and able to deal with different environment circumstances and scenarios. It also deals with the quality level of the camera available and also the interest object's nature itself, which may vary quite significantly from one application to another. The method relies on parallel processing for building the model that is best suited to the current scenario, thus dismissing heuristics for selecting the most adequate features. Two variants were devised to cope with a number of different scenarios, as well as equipments, which accounts for the likeness of the method to succeed in a great deal of applications.**

Keywords—point matching; NCC; LSM; SIFT; parallel processing.

I. INTRODUCTION

Object recognition is a twofold issue. That is, provided there is (at least) one model of the interest object, the matter of recognizing it entails *identifying* this object in an image and, then, *locating* its position in 3-D space.

Object identification can be perceived as determining which model in the database matches the data in the input image. In other words, it is essentially a matter of comparing input data to a model to decide whether they match. Thus, the success of an identification algorithm relies a great deal on how effective a model is at describing a given object.

This paper presents a method for object tracking that is based upon features, namely points, which must be described by a collection of characteristics, so that it can be identified, assessed and compared with. Such collection is referred to as *feature descriptor*. Several different descriptors are used, as shown in section II.

The ability of a descriptor to represent a feature is crucial for devising an effective object recognition method. Mikolajczyk and Schmid [1] survey the performance of several feature descriptors. They were evaluated, compared to one another, and finally ranked by the authors, who attest that the ones based upon image region perform best. Indeed, the method proposed herein uses only this kind of descriptors, with varying degrees of complexity.

Two fundamental issues present the greatest difficulties, as far as the problem of object recognition based on image points is concerned: processing time and exactly *which* points are the most suitable to use as both input data and model. The first is a stumbling block for real-time applications, due to the computation cost involving the state-of-art algorithms for point detection and matching. The latter issue concerns more directly the quality of the tracking algorithm. Not only all object views must be modeled by a single set of image points – so that it can be identified (or tracked) at all times – but also the tracking system must account for problems regarding the environment conditions – e.g. occlusion, varying illumination patterns and view angle – and be robust to them. This means that the input features have to be equally well representative and furthermore less subjected to a false match.

The purpose of this work is to devise a novel real-time method for object tracking, based upon parallel processing and point correspondence. Once implemented, it would be desirable for the method to be capable of functioning in various conditions. Hence, several video sequences are used to address this matter. Being of different natures – regarding not only the environment, but also equipment and interest object – they provide ground for attesting the method's efficiency in various applications and explain the reason for proposing two variants. Tests using two of these sequences are depicted herein; the full records of the experiments are, however, documented in [2].

The remainder of this paper is organized as follows. The method itself is presented in section II, which contains the ideas underlying its devising, as well as the details of both of its variants. Section III describes both the database and the experiments used for accessing the method's performance. The results are presented in section IV, followed by concluding remarks.

II. THE METHOD

A. Overview

This section introduces two algorithms devised for tracking keypoints (i.e., points of a scene that can be identified unmistakably on images taken from different views) in a video sequence. The general guidelines given

in this preamble will help understanding their broad functioning, for they may actually be perceived as two variants of the same method, rather than different approaches to tackle the same problem.

Generally, in a video sequence, two subsequent frames are converted into the image pair for tracking each keypoint position. The processing time of the matching procedure determines the frame sampling rate; that is, the faster the procedure, the closer (in time scale) the two frames.

Consecutive frames that are closer in time tend to be more similar to one another, making keypoints easier to trace. Thus, in such cases, the robustness demanded by the matching technique may drop, to a certain degree, without corrupting the tracking progress. That said, one is left with the circular tradeoff dilemma seen in Fig. 1.

Environmental problems, such as occlusion and change in illumination, make the selection of the initial keypoints rather troublesome and in fact suggest such collection of characteristics should not be static. Which brings on the matter of how to assess the suitability of a keypoint to the current scene, as addressed in [3] and [4], for instance.

The proposed method does not need to implement any heuristics in such assessment. The adequacy of the collection is automatically assured by the fact that its entries are the keypoints found, as close in time as possible, to the sequence frame being processed at a given instant. Thus, they have a higher chance of being consistent with the upcoming scene. This is achieved by parallel processing, i.e., separate threads for the tracking and updating processes.

The updating strategy is as follows. Using the past matching sequence frames, a thread should be held responsible for carrying on the collection update. New features, in these last frames, are matched to the model image(s) to assemble a new starting list, i.e., the keypoint collection to be processed by the next tracking thread.

The block diagram depicted in Fig. 2 illustrates the overall process throughout time. Having introduced the kernel of the object tracking process, the two variants devised for running in the matching thread are presented in detail next.

B. SLN Variant

At the very beginning of the tracking process, there is hardly any knowledge about the camera position, with

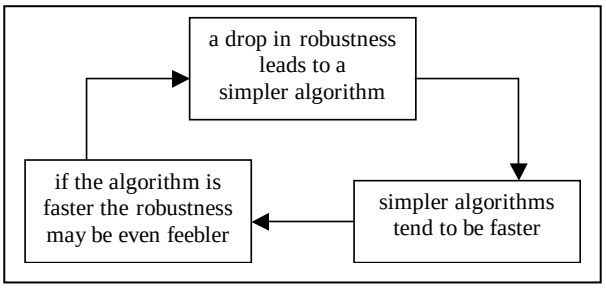

Figure 1. Scheme that drove the devising of the object tracking method.

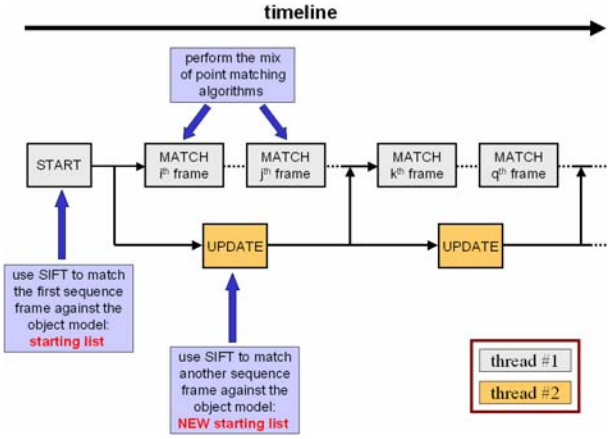

Figure 2. Block diagram of the object tracking process (SIFT stands for Scale Invariant Feature Transform).

respect to the pose of the object model (or the keypoint collection). Hence, the point correspondence in the template-matching frame pair at this stage is rather more difficult, with respect to the ones hereafter. Therefore, a more robust procedure should be adopted.

From hereon, as to assure keypoint stability and pairing consistency, an algorithm that can not only provide a match, but also eliminate false initial matches, is desirable. As the video sequence moves further in time and the camera movements have stabilized, as well as the false matches have been eliminated, the tracking process will not require a great filtering ability nor a wide image transformation coverage. Consequently, the matching algorithm can be even simpler as the current tracking thread advances in time.

Based upon the above discussion, the mix proposed for this variant is as follows.

- Apply the Scale Invariant Feature Transform (SIFT) algorithm [5,6] and match the starting frame against the model.
- Whenever the former step is completed, grab the incoming frame and use Least Squares Matching (LSM) [7,8] to correct the coordinates of all keypoints, using their present values (i.e. the SIFT output) as initial guesses and with the last processed frame (the SIFT-frame) as the template.
- Consecutively apply Normalized Cross Correlation (NCC) from then on to refresh the keypoints that are still fit. The initial guesses and template are analogous to the LSM-step.

This procedure restarts every time a collection update is released. Fig. 3 shows the object tracking scheme.

C. SLS Variant

Both the LSM and the NCC algorithms have matrices of neighboring image pixel values as descriptors. Hence, not only they behave rather poorly in cases where the geometric difference between the pairs of frames is too significant – such as in high movement sequences – but they also may become quite unstable when processing severely compressed video sequences,

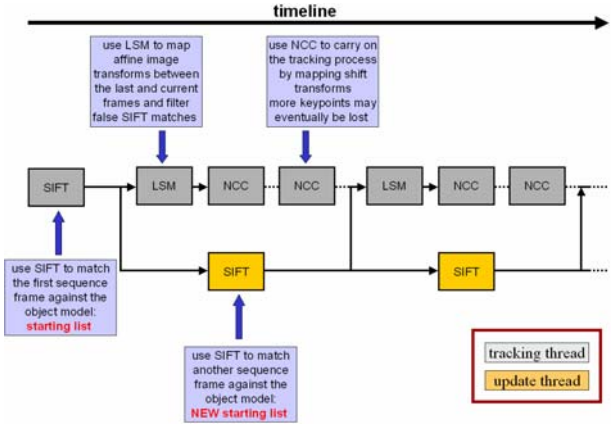

Figure 3. Block diagram of the object tracking method: SLN variant.

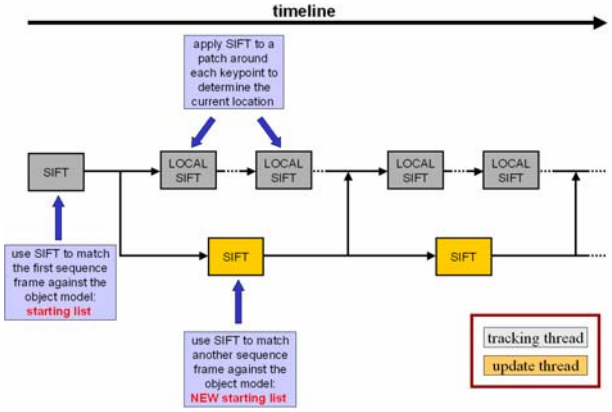

Figure 4. Block diagram of the object tracking method: SLS variant.

namely those where the quantization blocks are noticeable and a pixel's neighbours – and even location – are far from corresponding to the ones in the original uncompressed frames.

This second variant was devised to deal with such sequences and its functioning is very similar to its peer's. Here, the LSM-NCC routine is replaced by a SIFT-based matching technique. Fig. 4 depicts the alternative matching strategy.

Running the SIFT algorithm as implemented in the first step throughout the whole sequence would incur in a massive loss of image frames, due to its processing time. Again, if the subsequent frames used as template-matching pairs are kept close enough in time to one another, it is fair to admit that a keypoint location will not wander off, in terms of absolute image coordinates, regardless of the significance of the geometric transformation it went through. Therefore, processing only a patch of the matching frame, around the coordinates of each template keypoint, should be enough to determine the new location.

III. Experiment Design

Two sequences, from two different cameras, were used to evaluate the method's performance. They are:

- *Fast Pre-Amp.*: High movement JPEG-compressed sequence, where the interest object, an ancient preamplifier unit, placed in a dry environment, is the only one appearing in the scene. However, the background is still not perfectly neutral and illumination changes and incidence cause it to incur perceptible alterations in contrast. The incidence of keypoints is significantly low, due to the object's own nature (i.e., its appearance that is remarkably uniform), which adds difficulty to the tracking conditions. The sequence is processed by the SLN variant.
- *Pool*: Low movement uncompressed sequence, shot underwater in a (rather dirty) pool. The object is an underwater reservoir, located in a fairly more interesting environment, i.e., full of potential keypoint candidates, to be recognized by the SIFT algorithm. The sequence is processed by the SLS variant.

The degree of movement and the video compression were the main criteria for selecting the most suitable variant in each case. The method's performance was assessed from:

- Charts of total number of keypoints throughout time that show the cyclical routine of continuous drops – as frames are processed – and sudden increases – whenever an updated is released.
- Records of the actual keypoint locations in all processed frames, merged into an output movie, in which each one that was eventually lost was replaced by the last processed frame.

IV. Results

This section reports the results of the performance measures previously described.

Figures 5 and 6 show the charts generated from the *Fast Pre-Amp.* and the *Pool* sequences, respectively. Time – measured in seconds – elapsed during the starting step of the object tracking process (the "START" block on Fig. 2) was excluded from these records.

The SLN and SLS variants are instantly recognizable from their general response depicted in the charts. They

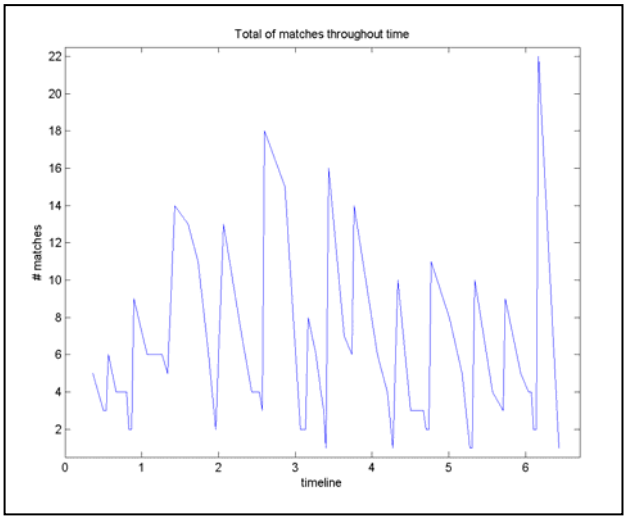

Figure 5. Fast Pre-Amp.: Incidence of matched keypoints throughout time (SLS variant).

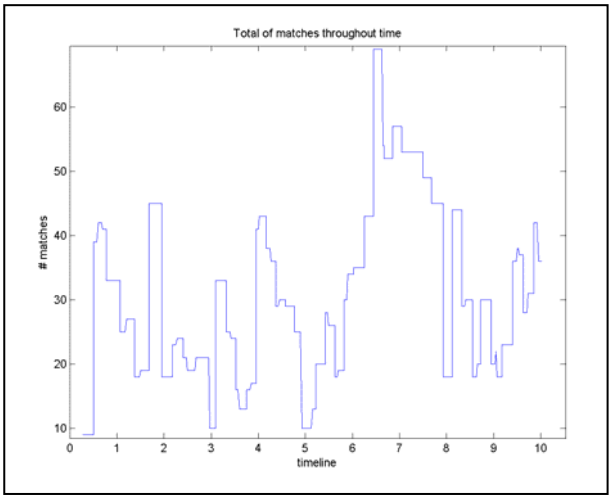

Figure 6. Pool: Incidence of matched keypoints throughout time (SLN variant).

show that SLN is able to track more keypoints, missing less frames, as indicate the series of short steady periods, during which the total number of keypoints remains the same. The update and tracking processes are more easily identified on the SLS chart, for subsequent scenes. When further apart in time they hardly ever have the same number of keypoints. The overall duration of a tracking chain is bound by the time needed for the next update release. However, in SLN's tracking chain ("thread #1" on Fig. 2), each link is faster, which is crucial to the effectiveness of the area-based keypoint descriptors (LSM and NCC procedures) in this case. Hence, this accounts for the steadier behavior of this variant, in terms of match losses.

As visual aid for the performance evaluation, mosaics showing the different localizations of some keypoints throughout a few frames, i.e., describing how the object moves along the scene, are presented next. A color change denotes that an update has taken place and, in this case, kept the feature in the tracking list. Keypoint locations on Figs. 7 and 8 were rounded for the display. Consecutive frames on the mosaics do not necessarily correspond to consecutive processed frames. They followed, nevertheless, the time line.

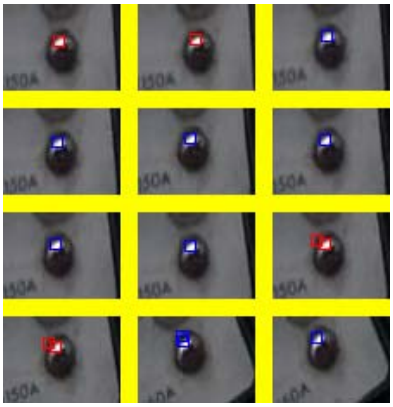

Figure 7. Fast Pre-Amp.: a keypoint moving throughout time (row-wise, from top left). A keypoint color switch indicates that an update occurred sometime between these consecutive framelets.

At first, in the cases of sequences processed by the SLN variant, it might seem like both versions of the proposed method could actually be used interchangeably. In fact, the SLS variant would work well with all sequences. However, the former produces a greater average number of matches. Thus, the decision for one or the other is essentially a tradeoff between match incidence and range of transformation mapping.

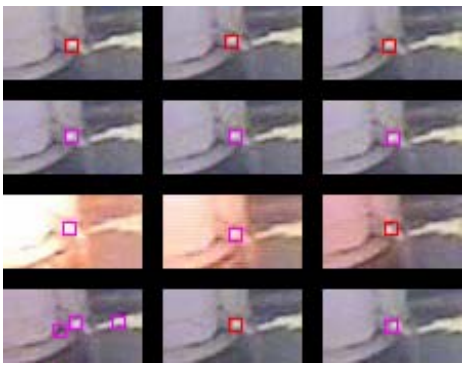

Figure 8. Pool: a keypoint moving throughout time (row-wise, from top left to bottom right). A keypoint colour switch indicates that an update occurred sometime between these consecutive framelets.

V. Concluding Remarks

This work aimed at the development of a novel method for tracking a target object in real-time. By determining the locations of several keypoints throughout a video sequence, the object's pose can then be estimated. Although it was meant for self-sufficient use, the proposed algorithm may also be embedded in a control system to function as the visual-based fine tuning stage, where it would be occasionally triggered.

Parallel processing automatically assures that the set of keypoints is always consistent to the current scene, thus discarding algorithms for monitoring or controlling these features' incidence. Two variants of the method were designed to increase its robustness and, thus, widen its applicability in the face of the diversity of working conditions that may be faced. Indeed, the procedure proved to be case-driven, dependent on the inherent qualities of the sample sequences.

References

[1] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. on Pattern Analysis & Machine Intel., vol.27, pp.1615–1630, 2005.

[2] C. M. B. Fredrich, A Parallel Method for Object Tracking, MSc Thesis, Department of Electrical Engineering. Rio de Janeiro, RJ: PUC-Rio, June 2009.

[3] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," Proc. IEEE Int. Conf. on Rob. Autom. (ICRA), pp. 2051–2058, 2001.

[4] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," Int. Conf. on Intel. Robots and Systems, Switzerland, pp.226–231, 2002.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," Int. Conf. Comp. Vision, Greece, pp.1150–7, 1999.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comp. Vision, vol.60, pp.91–100, 2004.

[7] F. Ackermann, "Digital image corrrelation: performance and potential application in photogrammetry," The Photogrammetric Record, vol.11, pp.429–439, 1984.

[8] A. W. Gruen, "Adaptive least squares correlation: a powerful image matching technique," South African J. Photogrammetry, Remote Sensing and Cartography, vol.14, pp.175–187, 1985.