

A Novel Optical Tracking Algorithm for Point-Based Projective Invariant Marker Patterns

Manuel Loaiza¹, Alberto Raposo¹, and Marcelo Gattass¹

¹ Tecgraf - Computer Graphics Technology Group
Computer Science Department
Pontifical Catholic University of Rio de Janeiro
R. M. S. Vicente, 225 – Gávea, Rio de Janeiro, RJ - Brazil
{manuel,abraposo,mgattass}@tecgraf.puc-rio.br

Abstract. In this paper, we describe a novel algorithm to group, label, identify and perform optical tracking of marker sets, which are grouped into two specific configurations, and whose projective invariant properties will allow obtaining a unique identification for each predefined marker pattern. These configurations are formed by 4 collinear and 5 coplanar markers. This unique identification is used to correctly recognize various and different marker patterns inside the same tracking area, in real time. The algorithm only needs image coordinates of markers to perform the identification of marker patterns. For grouping the dispersed markers that appear in the image, the algorithm uses a “divide and conquer” strategy to segment the image and give some neighborhood reference among markers.

Keywords: Point-based Feature, Point Set Matching, Projective Invariants, Optical Tracking.

1 Introduction

For many years, tracking systems have been essential and intensively used tools in the implementation of virtual and augmented reality applications. Their purpose is to track different objects or markers, either individually or as a group, within a predefined area. Among the diverse technologies used to develop these tracking devices, optical technologies have been one of the mostly used due to some advantages they have in relation to others, such as: their sensors require no wires, they are less sensitive to noise, and they allow incrementing simultaneously the number of markers tracked within an area without significantly affecting the system, especially hardware.

Optical tracking systems often have a well structured and standardized hardware and software architecture that can be summarized as follows:

Concerning hardware, this type of tracking is usually composed of video capturing devices and different types of markers, which can range from small retro-reflective spheres, IR (infrared) leds, up to features of the scene itself, such as corners, lines and other characteristic details the system might use as tracking markers.

Regarding software, optical tracking systems present a more standardized operation flow based on the implementation of different computer vision techniques with the purpose of extracting, recognizing and tracking markers. Normally image processing is the first technique implemented and has the objective of extracting the 2D coordinates that represent markers' positions within the image captured from the tracking area. Such coordinates constitute the principal information on which the system will operate.

The following stages are the stereo matching of the marker sets appearing in the images (when more than one camera is used) and the 3D reconstruction of the markers based on information about the cameras and the 2D coordinates provided by image processing. After 3D reconstruction, identification of markers generates expensive combinatory operations due to wrong results during matching stage.

The matching stage is usually made with stereo correspondence and epipolar geometry. Due to the lack of a previous standardized identification of markers included in 2D coordinates within the image, false matches and subsequent wrongly reconstructed 3D points are generated. To deal with this problem, other heuristics are required to discard these false markers, such as using 3D metrics to find the correct combination of markers that belong to a specific pattern. These metrics may be implemented based on distances, angles or graphs representing markers distribution.

It is precisely in this stage prior to matching that the algorithm proposed in this paper will be used to reach a way of individually grouping and identifying in 2D the markers sets that form a specific pattern to be tracked. The objective of the proposed algorithm is to be a support tool to reduce the number of cases of false markers generated in the matching stage and subsequently in the 3D reconstruction stage.

This paper is organized as follows: in the next section previous work is discussed. In section 3 the configuration of the hardware system used is presented. In section 4 the theoretical part of computer vision techniques used to implement the algorithm is described, as well as the algorithm's process flow. In section 5 some results are presented, and in section 6 we draw conclusions and future work.

2 Related Work

The marker matching and pattern identification process has been widely investigated in the area of pattern recognition while being used in computer vision. Often, robust and well developed techniques in the pattern recognition area are "exported" to be used in computer vision. In the case of the proposed algorithm, we have used two techniques employed in pattern recognition, here applied in the implementation of marker pattern matching in an optical tracking system. These two techniques on which the algorithm is based are the theory about projective invariant properties described in [3],[8],[10] and the "divide and conquer" strategy exemplified by the use of a quadtree to segment an image where the leaves will be the image coordinates of the markers, which are spread over the image.

The use and efficiency of these techniques have been presented in separate works. The implementation of projective invariant properties for pattern recognition was presented in [11],[12]. Image segmentation based on the "divide and conquer"

strategy was presented in [6],[7]. In these works one can see that both techniques have good individual performance, but so far no work presented an integration of these techniques. This was a further motivation behind the present work, which intends to show that these techniques can yield good results by working together.

3 System Setup

To run and test the algorithm, the implementation of a basic optical tracking system was required. It is based on the standard architecture for optical tracking systems, i.e. an architecture based on cameras with infrared light spots and filters that irradiate a scene containing markers covered with retro-reflective material, with the purpose of highlighting the markers in relation to other objects present in the scene. The idea behind this architecture was used in our system, which also uses cameras with infrared filters but without the light spots – instead, the markers are small 5mm and 2mm incandescent lights powered by a set of batteries for each pattern. As was previously defined, our tracking patterns will have a predefined format with the following configurations:

- Pattern I: formed by 4 markers placed in a collinear manner.
- Pattern II: formed by 5 markers placed in a coplanar manner, where one marker will be surrounded by the 4 other markers.

These formats can be seen in Fig. 1 and will be further explained in section 4.3.

4 The proposed algorithm

In this section we will briefly summarize the theory behind the techniques used in the implementation of the proposed algorithm, and present a general description of it.

The algorithm's purpose is to group, label, individually identify each pattern, and track it optically in real time, having as operational data only the image coordinates of the markers that compose the tracking patterns. To achieve this goal, the algorithm has two steps: firstly an offline step used for training and generation of a unique identifier to each pattern to be tracked, and secondly an online step that will be the basic optical tracking system, running in real time, used to test the algorithm.

The main techniques used in each step will be presented as follows. Then, the process flow of each step of the algorithm will be presented.

4.1 Image Processing

The image processing used in the algorithm has as main goal analyzing and extracting a 2D representation for each marker displayed on the video images captured by the tracking system cameras. This 2D representation indicates the position of the markers in image coordinates. This sub-process is composed by a set of techniques that analyze the image sequentially, according to the following action flow:

- Capture the video images with the devices used (cameras).

- Convert each image to a single grayscale channel and apply a threshold filter to make it binary.
- Apply a connected-component algorithm to identify circular areas that will be the representations projected on the marker images.
- Extract the center of each connected area as the image coordinate for the marker candidate inside the image.

To implement these techniques in the proposed algorithm, the OpenCV [4] library was employed. This technique is widely used in diverse implementations of optical tracking systems, both commercial [1] and academic [5].

4.2 Quadtree

The quadtree implementation has the purpose of grouping, in quadrants, the total set of markers spread throughout the image. This grouping is based on the principle that markers belonging to the same pattern should be very close to one another. The key step to take advantage of this marker quasi-grouping is reading and traversing the quadtree to extract very close subgroups of markers. This way, candidate subgroups of markers will be formed and tested, and they will contain the tracking patterns inside the tracked image.

The process of traversing the quadtree and generating the test candidate pattern groups follows this action flow:

- Generate a quadtree for each group of markers detected in the image processing stage for each frame captured by the camera.
- Traverse each branch of the quadtree to find pattern I and then pattern II.
 - If the tested node doesn't have a child with less than 4 (or 5 depending of what pattern we are testing) markers then
 - Generate test patterns by combining markers as C_n^4 (or C_n^5), where "n" is the number of leaf nodes the parent has, provided that "n" ≥ 4 (or "n" ≥ 5) correspondingly.
 - Else, move down through the 4 leaves of the node running the previous step recursively
- After generating the combinations to compose the marker sets, the test of correspondence between each of these sets against the patterns defined in the training stage is done.
- If some set is correctly matched to a given pattern, then the markers forming the matching group are removed from the quadtree structure.
- The unmatched markers return to the parent node identified as unmatched, and will be part of the same matching process with other unmatched children nodes of the same parent.

As can be seen, matching is a recursive process made at each branch, firstly top-down but becoming bottom-up as the markers are matched in each branch analyzed.

The role of the quadtree is merely as a tool that implicitly provides a neighborhood reference among the markers. An example of this implementation in our system is showed in Fig. 1.



Fig. 1. Grouping for image coordinates of our patterns done with a quadtree.

4.3 Projective Invariants

The implementation of projective invariant properties has been discussed especially in the area of pattern recognition [3],[10]. It is based on the invariant property of cross ratio, particularly in the case of perspective projection. The cross ratio property states that if we have 4 collinear points (A,B,C,D) we can define a cross ratio value based on these points' distances according to the following relationship:

$$\text{Cross ratio (A,B,C,D)} = \frac{|AC|/|BC|}{|AD|/|BD|} \quad (1)$$

This property has been expanded in order to cover not only collinear patterns but also patterns with coplanar markers. In this case, the cross ratio is obtained based on the areas of the triangles generated using combinations of the vertices that constitute a coplanar pattern, as shown in [9]. This extended variation of cross ratio projective invariant properties applied to coplanar points was developed until it became invariant to possible changes in the labels that compose coplanar patterns [2].

In these works, the identifier of the projective invariant for a set of 5 coplanar points is defined as a vector with 5 ranges, with a minimum and maximum value for each vector position. This vector is obtained by applying the P^2 -Invariant technique [2] over the 5-point sample, where each vector position is related to each marker in the set. According to [3] this allows not only group matching but also individual matching for each marker in the 5-point set. The only drawback is the need to generate the whole 5-range vector for each candidate group of markers to be matched to a given pattern. In [10] an improvement in relation to [3] was presented, especially to the case of 5 coplanar points. It was demonstrated that the 5-range vector can be reduced to a 2-range vector. In fact, this unique 2-range vector corresponds to the values of the base projective invariants for an especially 5-coplanar pattern that has the following restriction: in the tracked 5-coplanar pattern a specific marker from the group must be well recognized in all views while the pattern is being tracked. This is required because the value of the projective invariant vector is generated based on the cross ratio of the triangles formed by the combination of the vertices, where that

specific marker is part of the 4 triangles used to compute the cross ratio. Another overall restriction is that no 3-marker set in a group of 5 be collinear, as this would generate a null triangle area.

Finally, based on the results of the works described above, we have defined the configuration of the patterns used in the algorithm proposed here. Pattern I has two invariant features to perspective projection: collinearity and cross ratio. Pattern II has 4 points around a 5th point, with the need to identify a specific marker in the 5-marker set – in this case, the central marker. These characteristics not only allow computing the value(s) of the properties that are invariant to projection and permutation, but also this specific configuration can be used as filter to quickly discard false candidates.

4.4 Auxiliary Methods

Some auxiliary techniques were implemented to further restrict the selection of marker sets candidate for tracking patterns. These techniques are used as filters that were implemented based on specific characteristics of each type of patterns.

The first one is collinearity test of the 2D coordinates of the markers to be analyzed in a given candidate set. As was already described, each time the algorithm moves down the quadtree branches it will stop at branches containing 4 or more markers (leaves). In the case of pattern I, the algorithm will test first the collinearity of the markers that compound a candidate to be matched. This quick test discards in advance false candidates for pattern I. A second filtering is the matching using the projective invariant property value.

In the case of pattern II, the subgroup of 5 candidate coordinates will be evaluated through the generation of a convex hull. This test is made by checking if 4 coordinates form a convex hull enclosing a fifth coordinate. This restriction is based on the shape defined by the configuration of pattern II, as previously discussed, and used like the first filter to recognize candidates for pattern II.

A second filtering is optional and can be added before matching the projective invariant properties. It consists in creating an oriented and expanded bounding area around each correctly identified pattern in a frame “t”. This tool can be used as a simple way to predict and restrict the area where a well recognized pattern may appear, based on information from frame “t” in frame “t+1”. This second filtering can only be executed after a pattern is identified by matching the projective invariant properties in a previous frame.

4.5 The algorithm

The proposed algorithm is divided in two stages: training and tracking. In both stages the system presents process flows integrating the basic techniques described in previous sections.

The training stage executes the following flow:

1. Process the image to generate the 2D coordinates representing the markers' positions in the image.

2. Generate a line in the case of pattern I or a convex hull around the center point in the case of pattern II.
3. Generate the unique identifier defined by the range containing a minimum and maximum value for pattern I, and the identifying vector composed by 2 ranges with minimum and maximum values for pattern II.

The tracking stage flow is composed of:

1. Process the image to generate the 2D coordinates representing the markers' positions in the image.
2. Generate the quadtree for each frame using the 2D coordinates of the markers as data.
3. Create sets of 4 markers using the concept of neighborhood provided by the quadtree based on the position of the 2D coordinates of the markers. These sets will be candidates for patterns of the tracking system.
4. Discard some candidate sets by running the collinearity test over the 4-marker sets.
5. For each marker set, generate the respective identifier, for pattern I.
6. Compare the values of the identifier generated for each candidate set tested against the pattern I values predefined in the training stage.
7. Remove from the quadtree each marker set, which is correctly matched with some specific pattern I.
8. With the remaining markers, create sets of 5 markers using again the concept of neighborhood provided by the quadtree structure.
9. Discard some candidates' sets by running the first filter to coplanar pattern.
10. For each marker set, generate the respective identifier, for pattern II.
11. Compare the values of the identifier generated for each candidate set tested against the pattern II values predefined in the training stage.
12. Label the matched and recognized pattern sets as one of the training patterns, in order to track them frame by frame.

Once the tracking stage is under execution identifying each pattern individually, we can insert two sub-processes to help the process of discarding false candidates:

- In step 12, once the pattern is identified, we can generate a 2D extended bounding box around markers that compose each pattern. This is helpful to predict where the pattern will appear in the next frame.
- In steps 4 and 9, test whether the candidates' sets that satisfy the collinearity tests are inside the 2D bounding boxes created in a previous frame for each respective pattern.

5 Results

To test the algorithm, we used the basic tracking system with a single camera and used the algorithm to track a set of 2 collinear patterns of type I, and 2 coplanar patterns of type II (Fig. 2). The goal of the test was to measure the efficiency of the algorithm in a continuous tracking of the patterns and to show its robustness in the discard of false candidates during the matching stage.



Fig. 2. Optical tracking test with two patterns of each type in the image.

The first step was to make the individual training for each pattern to generate a unique identification, based on the projective and permutations invariant values calculated for each pattern. This training was done moving each pattern, one per time, in front of the camera for no more than 2 minutes. For each frame captured, the projective and permutation invariant values of the pattern were calculated and stored. After these 2 minutes we had a sample of approximately $2 \times 60 \times 30 = 3600$ values. Later, this sample was analyzed to get a range of values for the projective invariant values for pattern I and two ranges for pattern II (Table 1).

Table 1. Projective invariant values for each pattern.

Pattern	Type	Minimum Value	Maximum Value
1 st -Pattern	I	[2.048]	[2.085]
2 nd -Pattern	I	[2.368]	[2.460]
3 rd -Pattern	II	[0.06 , 0.06]	[0.14 , 0.13]
4 th -Pattern	II	[0.16 , 0.16]	[0.23 , 0.24]

We can observe that the range of values that have the same pattern type are disjoint, meaning the patterns with same configuration will be identified separately. In Tables 2 and 3 we show some values produced in the process of recognizing and tracking the patterns in the case showed in Fig. 2.

Table 2. Test in each quadrant for getting candidates for pattern I.

Quadrant	N° Markers	N° Candidates for Pattern I
Left – Up	4	$C^4_4 = 1$
Left – Down	5	$C^4_5 = 5$
Right – Up	0	0
Right – Down	9	$(C^4_6 =) 15 + (C^4_9 =) 126 = 141$
Total:	18	147

Table 3. Filters applied over candidates for pattern I.

Quadrant	Candidates Pattern I	Collinearity Test	Projective Inv. Match	Pattern ID	Unmatched Markers
Left – Up	1	1	1	1 st -Pattern	0
Left – Down	5	0	0	0	5
Right – Up	0	0	0	0	0
Right – Down	141	0 + 1 = 1	1	2 nd -Pattern	5
Total:	147	2	2	2 (recognized)	10

In Tables 4 and 5 we show the following process of recognizing coplanar patterns.

Table 4. Test in each quadrant for getting candidates for pattern II.

Quadrant	Nº Markers	Nº Candidates for Pattern II
Left – Up	0	0
Left – Down	5	$C_5^5 = 1$
Right – Up	0	0
Right – Down	5	$C_5^5 = 1$
Total:	10	2

Table 5. Filters applied over candidates for pattern II.

Quadrant	Candidates Pattern II	Convex Hull Test	Projective Inv. Match	Pattern ID	Unmatched Markers
Left – Up	0	0	0	4 th -Pattern	0
Left – Down	1	1	1	0	0
Right – Up	0	0	0	0	0
Right – Down	1	1	1	3 rd -Pattern	0
Total:	2	2	2	2 (recognized)	0

The strategy to remove from the quadtree the markers that were correctly matched with pattern I reduces the number of combination cases to test in the next process of recognizing the coplanar patterns. A key point in the execution of the algorithm was the implementation of an efficient routine to make the collinearity test of the candidates. For example, the case of quadrant Right-Down in Table 3 where 141 candidates are reduced to only 1 valid candidate.

Other important point is the robust sample used to train the patterns, which helped us to get a range of projective and permutation invariant values that are very flexible and good to discard false candidates after the collinear and convex hull filters. Normally, these false candidates appear as a consequence of the excessive number of possibilities generated by the random combination of markers that appear in a specific branch of our quadtree.

Finally, the tracking system was executed in a PC Pentium IV, 2.5 GHz, 2GB RAM, using a simple webcam at 30fps. In the execution of the system using the algorithm, the average frame rate was 28±2 fps.

6 Conclusions and Future Works

In this paper a novel algorithm for optical tracking of points-based patterns was presented, with the goal of making a fast matching and individual identification of predefined patterns. An interesting point of the matching process used is that all processes are implemented in a processing 2D stage only, differently from other optical tracking systems, which generally need to reconstruct the 3D position of the markers set to provide a correct matching.

As future projects, there is still space for new optimizations in the filters to discard false candidates, especially in the case of coplanar patterns. Another application area is the use of the tracking algorithm for markerless tracking. In this case, specific points, lines or textures of real objects are used as tracking markers. Since these features can have projection invariant characteristics, there is room for the use of the proposed algorithm.

References

1. Advanced Real Time Tracking GmbH. 2006. A.R.T. System. <http://www.ar-tracking.de/>.
2. Meer, P., Lenz, R., Ramakrishna, S.: Correspondence of Coplanar Features Through p2 Invariant Representations. Applications of Invariance in Computer Vision. pp. 473-492. Springer - Verlag Press (1993).
3. Meer, P., Lenz, R., Ramakrishna, S.: Efficient Invariant Representations. International Journal of Computer Vision (26). pp. 137-152. (1998).
4. OpenCV - Intel Open Source Computer Vision Library. <http://www.intel.com/technology/computing/opencv/>.
5. Ribo, M., Pinz, A., Fuhrmann, A.: A New Optical Tracking System for Virtual and Augmented Reality Applications. In: Proceedings of the IEEE Instrumentation and Measurement Technology Conference. Vol(3).pp 1932-1936. Budapest, Hungary(2001).
6. Santos, P., Stork, A., Buaes, A., Jorge, J.: Innovative Geometric Pose Reconstruction for Marker-based Single Camera Tracking. In : Proc. of ACM SIGGRAPH Int. Conf. on Virtual Reality Continuum and its Applications, pp. 237-244. ACM Press, New York, USA (2006).
7. Santos, P., Stork, André, Buaes, A., Jorge, J.: PTrack: Introducing a Novel Iterative Geometric Pose Estimation for a Marker-based Single Camera Tracking System. In: Proceedings of IEEE Virtual Reality. pp. 143-150. IEEE Press, California, USA (2006).
8. Smit, F.A., Van Rhijn, A.J., Van Liere, R.: A Topology Projection Invariant Optical Tracker. In: Proceedings of the Eurographics Symposium on Virtual Environments. pp. 63-70. Lisbon, Portugal (2006).
9. Suk, T., Flusser, J.: The features for recognition of projectively deformed point sets. In: Proc. of the IEEE International Conference on Image Processing. pp. 348-351. IEEE Press, Washington, USA (1995).
10. Suk, T., Flusser, J.: Point-based projective invariants. Pattern. Recognition(33). pp. 251-261 (2000).
11. Tsionis, V. S., Chandrinou, K.V., Trahanias, P.E.: Landmark-based navigation using projective invariants. In : Proc. of International Conference Intelligent Robots and Systems. pp. 342-347. (1998).
12. Van Liere, R., Mulder, J.D.: Optical Tracking Using Projective Invariant Marker Pattern Properties. In : Proc. of IEEE Virtual Reality. pp. 191-198. IEEE Press, Los Angeles, USA (2003).