Pattern Tracking Problem

Subjects: Computer Science, Artificial Intelligence

Contributor: Nadia Nedjah , Alexandre V. Cardoso , Yuri M. Tavares , Luiza de Macedo Mourelle , Brij Booshan Gupta , Varsha Arya

A pattern is a collection of objects that are similar to each other, arranged in a way that is in contradiction of their natural arrangement. It can also be defined as the opposite of chaos, an entity, loosely defined, which one can assign a specific name. For pattern tracking, tracked objects are usually called patterns. Objects can be defined as something of interest for future analysis. For example, in images, tracking boats at sea, vehicles on the road, aircraft in the air, and people walking on the street can be considered monitoring for a certain purpose and thus tracking.

object tracking template matching swarm intelligence image cross-correlation

1. Introduction

Pattern recognition is one of the most important and active branches of artificial intelligence. It is the science that tries to make machines as smart as human beings in recognizing patterns, among the desired categories, in a simple and reliable way [1][2]. It is also defined as the study of how machines can observe the environment, distinguish various patterns of interest, and make rational decisions. Pattern recognition provides solutions to problems in the most diverse areas such as image analysis, industrial automation, computer vision, biometric identification, remote sensing, voice recognition, face recognition, surveillance, and defense, among many others. Recognizing patterns in images and tracking their positions in videos has been the subject of several studies and has stood out for being a demanding area of image processing and computer vision [3][4].

2. Pattern Detection

Any tracking method requires a mechanism that can identify the object the first time it appears in the video and also in each frame. The most common approaches used for this purpose are based on segmentation, background modeling, point detection, and supervised learning.

Segmentation partitions the image into similar regions to obtain the object of interest. Segmentation algorithms have to balance criteria for good and efficient partitioning. Some examples of algorithms used for segmentation include *graph-cut* [5] and *active contours* [6]. Background modeling builds a representation of the scene and performs object detection based on the deviations observed in each frame [7]. Scene objects are classified by forming a boundary between the background and the foreground. The foreground contains all objects of interest.

Some examples of algorithms used for background modeling include *background subtraction* and *frame differencing*, mixture of Gaussian functions [8], *eigenbackground* [9] and *optical flow* [10].

Point detectors are used to find points of interest in images. These points are called *features* and are highlighted by their distinguishing characteristics in terms of color, texture, geometry, and/or intensity in gradient variation. Object detection is performed by comparing these points. An interesting feature of this approach is its invariance to changes in light and camera position [7]. Some examples of algorithms based on *features* include *Scale Invariant Feature Transform* (SIFT) [11], *invariant point detector* [12], and *Speeded-Up Robust Features* (SURF) [13]. Supervised learning can also be used for object detection. In this case, the task is performed by learning the different points of view of the object, from a set of samples and a supervised learning mechanism. This method usually requires a large collection of samples regarding each class of objects. In addition, the samples must be manually labeled, a time-consuming and tedious task [14]. The selection of the characteristics of the objects in order to differentiate the classes is also an extremely important task for the effectiveness of the method. After learning, the classes are separated, as best as possible by hyper-surfaces in the feature space. Some methodologies using this approach include neural networks [15], *adaptive boosting* [16], and decision tree [17].

It is noteworthy to point out that object detection and tracking are very close and related processes because tracking normally starts with object detection, while repeated object detection in subsequent frames is required to help perform tracking [3].

In order to track an object and analyze its behavior, it is essential to classify it correctly. The classification is directly linked to the characteristics of the object and how it is represented. Approaches to classification are often based on the object’s shape [18], movement [10][19], color [3], and texture [20][21].

3. Tracking Techniques

Tracking can be defined as a problem of approximating the trajectory of an object in a given scene. The main purpose is to find the trajectory of this object by finding its position in each video frame [19]. Basically, tracking techniques can be divided into the following categories: point-based tracking, kernel-based tracking, and silhouette-based tracking. **Figure 1** illustrates the three categories for camera tracking in the known “Cameraman” image. The tasks of detecting the object and matching those of the previous and subsequent frames can be performed together or separately [14].

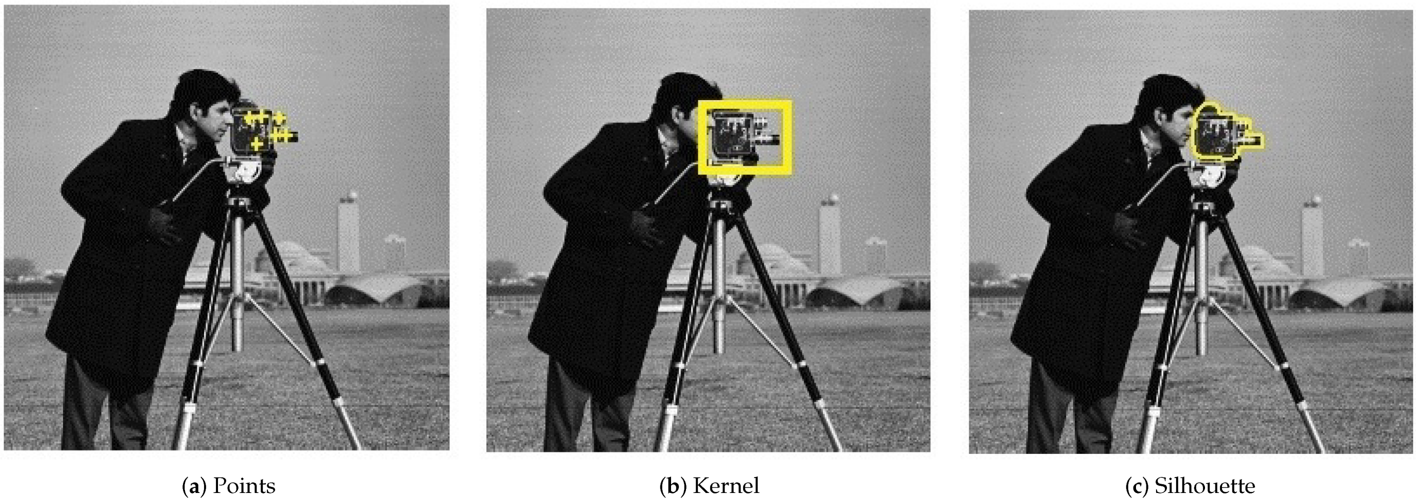


Figure 1. Illustration of existing tracking techniques.

3.1. Point-Based Tracking

For point-based tracking, the objects are represented by dots and the position of the dots in the frame sequence allows the tracking to occur. This approach requires a mechanism to detect the objects in each frame. The Kalman filter, which is a recursive algorithm that provides a computationally efficient means of estimating the system state, is usually used to estimate the position of objects, based on the dynamics of movement along the video. A limitation of the Kalman filter is the assumption that the variables are normally distributed. Thus, when the state variables do not follow a Gaussian distribution, the estimate does not produce good [14] results. This limitation can be overcome with the particle filter, which uses a more flexible state space model. Multiple Hypothesis Tracking (MHT) is another method which is generally used to solve multiple target tracking problems. It is an iterative algorithm based on predefined assumptions about the object trajectories. Each hypothesis is a set of disconnected trajectories. For each hypothesis, the estimate of the target in the next frame is obtained. This estimate is then compared to the current measurement using a distance measurement. This algorithm can deal with occlusions and has the ability to create new trajectories for objects that enter the scene and finalize those related to objects that disappear from the scene.

3.2. Kernel-Based Tracking

In pattern tracking, a kernel refers to an object with a notable region related to its shape and appearance. It can be a rectangular area or an elliptical shape. Objects are tracked by the location after their movements, starting from the embryonic region represented by the kernel, from one frame to the next. These movements are usually represented by affine transformations such as translation, rotation, and scaling. Some of the difficulties of this approach are that kernel does not cover the entire procured object and it includes background contents. The latter is usually mitigated by the layering-based technique, which models the image as a set of layers. One layer is associated with the background and the others are associated with each object in the image. The probability of each pixel belonging to a layer (object) considers the shape characteristics and previous movements of the object. This method is generally useful to track multiple objects.

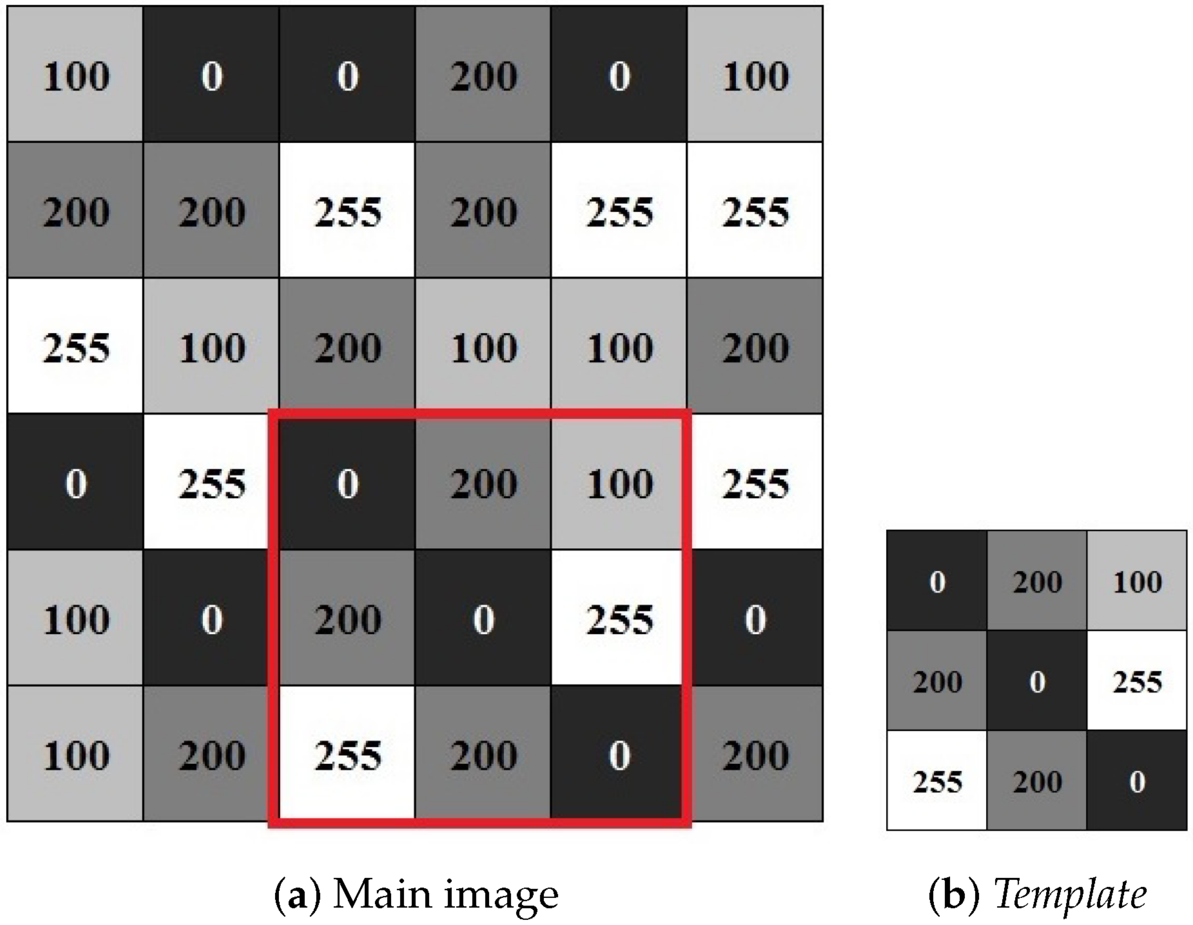
Template matching, also known as model matching, is a brute force method that looks for regions of the image that are similar to a reference image that represents the procured object, called the template. The position of the template in the image is computed from similarity measures, such as sum of absolute differences, sum of squared differences, cross-correlation, and normalized cross-correlation, among others. This method is capable of handling single-image tracking and background changes. A limitation of template matching is the high computational cost associated with brute force. Many researchers, in order to reduce this cost, limit the search area to the neighborhood of the object in the previous frame [14]. The researchers explore this method in this work; it will be further detailed in Section 4.

3.3. Silhouette-Based Tracking

Objects can have complex shapes that cannot be well described with simple geometric shapes [14]. Silhouette-based tracking methods aim to identify the precise shapes of objects in each frame. This approach can be divided into two categories, depending on how the object is tracked: by contours or by shapes. *(i)* Contour matching approaches evolve the initial contour of the object to its new position. It is necessary that part of the object in the previous frame overlaps with the object in the next one. There are many algorithms that extract object contours, such as the one called active contours (or *snakes*), based on the deformation of the initial contour at determined points [22]. The deformation is directed towards the edges of the object by minimizing the snake energy, pushing it towards lines and edges. *(ii)* Shape matching approaches are very similar to template matching. The main difference is that the model represents the exact shape of the object. An example of this type of method is presented in [23]. The algorithm uses the Hausdorff distance to find the location of the object.

4. Template Matching

Template matching (TM) is widely used in image processing to determine the similarity between two entities of the same type (pixels, curves, or shapes). The pattern to be recognized is compared with a previously stored model, taking into account all possible positions. The task basically boils down to finding occurrences of a small image, considered the template, in a sequence of larger images of the frames. **Figure 2** shows two matrices representing two black and white images. The image in **Figure 2**b represents the *template* to be found in the image of **Figure 2**a. In integer-byte representations for black and white images, the larger the value of a pixel, the closer to white it is, and the smaller the value of the pixel, the closer to black it is.



100	0	0	200	0	100
200	200	255	200	255	255
255	100	200	100	100	200
0	255	0	200	100	255
100	0	200	0	255	0
100	200	255	200	0	200

(a) Main image

0	200	100
200	0	255
255	200	0

(b) *Template*

Figure 2. Byte matrices representing the frame and template images in black and white.

The search in the frame is conducted by comparing the template, in each pixel, with pieces of image of the same size. The template slides, pixel by pixel, on the main image until all positions are visited. At each position, a similarity measure is computed and used to compare the images. After calculating all similarity measures, the one with the highest value, above a pre-established threshold, is considered to be the location of the sought template within the frame [24]. This operation is very costly when considering large models and extensive sets of frames [1]. The advantage of template matching is that the template stores several particular characteristics of the object (color, texture, shape, edges, centroid, etc.) which differentiate it from others, allowing greater accuracy and tracking of a specific object within a group of similar ones. Furthermore, object detection is not compromised by choosing how to classify or represent it. The disadvantage is the high computational cost required for the computation of the similarity measure at all image pixels.

To evaluate the degree of similarity of the template along the frame, a range of techniques are used. These include the sum of absolute differences (SAD), sum of squared (SSD), and cross-correlation (CCO). For a given patch, i.e., original image patch A of the same size as the procured template, these indices are computed as shown in Equations (1), (2), and (3), respectively:

$$\text{SAD} \;=\; \sum_{i=1}^{N} |(p_i - a_i)|; \qquad (1)$$

$$\text{SSD} \;=\; \sum_{i=1}^{N} (p_i - a_i)^2; \qquad (2)$$

$$\text{CCO} \;=\; \sum_{i=1}^{N} p_i a_i, \qquad (3)$$

where *N* is the overall number of pixels in the template and patch, pi is the intensity of pixel *i* in the template image, and ai is the intensity of pixel *i* in patch *A*.

Note that in the case of the similarity metrics SAD and SSD, the closer to zero the index is, the more similar the compared images are. However, CCO is sensitive to changes in the amplitude of images' pixels [25]. To overcome this drawback, normalized cross-correlation (NCC) is used. It is noteworthy to point out that, in this work, the researchers use NCC, which is explained in detail hereafter.

The term correlation is widely used in common language to mean some kind of relationship between two things or facts. In the field of signal processing, cross-correlation is obtained by the convolution of one signal by its conjugate. In this work, the term correlation has a more restricted meaning and refers to the similarity measure associated with the normalized cross-correlation between two images. This metric is an improved version of simple cross-correlation CCO. It features a normalizing value in the denominator that provides it invariance to global changes in brightness and results always within the range [−1,1]. The normalized cross-correlation, also known as Pearson's correlation coefficient (PCC) [26], is defined in Equation (4):

$$\text{PCC} = \frac{\sum_{i=1}^{N} (p_i - \overline{p})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{N} (p_i - \overline{p})^2}\sqrt{\sum_{i=1}^{N} (a_i - \overline{a})^2}}, \qquad (4)$$

where pi is pixel intensity *i* in the template image; $\underline{p}$ is the average pixel intensity of the template image; ai is the intensity of pixel *i* in patch *A*; and $\underline{a}$ is the average intensity of the pixels in patch *A*. The template and patch A must be the same size, and the overall number of pixels is *N*.

The PCC can be understood as a dimensionless index with values between −1and +1, inclusive, which reflects the intensity of the degree of the relationship between the two compared images. A coefficient equal to 1 means a perfect positive correlation between the two images. A coefficient equal to −1means a perfect negative correlation between the two images. A coefficient equal to 0 means that the two images do not linearly depend on each other.

The ideal use of the normalized cross-correlation, presented in Equation (4), considers that the appearance of the target remains the same throughout the video [27]. It is noteworthy to mention that any change in target scale or rotation can influence metric values. Additionally, the change in lighting conditions and/or noise, also known as clutter, that is inserted into the environment can cause errors. A possible solution to this problem is to update the template at every frame, allowing adaptive correlation.

References

1. Sharma, P.; Kaur, M. Classification in pattern recognition: A review. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2013, 3, 298–306.
2. Theodoridis, S.; Koutroumbas, K. Pattern Recognition, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2003.
3. Prajapati, D.; Galiyawala, H.J. A Review on Moving Object Detection and Tracking. Int. J. Comput. Appl. 2015, 5, 168–175.
4. Deori, B.; Thounaojam, D.M. A survey on moving object tracking in video. Int. J. Inf. Theory (IJIT) 2014, 3, 31–46.
5. Shi, J.; Malik, J. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 888–905.
6. Caselles, V.; Kimmel, R.; Sapiro, G. Geodesic active contours. In Proceedings of the Fifth International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; pp. 694–699.
7. Chen, F.; Wang, X.; Zhao, Y.; Lv, Y.; Niu, X. Visual object tracking: A survey. Comput. Vis. Image Underst. 2022, 222, 103508.
8. Stauffer, C.; Grimson, W.E.L. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 747–757.

9. Oliver, N.M.; Rosario, B.; Pentland, A.P. A bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 831–843.
10. Ramos, J.; Nedjah, N.; de Macedo Mourelle, L.; Gupta, B.B. Visual data mining for crowd anomaly detection using artificial bacteria colony. Multim. Tools Appl. 2018, 77, 17755–17777.
11. Lowe, D.G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 2004, 60, 91–110.
12. Mikolajczyk, K.; Schmid, C. An affine invariant interest point detector. Eur. Conf. Comput. Vis. (ECCV) 2002, 1, 128–142.
13. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
14. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. ACM Comput. Surv. (CSUR) 2006, 38, 13.
15. Rowley, H.A.; Baluja, S.; Kanade, T. Neural network-based face detection. IEEE Trans. Pattern Anal. Mach. Intell. 1998, 20, 23–38.
16. Viola, P.; Jones, M.J.; Snow, D. Detecting pedestrians using patterns of motion and appearance. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Nice, France, 13–16 October 2003; Volume 2, pp. 734–741.
17. Grewe, L.; Kak, A.C. Interactive learning of a multiple-attribute hash table classifier for fast object recognition. Comput. Vis. Image Underst. 1995, 61, 387–416.
18. Bowyer, K.; Kranenburg, C.; Dougherty, S. Edge detector evaluation using empirical ROC curves. In Proceedings of the Computer Vision Image Understand, Kauai, HI, USA, 8–14 December 2001; IEEE: Piscataway, NJ, USA, 2001; pp. 77–103.
19. Sindhuja, G.; Renuka, D.S.M. A Survey on Detection and Tracking of Objects in Video Sequence. Int. J. Eng. Res. Gen. Sci. 2015, 3, 418–426.
20. Laws, K. Textured Image Segmentation. Ph.D. Thesis, University of Southern California (USC), Marina del Rey, CA, USA, 1980.
21. Greenspan, H.; Belongie, S.; Goodman, R.; Perona, P.; Rakshit, S.; Anderson, C.H. Overcomplete steerable pyramid filters and rotation invariance. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 222–228.
22. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. Int. J. Comput. Vis. 1988, 1, 321–331.

23. Huttenlocher, D.P.; Noh, J.J.; Rucklidge, W.J. Tracking non-rigid objects in complex scenes. In Proceedings of the 1993 (4th) International Conference on Computer Vision, Berlin, Germany, 11–14 May 1993; pp. 93–101.

24. Nixon, M.S.; Aguado, A.S. Feature Extraction and Image Processing, 1st ed.; Academic Press: Cambridge, MA, USA, 2002.

25. Gonzalez, R.C.; Woods, R.E. Processamento de Imagens Digitais; Edgard Blucher: São Paulo, Brazil, 2000.

26. Miranda, A.N. Pearson’s Correlation Coefficient: A More Realistic Threshold for Applications on Autonomous Robotics. Comput. Technol. Appl. 2014, 5, 69–72.

27. Matthews, I.; Ishikawa, T.; Baker, S. The template update problem. IEEE Trans. Pattern Anal. Mach. Intell. 2004, 26, 810–815.

Retrieved from https://encyclopedia.pub/entry/history/show/105432