Very Fast Rotation and Scale Invariant Multiple Template Matching Using Histogram-based Associative Network

LEILA. ESSANNOUNI[1], FEDWA ESSANNOUNI[2]
[1]Ecole supérieure de Technologie de Salé, Université Mohammed V, Rabat, MOROCCO
[2]Faculté des sciences de Rabat, Université Mohammed V, MOROCCO

Abstract— This paper introduces a very fast method for histogram multi-template matching using a fast Multi-layer Perceptron (MLP) based auto associative memory. The method achieves a high-speed improvement by rapidly evaluating the hidden unit outputs. We demonstrate that the hidden unit outputs can be rapidly calculated for any region and scale in a constant time using the integral image technique. Compared to traditional methods like the conventional, distributive, and integral histogram methods, the proposed approach outperforms them with speedups of up to thousands of times. Furthermore, it preserves the desirable properties of rotation and scale invariance.

Keywords—Multi-template Matching, Histogram Matching, Neural Networks, Auto Associative Memory, Integral Histogram, Distributive Histogram.

Received: March 27, 2024. Revised: August 19, 2024. Accepted: May 4, 2025. Published: June 6, 2025.

1 Introduction

Template matching is main task in computer vision [1-5]. Neural networks have been widely utilized for successful template matching and object detection. The utilization of normalized image histograms as features is effective for rotation and scale invariance. Many histogram-based matching techniques have been proposed in the literature [5-10]. However, the computational cost associated with histogram-based neural networks methods presents a significant challenge. The computational complexity arises from the process of calculating local histograms and assessing the neural network outputs at every position. Although integral histogram [11] and distributive histogram [12] methods have been proposed for fast histogram extraction, they still rely on the number of histogram bins. Additionally, the histogram needs to be recomputed for each scale.

To overcome these challenges, we propose a highly fast method for template matching that utilizes the integral representation [13] of auxiliary images, eliminating the need for computing local histograms. We integrate the computation of the network outputs over preprocessed images, allowing for rapid matching at any scale and location.

The paper is organized as follows: the next section presents a novel approach for very fast rotation and scale invariant matching using auto associative memory. Section 3 discusses the computational complexity. Section 4 provides experimental results. Conclusions are drawn in section 5.

2 Extreme fast rotation and scale invariant histogram-based auto associative memory matching method

The proposed approach uses an auto-associative memory. A Multi-Layer Perceptron (MLP) is utilized to implement the auto-associative memory, where the number of input units is equal to the number of output units. During the training process, each input vector is mapped onto itself. The units use a linear activation function. The hidden layer has fewer units than the output layers, enabling the Perceptron to perform dimensionality reduction equivalent to that of PCA. The matching process involves comparing the outputs of the hidden layer of the test image sub-window with the outputs of the set of the given templates.

Traditional histogram-based neural network matching methods have a high computational cost due to the need to process numerous sub-windows in the test image. Additionally, the detection process necessitates extracting local histograms at each possible location. The proposed approach avoids histogram computation and integrates the computation of the hidden layer outputs over constructed auxiliary images.

In the detection phase, a sub-window I of size m x n is extracted from the test image. This sub-window is transformed into a vector of local probabilities that correspond to the local normalized histogram. The resulting vector is then provided as input to the neural network.

Let *Wi* represent the weights between the input units and the hidden layer. The outputs of the hidden neurons, hi, can be calculated as follows:

$$\mathbf{hi}=\sum_{k=1}^{B} \boldsymbol{Wi(k)\, p(k)}. \qquad (1)$$

Here, B represents the number of bins. The computation of hi can be obtained by summing the weights *wi(k)* within sub-window I:

$$\sum_{k=1}^{B} \boldsymbol{Wi(k)\, p(k)} = \frac{1}{|I|}\sum_{x\in I} \boldsymbol{Wi(f(x))}. \qquad (2)$$

The function *f* maps each pixel *x* to its corresponding bin index. By utilizing the integral image technique [13], the outputs of the hidden layer can be computed rapidly. Algorithm 1 provides a summary of the computation steps involved in evaluating the outputs of the hidden layer:

Algorithm 1: Fast evaluation of the outputs of the hidden layer

- Compute the bin index for each feature pixel in the test image.
- Construct an auxiliary image for each neuron in the hidden layer, with the same size as the test image, initializing all pixels to zero.
- Assign the value $w_i(k)$ to each pixel in the auxiliary image that corresponds to the bin index k in the test image.
- Compute the integral image for each auxiliary image.
- Evaluate the outputs of the hidden layer for each neuron and sub window using the integral image technique, utilizing four references located at the corners of the sub window.

3 Complexity analysis

This section presents the computational analysis of the computation of the outputs of the hidden layer at a single scale.
The number of pixels in a test image is given by N^2. n^2 is the pattern size.
Let B denote the number of histogram bins, and q the number of hidden neurons.
The operations considered in this analysis are as follows:

- a ≡ addition/subtraction
- d ≡ division
- f ≡ floor and type conversion
- m ≡ multiplication

The conventional method for the computation of local histograms relies on the brute-force construction of the histogram for each location it requires:

1. Compute the histogram bin for a pixel: 1 division and 1 floor.
2. Increment the corresponding bin: 1 addition.

The total computational cost of the conventional histogram method is:

$$N^2\, n^2\, (d + f + a) \qquad (3)$$

The computation of the outputs of the neural networks hidden layer needs:

$$q\ ((a(B-1)+m\ B)\ N^2) \qquad (4)$$

Consequently, the total operations required for the conventional histogram Neural Networks is:

$$(N)^2\, n^2\, (d + f + a) + q\ ((a(B-1)+m\ B)\ N^2) \qquad (5)$$

The computational cost of the proposed approach, obtaining the values for each pixel in the auxiliary images needs $(d+f)N^2$ operations. The computation of the integral image can be done using N^2 additions, and the number of operations for evaluating the outputs of the hidden layer for each neuron at each sub window is thus only 3 addition operations.
The total cost of the proposed approach is:

$$q\ (\ (4a+d+f)\ N^2) \qquad (6)$$

The computational costs of the integral histogram, the distributive histogram, the conventional method, and the proposed approach are summarized in Table 1. The computational costs of the conventional approach, the integral histogram, and the distributive histogram are proportional to the number of bins. In the case of a high number of bins, such as a color histogram, the computational costs of these methods would be very high. The proposed approach, however, is independent of the number of histogram bins. Overall cases, the proposed approach should outperform the conventional, integral, and distributive methods.

Table1. Computational costs for histogram template matching methods.

Method	Computational cost
Conventional	$N^2 n^2 (d+f+a)$+ q((a (B-1)+ m B) N^2.
Integral method	$N^2 B (d+f+5a)$ + q (a(B-1)+ m B) N^2.
Distributive method	$N^2 (2(d+f) + 2(B+1)a + \frac{n}{N}(d+f+a))$+ q (a (B-1)+ m B) N^2.
Proposed approach	q ((4a+d+f) N^2) .

$N \equiv$ image width/height, q $\equiv$ number of neurons in the hidden layer, $B \equiv$ number of histogram bins, $n\equiv$ template width/height, a, d, f, m are the costs of addition/subtraction, division, floor and multiplication resp.

4 Experimental results

To evaluate the computational improvement of the proposed method compared to the conventional, integral [11], and distributive histogram [12] extraction methods for a single scale, experiments were conducted using Python on a laptop with an Intel i3 processor and 4 GB of RAM. We perform template matching on 100 images taken from the web. The computational cost depends on several parameters, including the image size (N^2), template size (n^2), number of histogram bins (B), and number of neurons in the hidden layer (q).

Table 2 illustrates the speedup ratio of the proposed approach for varying numbers of hidden neurons (B=255, n=30, and N=250). The proposed approach demonstrates superior performance compared to other histogram extraction methods.

Table 3 gives the speed up ratio for different image sizes (B=255, n=30, q=20). As indicated in Table 3, the proposed method exhibits faster processing compared to other histogram-based methods.

Table 4 provides a comparison of the computation time between the proposed approach and other histogram extraction methods for different bin numbers. The proposed approach demonstrates significant improvement in computation time, with speed improvements of up to thousands of times.

The proposed approach is particularly well-suited for multi-scale, and its advantages in processing time become more pronounced when dealing with a high number of bins.

The outputs of the hidden layer for any scale and position can be rapidly calculated using only a few addition operations. This characteristic justifies the rapidity of the proposed method in comparison to other histogram-based methods.

Table 2: Time Improvement of the Proposed Method Compared to Conventional, Integral, and Distributive Methods for Different Numbers of Neurons in the Hidden Layer

q / Method	10	20	30	40	50
Conventional	124	99	95	91	92
Integral	112	94	91	88	89
Distributive	91	84	84	83	85

B=255, n=30, N=250

Table 3: Time Improvement of the Proposed Method Compared to Conventional, Integral, and Distributive Methods for Different Image Sizes

N / Method	100	200	300	400	500
Conventional	109	87	104	98	105
Integral	101	87	97	94	98
Distributive	90	77	87	83	87

B=255, n=30, q=20

Table 4: Time Improvement of the Proposed Method Compared to Conventional, Integral, and Distributive Methods for Different Number of Bins

B / Method	100	255	500	1000	5000
Conventional	56	105	182	344	1577
Integral	39	98	190	382	1854
Distributive	35	87	170	341	1651

N=500, n=30, q=20

5 Conclusion

In this paper, we present a novel rapid matching method, which is rotation and scale invariant by utilizing histograms-based auto associative memory. We demonstrate the rapidity of computing the outputs of the hidden layer using the integral image technique. The proposed approach outperforms conventional, distributive, and integral histogram methods, in terms of speed while producing a similar outputs map. The proposed method significantly improves computation time, particularly when dealing with a large number of bins. It is worth noting that the proposed approach is not limited to color or intensity histograms; it can also be extended to other image features.

References

[1] Annaby, M., & Fouda, Y.: Fast Template Matching and Object Detection Techniques Using φ-Correlation and Binary Circuits. Multimedia Tools and Applications, 83, 6469–6496, (2024).
[2] Wang, H., & Chen, S.: Learning to Detect Objects with a Histogram-of-Differences Descriptor. In 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1183-1187), (2021).
[3] Cui, Z., Qi, W., Liu, Y.: A Fast Image Template Matching Algorithm Based on Normalized Cross Correlation. Journal of Physics: Conference Series. 1693. 012163, (2020).
[4] Luo, C., Yang, W., Huang, P., and Zhou, J.: Overview of Image Matching Based on ORB Algorithm. Journal of Physics: Conference Series. 1237. 032020, (2019).
[5] Peng, X., & Wang, L.: An improved algorithm for template matching based on color histograms. Journal of Ambient Intelligence and Humanized Computing, 12(1), 1073-1081, (2021).
[6] Huang, X., Li, X., & Zhou, Y.: A novel template matching algorithm based on gradient histogram and correlation coefficient. Multimedia Tools and Applications, 79(17), 12195-12211, (2020).
[7] Chen, Z., Xu, T., & Yang, J.: A Fast and Robust Template Matching Algorithm Based on Adaptive Histogram Equalization and Improved SIFT. IEEE Access, 7, 16827-16837, (2019).
[8] Zhang, X., Li, X., & Wang, Y.: A new template matching method based on local binary pattern and histogram intersection similarity. Journal of Electronic Imaging, 27(4), 043017, (2018).
[9] Li, L., Zhang, L., & Wang, Y.: A novel template matching algorithm based on histogram of oriented gradients. Journal of Electronic Imaging, 26(2), 023017, (2017).
[10] Chang, H.W., & Chen, H.T.: "A square-root sampling approach to fast histogram-based search" CVPR, pp. 3043-3049, (2010).
[11] Porikli, F. M. : Integral histogram: 'A fast way to extract histograms in cartesian spaces'. *CVPR*, pp. 829–836, (2005).
[12] Sizintsev, M. K., Derpanis, G., & Hogue, A.: 'Histogram based search: A comparative study'. *CVPR*, (2008).
[13] Viola, P., & Jones, M.: Rapid using a boosted cascade of simple features. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), (2001).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.