Fast Normalized Cross Correlation Based on Adaptive Multilevel Winner Update

Shou-Der Wei and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
{greco, lai}@cs.nthu.edu.tw

Abstract. In this paper we propose a fast normalized cross correlation (NCC) algorithm for pattern matching based on combining adaptive multilevel partition with the winner update scheme. This winner update scheme is applied in conjunction with an upper bound for the cross correlation derived from Cauchy-Schwarz inequality. To apply the winner update scheme, we partition the summation of cross correlation into different levels with the partition order determined by the gradient energies of the partitioned regions in the template. Thus, this winner update scheme can be employed to skip the unnecessary calculation. Experimental results show the proposed algorithm is very efficient for image matching under different lighting conditions.

Keywords: pattern matching, normalized cross correlation, winner update strategy, multi-level successive elimination, fast algorithms.

1 Introduction

The pattern matching problem can be formulated as follows: Given a source image I and a template image T of size MxN, the pattern matching problem is to find the best match of template T from the source image I with minimal distortion or maximal correlation. Several previous works on pattern patching have been proposed [1][2][3][4] based on the measure of sum of absolute differences (SAD) or sum of squared differences (SSD). The NCC measure is more robust than SAD and SSD under uniform illumination changes, so it has been widely used in image matching, object recognition and industrial inspection.

The NCC between two images I and T with displacement (x,y) is defined by

$$NCC(x,y)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i+x,j+y)\cdot T(i,j)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i+x,j+y)^2}\cdot\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} T(i,j)^2}} \tag{1}$$

The sum table scheme [6] was proposed to reduce the computation in the denominator. In addition, Cauchy-Schwarz inequality has been employed to reduce the computation in the numerator [5].

H.H.S. Ip et al. (Eds.): PCM 2007, LNCS 4810, pp. 413–416, 2007.

2 The Proposed Fast NCC-Based Image Matching Algorithm

In this paper, we propose a fast algorithm for NCC-based image matching by applying the adaptive block partition in the Cauchy-Schwarz inequality with the winner update scheme. As shown in equation (2), we can divide a block into many subblocks and calculate the summation of each block's upper bound to obtain tighter bound by Cauchy-Schwarz inequality for the cross correlation (CC). Following the uniform partitioning scheme of MSEA [1], we have many upper bounds for different partitioning levels and the relation between the upper bounds for different levels are given in equation (3) and (4). At the final level, the upper bound is equal to the cross correlation. In contrast to the uniform partition, we can determine the partition order by the sum of gradient magnitudes for the subblocks in the template. The block with the current largest sum of gradient magnitudes is divided into 2x2 sub-blocks for consideration of further partitioning. The adaptive block partitioning algorithm is given in Algorithm 1 and an example of adaptive block partition is depicted in Fig. 1.

$$\sqrt{\sum_{i=1}^{N} a_i^2} \cdot \sqrt{\sum_{i=1}^{N} b_i^2} \geq \sqrt{\sum_{i=1}^{k} a_i^2} \cdot \sqrt{\sum_{i=1}^{k} b_i^2} + \sqrt{\sum_{i=k+1}^{N} a_i^2} \cdot \sqrt{\sum_{i=k+1}^{N} b_i^2} \geq \sum_{i=1}^{N} a_i \cdot b_i \tag{2}$$

$$UB_l(x,y) = \sum_{a \in AllSubblock} \left(\sqrt{\sum_{i \in AllPixels} I_{a_i}(x,y)^2} \cdot \sqrt{\sum_{i \in AllPixels} T_{a_i}^2} \right) \tag{3}$$

$$UB_0 \geq UB_1 \geq \cdots \geq UB_{L=\log_2 N} = CC \tag{4}$$

$$BV(x,y) = \frac{UB(x,y)}{|I(x,y)| \cdot |T|} \tag{5}$$

Algorithm 1: Algorithm for determining adaptive block partitioning order
Push the largest block into the queue Repeat until the queue is empty 1. Select the block with largest sum of gradient magnitudes from the queue. 2. Divide the selected block into four sub-blocks and calculate their sum of gradient magnitudes. 3. Check the four sub-blocks and push each sub-block into the queue if its sum of gradient magnitudes is greater than a given threshold T.

The proposed algorithm includes the adaptive block partitioning algorithm combined with the winner update scheme [4] for fast search of the location with maximal NCC. With the block partitioning, we have the relation of upper bounds for different levels as $UB_0 \geq UB_1 \geq \cdots \geq UB_{\max L} \geq CC$. We can calculate the boundary values from equation (5) and have the relation of boundary values of different levels as $BV_0 \geq BV_1 \geq \cdots \geq BV_{\max L} \geq NCC$. The BV_l value is closer to NCC as the level increases. Based on the above relation of BV_l, we can apply the winner update scheme to it. At first, we calculated BV_0 for all candidates, and then at each iteration we choose the candidate with the current maximal BV_l as the winner to update its level and recalculate the BV_{l+1}. This procedure is repeated until the chosen winner reaches the maximal level, thus its BV_{maxL} is the same as the maximal NCC value. This algorithm of applying winner update scheme with adaptive block partition for NCC is summarized in Algorithm 2. Similar to the winner update method in [4], we also use a hash table to find the temporary winner.

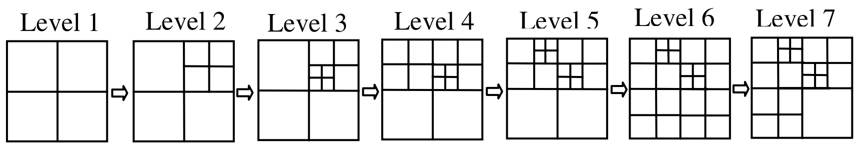

Fig. 1. An example of the adaptive block partitioning order

Algorithm 2: The proposed fast NCC pattern matching algorithm
Step 1: Determine the elimination order.
Step 2: Calculate the norm of template |T|
Step 3: Calculate the BV_0 of all candidate and initial the Hash Table
Repeat
Step 5: Select the candidate with maximal BV in hash table as the winner
Step 6: Update the level and BV of the winner
1. Retrieve the next partitioning next level l
2. Calculate the UB_l for level l. Compute $BV_l = UB_l / (\lvert T \rvert \lvert C(x,y) \rvert)$
3. Push candidate into Hash Table.
Until the winner reaches the maximal level.

3 Experimental Results

To compare the efficiency of the proposed algorithm, termed WUS_NCC, we also implemented the multi-level SEA with fixed partitioning scheme and the results are termed as MSEA_NCC. In our experiment, we used the sailboat image of size 512-by-512 as the source image and six template images of size 64x64 inside the sailboat image as shown in Figure 2. The experimental results of the proposed algorithms and the original NCC are shown in Table 1. All these three algorithms used the sum table to reduce the computation of denominator in NCC. For efficiently calculating the bound of the numerator, we also used the approach of BSPA [2] to build two block square sum pyramids for intensity image and the gradient map, respectively. The execution time shown in the table includes the time of memory allocation for sum table and pyramids, and building sum table, pyramids and the gradient map. These experimental results show the significant improvement in the efficiency of the proposed fast NCC-based pattern matching algorithm.

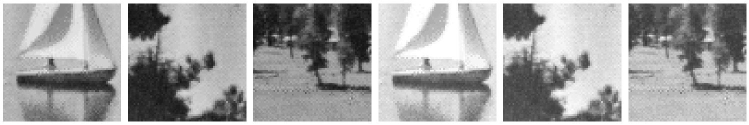
(a) (b) (c) (d) (e) (f)

Fig. 2. (a), (b), (c): The template images (64x64). (d), (e), (f): their brighter versions

Table 1. The execution time (in msec) of applying traditional NCC, MSEA_NCC and WUS_NCC on six templates shown in Figure2(a)~(f), Note that the NCC algorithm used the sum table to reduce the computation in the denominator of NCC.

Unit: msec	T(a)	T(b)	T(c)	T(d)	T(e)	T(f)
NCC	3235	3235	3235	3235	3235	3235
MSEA_NCC	281	203	656	234	219	563
WUS_NCC	109	94	94	94	94	94

4 Conclusion

In this paper, we proposed a very efficient algorithm for fast pattern matching in an image based on normalized cross correlation. To achieve very efficient computation, we partition the summation of cross correlation into different levels and apply the winner update scheme to find the location with maximal NCC. The block partition order is adaptively determined by the sum of gradient magnitudes for each partitioned regions in the template. Our experimental results show the proposed algorithm is very efficient and robust for pattern matching under linear illumination change.

References

1. Gao, X.Q., Duanmu, C.J., Zou, C.R.: A multilevel successive elimination algorithm for block matching motion estimation. IEEE Trans. Image Processing 9(3), 501–504 (2000)
2. Lee, C.-H., Chen, L.-H.: A fast motion estimation algorithm based on the block sum pyramid. IEEE Trans. Image Processing 6(11), 1587–1591 (1997)
3. Gharavi-Alkhansari, M.: A fast globally optimal algorithm for template matching using low-resolution pruning. IEEE Trans. on Image Processing 10(4), 526–533 (2001)
4. Chen, Y.S., Huang, Y.P., Fuh, C.S.: A fast block matching algorithm based on the winner-update strategy. IEEE Trans. Image Processing 10(8), 1212–1222 (2001)
5. Di Stefano, L., Mattoccia, S.: A sufficient condition based on the Cauchy-Schwarz inequality for efficient template matching. In: Proc. Intern. Conf. Image Processing (2003)
6. Lewis, J.P.: Fast template matching. Vision Interface, 120–123 (1995)