J.P. Lewis, Fast Template Matching, Vision Interface 95, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, 1995, p. 120-123.
(Also see the expanded and corrected version "Fast Normalized Cross-Correlation").

Fast Template Matching

J. P. Lewis
Industrial Light & Magic
P.O. Box 2459 San Rafael CA 94912 U.S.A.

Abstract

Although it is well known that cross correlation can be efficiently implemented in the transform domain, the normalized form of cross correlation preferred for template matching applications does not have a simple frequency domain expression. Normalized cross correlation is usually computed in the spatial domain for this reason. This short paper shows that unnormalized cross correlation can be efficiently normalized using precomputed tables containing the integral of the image and image^2 over the search window.

1 Template Matching by Cross Correlation

Correlation is an important tool in image processing, pattern recognition, and other fields. The correlation between two signals (cross correlation) is a standard approach to feature detection [1, 2] as well as a building block for more sophisticated recognition techniques (e.g. [3]). Textbook presentations of correlation commonly mention the convolution theorem and the attendant possibility of efficiently computing correlation in the frequency domain via the fast Fourier transform. Unfortunately the normalized form of correlation (correlation coefficient) preferred in many applications does not have a correspondingly simple and efficient frequency domain expression, and spatial domain implementation is recommended instead (e.g. [2], p. 585; also see e.g. [4] sections 13.2 and 14.5). This paper shows that the unnormalized correlation can be efficiently normalized using using precomputed tables of the integral of the signal and signal^2, i.e., *summed-area tables* [5].

Template matching techniques [1] attempt to answer some variation of the following question: Does the image contain a specified view of some feature, and if so, where? The use of cross correlation for template matching is motivated by the distance measure (squared Euclidean distance)

$$d^2_{f,t}(u,v) = \sum_{x,y}[f(x,y) - t(x-u,y-v)]^2$$

(the sum is over x, y under the window containing the feature positioned at u, v). In the expansion of d^2

$$d^2_{f,t}(u,v) = \sum_{x,y}[f^2(x,y) \quad - \quad 2f(x,y)t(x-u,y-v) \\ + \quad t^2(x-u,y-v)]$$

the term $\sum t^2(x-u,y-v)$ is constant. If the term $\sum f^2(x,y)$ is approximately constant then the remaining *cross correlation* term

$$c(u,v) = \sum_{x,y} f(x,y)t(x-u,y-v)$$

is a measure of the similarity between the image and the feature.

2 Normalized Cross Correlation

If the image energy $\sum f^2(x,y)$ is not constant however, feature matching by cross correlation can fail. For example, the correlation between the template and an exactly matching region in the image may be less than the correlation between the template and a bright spot. Another drawback of cross correlation is that the range of $c(u,v)$ is dependent on both the size of the template and the template and image amplitudes.

Variation in the image energy under the template can be reduced by high-pass filtering the image before cross correlation. In a transform domain implementation the filtering can be conveniently added to the frequency domain processing, but selection of the cutoff frequency is problematic – a low cutoff may leave significant image energy variations, whereas a high cutoff may remove information useful to the match.

Normalized cross correlation overcomes these difficulties by normalizing the image and template vectors to

unit length, yielding a cosine-like correlation coefficient

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}][t(x-u,y-v) - \bar{t}]}{\left\{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2[t(x-u,y-v) - \bar{t}]^2\right\}^{0.5}} \quad (1)$$

where $\bar{t}$ is the mean is the mean of the template and $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the template.

3 Computation

Consider the numerator in (1) and assume that we have images $f'(x,y) \equiv f(x,y) - \bar{f}_{u,v}$ and $t'(x,y) \equiv t(x,y) - \bar{t}$ in which the mean value has already been removed:

$$\overset{\text{num}}{\gamma}(u,v) = \sum_{x,y} f'(x,y)t'(x-u,y-v) \quad (2)$$

For a search window of size M^2 and a template of size N^2 (2) requires approximately $N^2(M-N+1)^2$ additions and $N^2(M-N+1)^2$ multiplications.

Eq. (2) is a convolution of the image with the reversed template $t'(-x,-y)$ and can be computed by

$$\mathcal{F}^{-1}\{\mathcal{F}(f')\mathcal{F}^*(t')\} \quad (3)$$

where $\mathcal{F}$ is the Fourier transform. The complex conjugate accomplishes reversal of the template via the Fourier transform property $\mathcal{F}f^*(-x) = F^*(\omega)$.

Implementations of the fast Fourier transform (FFT) algorithm generally require that f' and t' be extended with zeros to a common power of two. The complexity of the transform computation (2) is then $12M^2log_2M$ real multiplications and $12M^2log_2M$ real additions [6]. When M is much larger than N the complexity of the direct 'spatial' computation (2) is approximately N^2M^2 multiplications/additions, and the direct method is faster than the transform method. The transform method becomes relatively more efficient as N approaches M and with larger M, N.

4 Normalized Cross Correlation in the Transform Domain

Examining again the numerator of (1), we note that the mean of the template can be precomputed, leaving

$$\begin{aligned}\overset{\text{num}}{\gamma}(u,v) &= \sum f(x,y)t'(x-u,y-v) \\ &- \bar{f}_{u,v}\sum t'(x-u,y-v)\end{aligned}$$

Since t' has zero mean and thus zero sum the term $\bar{f}_{u,v}\sum t'(x-u,y-v)$ is also zero, so the numerator of the normalized cross correlation can be computed using (3).

Examining the denominator of (1), the length of the template vector can be precomputed in approximately $3N^2$ operations (small compared to the cost of the cross correlation), and in fact the template can be pre-normalized to length one.

The problematic quantities are those in the expression $\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2$. The image mean and local energy must be computed at each u, v, i.e. at $(M-N+1)^2$ locations, resulting in almost $3N^2(M-N+1)^2$ operations (counting add, subtract, multiply as one operation each). This computation is more than is required for the direct computation of (2) and it may considerably outweight the computation indicated by (3) when the transform method is applicable. A more efficient means of computing the image mean and energy under the template is desired.

These quantities can be efficiently computed from summed-area tables containing the integral (running sum) of the image and image square over the search area, i.e.,

$$s(u,v) = f(u,v) + s(u-1,v) + s(u,v-1) - s(u-1,v-1)$$

$$\begin{aligned}s^2(u,v) = f^2(u,v) &+ s^2(u-1,v) \\ &+ s^2(u,v-1) - s^2(u-1,v-1)\end{aligned}$$

with $s(u,v) = s^2(u,v) = 0$ when either $u, v < 0$. The energy of the image under the template positioned at u, v is then

$$\begin{aligned}e_f(u,v) = &\ s^2(u+N-1,v+N-1) \\ - &\ s^2(u-1,v+N-1) \\ - &\ s^2(u+N-1,v-1) \\ + &\ s^2(u-1,v-1)\end{aligned}$$

and similarly for the image sum under the template. This technique of computing a definite sum from a precomputed running sum was introduced in [5] to rapidly low-pass filter texture images.

The problematic quantity $\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2$ can now be computed with very few operations since it expands into an expression involving only the image sum and sum squared under the template. The construction of the tables requires approximately $3M^2$ operations, which is less than the cost of computing the numerator by (3) and considerably less than the $3N^2(M-N+1)^2$ required to compute $\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2$ at each u, v.

5 Application

The integration of synthetic and processed images into special effects sequences often requires accurate tracking of sequence movement and features. The algorithm described in this paper was used for this purpose in the recent movie *Forest Gump*. The special effect sequences in that movie included the replacement of various moving elements and the addition of a contemporary actor into historical film and video sequences. Manually picked features from one frame of a sequence were automatically tracked over the remaining frames; this information was used as the basis for further processing.

The relative performance of feature tracking by the transform domain algorithm is a two-dimensional function of the problem size (search window size) and the ratio of the template size to search window size. Relative performance increases along the problem size axis, with an additional ripple reflecting the relation between the search window size and the bounding power of two (Fig. 1). The property that the relative performance is greater on larger problems is desirable. Table 1 illustrates the performance obtained in practice.

Note that while a small (e.g. 10^2) template size would suffice in an ideal digital image, in practice larger feature sizes are more immune to imaging noise such as the effects of film grain and motion blur. Due to the high digital resolution required to represent video and film, a small movement across frames may correspond to a distance of many pixels. Also the selected features are of course constrained to the available features in the image; distinct "features" are not always available at preferred scales and locations. As a result of these considerations search windows of 50^2 and larger are often employed.

6 Conclusion

The transform domain normalized cross correlation algorithm is well suited to special effects feature tracking. Manual tracking of multiple features over several minutes of frames was not feasible, and small errors in manual tracking would have prevented the seamless integration of new image elements.

The approach presented here should also be compared to spatial multigrid convolution approaches. We note that while these approaches would work for some examples, we have encountered other cases in which the available features (e.g. a configuration of small spots on a wall or floor) have very little low-frequency content and the low-frequency search would not be robust.

References

[1] R.O. Duda and P.E.Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (third edition), Reading, Massachusetts: Addison-Wesley, 1992.

[3] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993.

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C, Second Edition*, Cambridge: Cambridge University Press, 1992.

[5] F. Crow, "Summed-Area Tables for Texture Mapping", *Computer Graphics*, vol 18, No. 3, pp. 207-212, 1984.

[6] A. Goshtasby, S.H. Gage, and J.F. Bartholic, "A Two-Stage Cross Correlation Approach to Template Matching", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 374-378, 1984.

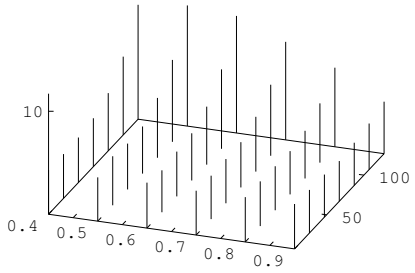

Figure 1: Measured relative performance of transform domain versus spatial domain normalized cross correlation as a function of the search window size (depth axis) and the ratio of the template size to search window size.

search window(s)	length	direct	transform
168×86	896 frames	15 hours	1.7 hours
$115 \times 200, 150 \times 150$	490 frames	14.3 hours	57 minutes

Table 1. Two tracking sequences from *Forest Gump* were re-timed using both transform and direct methods using identical templates and search windows on an ≈ 60 SPECmark workstation. These times include a 16^2 sub-pixel search at the location of the best whole-pixel match. The sub-pixel search was computed using Eq. (1) (direct approach) in all cases.