

Kernel-Based Non-linear Template Matching

Barend J. van Wyk, Michaël A. van Wyk, and Guillaume Noel

French South-African Technical Institute in Electronics,
Tshwane University of Technology
Staatsartillerie Road, Pretoria, South Africa
{ben.van.wyk,guillaume.noel,mavw}@fsatie.ac.za

Abstract. A new non-linear minimum norm template matching technique is introduced. Similar to the theory of Support Vector Machines the proposed framework is also based on Reproducing Kernel Hilbert Space principles. Promising results when applied to aerial image matching are reported and future work is highlighted.

1 Introduction

In this paper the problem of finding the location of a known reference image, or template, in a larger input image is addressed. Finding the location of the reference image can be done by searching through the input image using the normalized cross correlation as a similarity measure [1]. As the normalized cross correlation technique won't work when our reference image is rotated or scaled, several modifications were proposed. The Fourier and Mellin transforms can be combined [2], multiple templates can be used, or the reference and sub-images can be described in terms of invariant moments and then the correlations involving these moments can be used as a similarity measure [3]. As shown by Ueno-hara and Kanade [4] the multiple template approach can be made more efficient by implementing a dual decomposition using the Fourier and Karhunen-Loève transforms. Ben-Arie and Rao [5] [6], on the other hand proposed non-orthogonal image expansions where the search area is represented by basis functions that are effectively the template translated to different positions.

The technique proposed in this paper is also based on the idea of having multiple templates, but differs from other popular methods in the way the templates are selected. Another differentiating factor is that it is non-linear, with the linear case as a special instance of the proposed framework.

2 Non-linear Template Matching Framework

Similar to the theory of Support Vector Machines (SVMs) our framework is also based on Reproducing Kernel Hilbert Space (RKHS) principles, in particular on the idea of an RKHS interpolator. For a more general discussion on RKHS interpolators, the reader is referred to [7], [8], [9], [10] and [11]. The following theorem, stated by Zyla and De Figueiredo [12] for Bochner spaces, but adapted here for our purposes, is the core of our methodology:

Theorem 1. [12] *Given that an input-output map \tilde{F} to be identified belongs to \mathcal{H}_n , an RKHS, and assuming that we are provided with a set of test input-output pairs*

$$\{(\mathbf{x}_i \in \mathbb{R}^N, y_i)\}_{i=1}^m \quad (1)$$

where \mathbf{x}_i , $i = 1, \dots, m$, are linearly independent elements of \mathbb{R}^N , the problem has a unique minimum norm solution expressed by

$$\tilde{F}(\mathbf{x}) = \sum_{i=1}^m C_i K(\mathbf{x}_i, \mathbf{x}) \quad (2)$$

where $K(\mathbf{x}_i, \cdot)$ is a reproducing kernel of the space \mathcal{H}_n . The coefficients C_i are given by the expression

$$\mathbf{C} = \mathbf{G}^{-1} \mathbf{y} \quad (3)$$

where

$$\begin{aligned} \mathbf{C} &:= (C_1, \dots, C_m)^T, \\ \mathbf{y} &:= (y_1, \dots, y_m)^T \end{aligned}$$

and the Gram matrix, \mathbf{G} , is given by

$$\mathbf{G} := (G_{ij})$$

where

$$G_{ij} := K(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \dots, m.$$

Theorem 1 will now be used for the derivation of our template matching scheme. To apply theorem 1 to template matching five factors need to be considered namely, defining the test input-output pairs, choosing a suitable kernel, calculating the interpolator coefficients, deriving a minimum norm template and implementing the matching process. From theorem 1 it is clear that once the test input-output pairs are defined and an appropriate kernel chosen, the interpolator constraints are obtained by simply inverting a Gram matrix. Refer to Luenberger [13] for conditions under which the Gram matrix is invertible. If the Gram matrix is found to be ill-conditioned or badly scaled one can resort to the pseudo-inverse. In the approach followed by De Figueiredo and Zyla [7] [12], every reproducing kernel $K(\mathbf{x}_i, \cdot)$ is associated with a specific norm on \mathcal{H}_n . It is important to note that because of this relationship *only* knowledge of the reproducing kernel is required when applying theorem 1. Refer to [14] to see how theorem 1 relates to the well-known representer's theorem.

The definition of the test input-output pairs is discussed in section 2.1, examples of suitable kernels are given in section 2.2, the derivation of minimum norm templates are discussed in section 2.3 and the matching process is detailed in section 2.4.

In short we will infer a *Minimum Norm Template* (MNT) based on Eq. 2 using k *Desirable Image Templates* (DITs) and $m - k$ *Undesirable Image Templates* (UITs) where $m \geq k$, as input-output pairs. For the application considered in

this paper the DITs will be rotated and scaled versions of the region of interest as shown in figures 1 to 4, i.e. instances of the entity we want to find in a complex image. The UITs will be instances of objects or backgrounds we don't want to recognize as our region of interest such as undesirable complex backgrounds as shown in figure 5. For simplicity it will be assumed that the DITs and UITs are square, have equal dimension and are represented by $\mathbf{X}_i \in \mathbb{R}^{\overline{N} \times \overline{N}}$.

2.1 Test Input-Output Pairs $\{(\mathbf{x}_i \in \mathbb{R}^N, \mathbf{y}_i)\}_{i=1}^m$

For the DITs (i.e. $i \leq k$) the y_i values in Eq. 1 are normally chosen equal to some positive value, say γ . The rest of the y_i values for the UITs are normally set to α , where α is zero or $-\gamma$. Each \mathbf{x}_i is simply set equal to $\text{vec}(\mathbf{X}_i) \in \mathbb{R}^N$ where $\text{vec}(\cdot)$ is the matrix vectorization operator and $N = \overline{N}^2$.

2.2 Examples of Reproducing Kernels $K(\mathbf{x}, \mathbf{z})$

Although the theory is general enough to allow other reproducing kernels we will only focus on three types of kernels, namely the linear kernel

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z}, \tag{4}$$

the polynomial kernel,

$$K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^T \mathbf{z})^d, \quad d \geq 1, \tag{5}$$

and the polynomial kernel without cross terms

$$K(\mathbf{x}, \mathbf{z}) = \left(1 + \sum_{\beta=1}^d \left(x_1^\beta z_1^\beta + x_2^\beta z_2^\beta, \dots, x_N^\beta z_N^\beta \right) \right), \quad d \geq 1. \tag{6}$$

2.3 The Minimum Norm Template

Once the interpolator coefficients are obtained an MNT can be inferred. When using the linear kernel it is easy to show that the MNT has the form

$$\tilde{\mathbf{x}} = \sum_{i=1}^m C_i \mathbf{x}_i \tag{7}$$

and that $K(\tilde{\mathbf{x}}, \cdot)$ will satisfy

$$K(\tilde{\mathbf{x}}, \mathbf{x}_i) = \begin{cases} \gamma & \text{for } i = 1, \dots, k \\ \alpha & \text{for } i > k \end{cases}.$$

When the DITs and UITs are linearly separable, the suitability of using $\text{devec}(\tilde{\mathbf{x}}) \in \mathbb{R}^{\overline{N} \times \overline{N}}$ as an object template is obvious. Here devec denotes the inverse of vec .



Fig. 1. Original image.



Fig. 2. Example of rotated (-24 degrees) and scaled (factor 20) input image.



Fig. 3. Image template: Region searched for in input image.

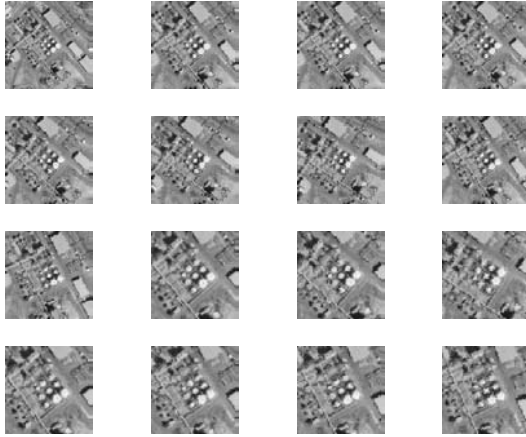


Fig. 4. Example of 16 Desirable Image Templates (DITs).

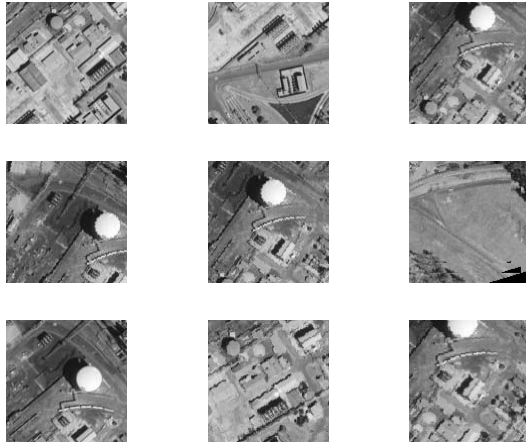


Fig. 5. Example of 9 Undesirable Image Templates (UITs).

When our input training pairs are not linearly separable we will resort to polynomial kernels. First consider the kernel given by Eq. 5. For simplicity we will consider the case where $d = 2$. By using similar arguments as for the linear kernel case it can be shown that

$$\tilde{\mathbf{x}} = \sum_{i=1}^m C_i \bar{\mathbf{x}}_i, \quad (8)$$

where $\bar{\mathbf{x}}_i = [[1 \ \mathbf{x}_i^T] \otimes [1 \ \mathbf{x}_i^T]]^T$ and \otimes denotes the *Kronecker Tensor Product*. Similar to the linear case,

$$\tilde{\mathbf{x}}^T \bar{\mathbf{x}}_i = \begin{cases} \gamma & \text{for } i = 1, \dots, k \\ \alpha & \text{for } i > k \end{cases}.$$

These results can be readily extended for cases where $d > 2$. When $d = 3$ for example we have

$$\tilde{\mathbf{x}} = \sum_{i=1}^m C_i \left[[1 \ \mathbf{x}_i^T] \otimes [1 \ \mathbf{x}_i^T] \otimes [1 \ \mathbf{x}_i^T] \right]^T.$$

The polynomial kernel without cross terms given by Eq. 6 can be seen as a compromise between the linear kernel and the polynomial kernel given by Eq. 5. The *Minimum Norm Template* (MNT) for this case can be expressed as a concatenated vector given by

$$\tilde{\mathbf{x}} = \left[\sum_{i=1}^m C_i, \sum_{i=1}^m C_i \mathbf{x}_i^1, \sum_{i=1}^m C_i \mathbf{x}_i^2, \dots, \sum_{i=1}^m C_i \mathbf{x}_i^d \right]^T \tag{9}$$

where \mathbf{x}_i^d denotes that every element of \mathbf{x}_i^T is raised to the power d . Once again

$$\tilde{\mathbf{x}}^T \bar{\mathbf{x}}_i = \begin{cases} \gamma & \text{for } i = 1, \dots, k \\ \alpha & \text{for } i > k \end{cases},$$

where $\bar{\mathbf{x}}_i := [1 \ \mathbf{x}_i^1 \ \mathbf{x}_i^2 \dots \mathbf{x}_i^d]$.

2.4 The Matching Process

In summary the kernel-based template matching process involves the following steps:

1. Calculate $\tilde{\mathbf{x}}$, the MNT, offline using the DITs and the UITs.
2. Once the MNT $\tilde{\mathbf{x}}$ has been obtained, $\tilde{\mathbf{x}}^T \bar{\mathbf{x}}_{kl}$ which serves as our similarity measure, is calculated for all (or selected) portions over an area of interest in the input image. When for example the polynomial kernel without cross terms is used, then from the previous section we have $\bar{\mathbf{x}}_{kl} := [1 \ \text{vec}\mathbf{X}_{kl}^1 \ \text{vec}\mathbf{X}_{kl}^2 \dots \text{vec}\mathbf{X}_{kl}^d]$. Here $\mathbf{X}_{kl} \in \mathbb{R}^{\overline{N} \times \overline{N}}$, an $\overline{N} \times \overline{N}$ region centered at position (k, l) in the $K \times L$ input image and $\text{vec}\mathbf{X}^d$ denotes that every element of $(\text{vec}\mathbf{X})^T$ is raised to the power d . It is assumed that $K, L > \overline{N}$.
3. Position (k, l) in the input image where the value for $\tilde{\mathbf{x}}^T \bar{\mathbf{x}}_{kl}$ is a maximum is taken as the location of the object or region to be identified.

The template matching method presented here differs from conventional SVM classification strategies mainly in two aspects: 1) Kernel evaluations are only performed to calculate the MNT. Once the MNT is calculated no further kernel evaluations are required during the execution of the matching process. 2) The interpolator coefficients used to construct the MNT are obtained by simply inverting a Gram matrix. Complex optimization methods are not required.

2.5 Efficient Implementation

If the linear kernel is used the matching process is equivalent to 2D filtering, where the 2D filter coefficients are given by $\tilde{\mathbf{X}} = \text{devec}(\tilde{\mathbf{x}})$. By using the fact that discrete convolution in the spatial domain is equivalent to point-wise multiplication of discrete Fourier spectra in the frequency domain, the cost of searching for a match in an area of interest can be reduced by calculating two *Fast Fourier Transforms* (FFTs), performing a point-wise multiplication, and calculating the Inverse FFT (IFFT) of the result. It is assumed that sufficient zero padding is added to implement linear and not circular convolution. When using the polynomial kernel given by Eq. 5, a single-pass FFT technique is not feasible in general as 2D non-linear filtering is performed. However even when using the polynomial kernel,

$$\left(\sum_{i=1}^m C_i (1 + \mathbf{x}_i^T \mathbf{x}_{k,l})^d \right),$$

sequential or parallel FFT processing can still be used to process the inner terms $\mathbf{x}_i^T \mathbf{x}_{k,l}$. When m , that is dependent on the number of DITs and UITs, is large, say > 10 which will usually be the case, all \mathbf{x}_i associated with a near zero C_i can be ignored to save complexity. Alternatively a more optimal SVM approach can be followed to only obtain those \mathbf{x}_i labelled as support vectors.

Most of the computational difficulties associated with the kernel given by Eq. 5 can be overcome by implementing the kernel given by Eq. 6. What is important to note in this case is that fast frequency domain algorithms developed for the linear template matching case can be adapted for use with the kernel given by Eq.6 by calculating d sub-templates and executing the linear algorithm d times. Since d is normally chosen as a small value, say 3 or 4, the computational load will still be manageable for most cases. However, performance versus complexity will in the end be dictated by the application.

3 Simulation Results

To test our kernel-based approach, figure 1 (400×400 pixels) was used as a reference image. The objective was to locate the (121×121 pixel) region depicted by figure 3, in figures such as figure 2. Here figure 2 is a rotated and scaled version of figure 3.

Thirty-three DITs were constructed by first rotating the reference template, i.e. figure 3, through -20 -16 -8 -4 , 4 , 8 , 12 , 16 and 20 degrees to obtain 9 additional templates. Each rotated template as well as the original template was then scaled by factors 26 and 52 to produce another additional 20 templates. A scale factor of 26 implies that a 26 pixel wide border was removed from the original template after which the cropped template was again made the same size as the original template using bilinear interpolation. Sixteen of the 30 DITs are shown in figure 4.

Sixteen UITs were obtained by extracting 121×121 pixel regions around the neighborhood of the zero rotated and scaled DIT and elsewhere from the

reference image. Nine of the UITs are depicted in figure 5. The DITs and UITs were then used as described in section 2.4.

To test the performance of the different kernels, a series of input images was generated by first rotating the reference image by -21 , -15 , -9 , -3 , 3 , 9 , 15 and 21 degrees, and then scaling by factors 0 , 8 , 16 , 24 , 32 , 40 and 48 . For each input image the Euclidean distance (measured in pixels) from the midpoint of region with highest match, to the midpoint of the true location of the region searched for, was calculated. Note that the rotation and scale factors used to produce the input images *do not correspond* the rotation and scale factors used to generate the DITs. The results of the experiment for the *linear* kernel is reported in table 1. Note that ordinary normalized cross correlation is only of value when no rotation or scaling is involved. Both the polynomial kernel given by Eq. 5 with $d = 2$ and Eq. 6 with $d = 3$ yielded improved results. If a distance of more than 10 pixels is taken as a mismatch, then the results obtained using the polynomial kernel without cross terms reported in table 2 show that there is much to gain by using non-linear kernels.

Table 1. Euclidean distance measured in pixels from midpoint of region with highest match to midpoint of true region using the linear kernel.

Rotation and Scale	-21^0	-15^0	-9^0	-3^0	3^0	9^0	15^0	21^0
0	117	0	0	0	0	11	115	124
8	114	1	1	1	0	10	9	126
16	119	2	1	5	0	9	10	56
24	8	2	1	0	0	31	80	57
32	1	1	1	1	0	1	1	1
40	1	1	1	2	1	1	0	1
48	2	1	1	2	1	1	1	1

Table 2. Euclidean distance measured in pixels from midpoint of region with highest match to midpoint of true region using the polynomial kernel without cross terms.

Rotation and Scale	-21^0	-15^0	-9^0	-3^0	3^0	9^0	15^0	21^0
0	2	0	0	0	0	0	0	0
8	2	1	1	1	1	1	1	1
16	1	2	1	2	9	9	9	7
24	2	2	1	2	9	9	9	1
32	1	1	1	2	8	1	1	1
40	1	1	1	2	1	1	1	1
48	10	1	2	2	1	1	1	1

4 Conclusion

A novel (non-linear) template matching technique was presented. The proposed framework, based on the derivation of an MNT using interpolator coefficients,

DITs and UITs, shows promising results. An efficient implementation strategy using the FFT and IFFT was highlighted. Future work will focus on the improvement of robustness and real-time implementation issues.

References

1. Theodoridis S. and Koutroumbas K. (1999), *Pattern Recognition*, Academic Press.
2. Ravichandran G. and Trivedi M.M. (1995), Circular-Mellin Features for Texture Segmentation, 2(12)1629-1641.
3. Hall E. (1979), *Computer Image Processing and Recognition*, Academic Press.
4. Uenohara M. and Kanade T. (1997) Use of the Fourier and Karhunen-Loeve Decomposition for Fast Matching with a Large Set of Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8)891-899.
5. Ben-Arie J. and Rao K.R. (1993), A Novel Approach for Template Matching by Nonorthogonal Image Expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, 3, 71-84.
6. Ben-Arie J. and Rao K.R. (1994), Nonorthogonal Image Expansion Related to Optimal Template Matching in Complex Images, *CVGIP: Graphical Models and Image Processing*, 56, 149-160.
7. De Figueiredo J.P. (1983), A Generalized Fock Space Framework for Nonlinear System and Signal Analysis, *IEEE Transactions on Circuits and Systems*, 30(9)637-647.
8. Van Wyk M.A. and Durrani T.S. (2000), A Framework for Multi-scale and hybrid RKHS-based Approximators, *IEEE Transactions on Signal Processing*, 48(12)3559-3568.
9. Máté L. (1990), *Hilbert Space Methods in Science and Engineering*, Adam Hilger, New York.
10. Wahba G. (2002), *Soft and Hard Classification by Reproducing Kernel Hilbert Space Methods*, Technical Report 1067, Department of Statistics, University of Wisconsin.
11. Aronszajn N. (1950), Theory of Reproducing Kernels, *Transactions of the American Mathematical Society*, (68)337-404.
12. Zyla L.V. and De Figueiredo J.P. (1983), Nonlinear System Identification Based on a Fock Space Framework, *SIAM Journal of Control and Optimization*, 21(6)931-939.
13. Luenberger D.G. (1969), *Optimization by Vector Space Methods*, John Wiley and Sons, New York.
14. Schölkopf B and Smola A.J. (2001), *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond (Adaptive Computation and Machine Learning)*, MIT Press.