Insights on using Non-Evolutionary Optimisation Methods for Template Based Image Registration

L. Machowski (0001084K)

Department of Electrical and Information Engineering,
University of Witwatersrand, South Africa.

Abstract

This paper investigates the use of non-evolutionary, or classical, optimisation techniques to register a template with a scene image. An error function is created to measure the correspondence of the template to the image. The problem presented here is to optimise the horizontal, vertical and scaling parameters that register the template with the scene. The nature of the optimisation algorithms tend to find the local minima of the objective function, therefore a simple technique is employed to determine feasible starting points in order to explore the entire scene image. The results show that gradient-based methods perform poorly due to the nature of the partial derivatives of real-world images. Numerical errors also contribute to their poor performance. The direct-search (simplex) method proves to perform very well, requiring fewer function evaluations than the gradient based methods. The algorithm successfully manages to register templates with lots of noise and substantial corruption.

Keywords: Optimisation, Registration, Template

I. Introduction

The automatic matching of a template image onto a scene has great practical application in the field of computer vision, medicine, remote sensing and image watermarking [1]-[3]. Being able to determine how best the template image fits into the scene poses several problems that have to be overcome. The registration process may involve shifting, scaling, rotation, perspective projection or other non-linear transformations. The shear number of possible transformations makes it difficult to automate the process and usually requires a person to verify the results. This paper presents findings on the use of non-evolutionary (classical) optimisation methods for automating the template matching of 2-dimensional intensity images.

A. Image Registration

Image registration is the process by which a template is oriented in such a way as to match an entire, or a portion of, a given scene [1][2]. Either the scene or the template is transformed in such a way that it matches the other image as closely as possible. This can be used to determine the position of characteristic objects in a scene, as is commonly done in automatic target recognition and tracking [4]-[6]. Image registration is used to stitch multiple views of a scene together to make a larger representation of the scene [1]. When several images of the same object are taken from different sources, the data must be combined or reinterpreted [7][8]. This sensor fusion requires that the position of the same object is known in each of the separate images so that the information can be processed.

There are four main steps required for registration of an image. These are feature detection, feature matching, transform model estimation and image transformation [1]. It is possible to extract higher level features from the image and attempt to perform the registration based on these. It normally means extracting the data into a feature space that is usually very specific. It is also possible to use area based detection which compares the actual image intensity values and other low level information that makes up the appearance of the image. Feature based detection makes it easier to determine the orientation of the template with respect to the scene. Area based detection methods are much more computationally expensive due to the amount of data that needs to be processed. Since the area based detection methods depend on the appearance of the images, they are intolerant of changes in illumination and ambient conditions [1][2]. The feature based detection methods do not suffer from this but it is more difficult to automatically extract the features for any general image. It is common to combine the advantages from both methods to form a hybrid approach to the registration process [2].

Correlation-like methods are typically used for area-based detection methods where a correlation surface is calculated for the template and the maximum point is found and interpreted as the best fit for the template [1]. This method is adversely affected by self similarity in the image and it is characterised by high computational complexity. It also does not allow much variance in template rotation or other more complex transformations. This approach, is still however attractive for real-time object tracking [1][6][9].

An alternative to cross correlation is to use optimisation to find the best fit for the template in the scene [7][8][10]. The advantage of this approach is that one can apply more complex transformations to the templates, and thus make the method robust when compared to cross correlation. This method also requires less computation because the correlation surface does not have to be determined.

In this paper, we use different non-evolutionary (classical) optimisation techniques to register a template to a particular scene. For simplicity, only three transformation parameters are defined. The first two, horizontal and vertical translation, define where in the scene the template belongs. They represent the most elementary operations required for image registration. In order to demonstrate the ability of the algorithms to register images with more complex transformations, a third parameter: uniform scaling, is introduced. This is a simple transformation where the size of the template is changed proportionally, while keeping the aspect ratio the same. It is intentionally left as a simple transformation so that we do not get distracted by the nuances of more complex transformations.

B. Non-evolutionary Optimisation

To find the optimal registration parameters for template matching, it is important to construct a multivariate cost function that represents how well the template matches the scene [11]. The traditional techniques for optimisation make use of the objective function value, first derivative or its second derivative [11][12]. The general approach for all non-evolutionary optimisation methods is to select an initial guess for the registration parameters and travel in a direction as to improve the objective function. Once a suitable direction is found, it is possible to make either fixed or varying successive steps towards the local optimum. A brief review of the methods suitable for image registration is given below:

1) Random Search

This method relies on trying random values for the objective function parameters [13]. Although this does not seem like a feasible technique, it has certain advantages. If one tries a sufficient number of samples then the global optimum will always be found. Another advantage of this method is that it does not require a continuous function or any gradient evaluations. The disadvantage is that a large number of samples need to be taken to find the global optimum.

2) Univariate Search

This method performs a series of linear (one dimensional) optimisations by keeping all but one parameter constant at a time [13]. This is a direct search method and therefore does not require any gradient evaluations. The one dimensional line search can make use of bracketing techniques or more complex gradient based methods [13]-[16]. It is only necessary to find the optimum in the line search approximately, since it must merely take us in the general direction of the multi-dimensional optimum.

3) Pattern Search

A pattern search relies on the accelerated performance of line searches along conjugate directions towards the optimum [13]. Most non-linear problems can be approximated by a quadratic function and it has been proven that searches along conjugate directions are quadratically convergent [13][17]. This method uses direct search and the most common algorithm is given by Powell's method (cited in [13]).

4) Simplex Method

A simplex is a geometric figure that has one more vertex than the number of dimensions in the parameter space (a triangle in two dimensions) [14]. The objective function is sampled at each vertex and the one that has the worst value gets removed from the simplex. A new vertex is then created by reflecting the simplex about the centroid. In this manner, the algorithm steps its way towards the optimum point. This method is relatively robust when used for discontinuous objective functions [15].

5) Steepest Descent

All of the direct search methods described above tend to converge relatively slowly towards the optimum. This is because the methods do not make complete use of the objective function information at the current sampling point [13]. By travelling in the opposite direction to the gradient of the objective function, the local optimum can always be found [14], and it can be shown that this process is linearly convergent [13]. The Steepest Decent method performs well when it is far from the optimum and the objective function is smooth and continuous. This is because the gradient tends to be large the further away we are from an optimum, and therefore the algorithm takes bigger steps in this situation.

6) Conjugate Gradient

This method makes use of a similar idea to conjugate directions already described in the Pattern Search. The journey towards the optimum is accelerated by using conjugate gradients with the steepest descent [13]. The optimal step size is computed at each iteration by performing a line search in a conjugate direction [17].

7) Scaled Conjugate Gradient

This method does not perform a line search to find the optimal step size, since it is computationally expensive [16]. It rather calculates it from the Hessian of the objective function [17] but this method is only appropriate when the Hessian is easily available, such as when an analytic solution can be provided.

8) Newton's Method

All of the Newton-like methods make use of a second order Taylor series expansion about the current sampling point to decide on a suitable search direction [13]. The new parameters for the next iteration are calculated as follows:

$$\underline{x}_{k+1} = \underline{x}_k + \underline{s}_k \tag{1}$$

where $\mathbf{\underline{x}}_{k+1}$ is the parameter vector for the next iteration, $\mathbf{\underline{x}}_k$ is the current parameter vector and $\mathbf{\underline{s}}_k$ is the current search direction. The search direction is calculated as:

$$\underline{s}_k = -\left(\nabla_{\underline{x}\underline{x}}{}^2 P\Big|_{\underline{x}_k} \right)^{-1} \left(\nabla_{\underline{x}} P\Big|_{\underline{x}_k} \right)^T \tag{2}$$

Where $\nabla_{\underline{xx}}{}^2 P\Big|_{\underline{x}_k}$ is the second derivative (Hessian matrix) of the objective function evaluated at $\mathbf{\underline{x}}_k$ and $\nabla_{\underline{x}} P\Big|_{\underline{x}_k}$ is the gradient of the objective function evaluated at $\mathbf{\underline{x}}_k$.

This method outperforms the Steepest Descent near to the optimum and often converges faster than Conjugate Gradient methods [16]. It does however require that the Hessian be calculated at each step as is shown in (2), followed by a matrix inversion.

It is therefore not suitable to use this method when it is difficult to get an analytical equation (that is differentiable) for the objective function. This automatically rules out using the direct form of Newton's method to most image processing tasks because it is expensive to find an analytic equation for any general image while still maintaining sufficient detail.

9) Quasi-Newton Method

The Quasi-Newton approach is suitable when the Hessian can only be calculated numerically, such as in image registration. It makes use of partial derivative information to update and build an approximate Hessian matrix which gives the second order curvature of the objective function [13][14]. This method therefore makes use of two approximations: the Taylor Series expansion and the approximate Hessian. There are two major variations for computing this approximate Hessian: the Davidon-Fletcher-Powell (DFP) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms. The algorithms differ by how they handle round-off error and convergence issues, while the BFGS algorithm is considered to be better for most general problems [13][15][16].

The Quasi-Newton methods require more memory storage and computation than the conjugate gradient methods. The extra computation is due to the approximate Hessian that must be calculated at each iteration, along with a matrix inversion, as shown in the equation for the search direction (2). This may pose a problem for large dimensional problems since the matrix has a n^2 relationship [16].

10) Levenberg-Marquardt Method

This method is a hybrid of the Steepest Descent and the Newton Method. When it is far from the optimum, it behaves like the Steepest Descent method, and thus always travels in a direction to decrease the objective function. When it approaches the optimum then it behaves like the Newton Method in order to converge quickly [13]. This is accomplished by modifying the diagonal of the Hessian matrix during iteration. The Marquardt method is most suited to non-linear least squares problems [13].

C. Pyramidal Approach

Registering real-world images by using non-evolutionary optimisations methods poses several problems in the form of multiple local-minima. These minima are also non-unique in areas of uniform intensity, which affects the optimisation due to the lack of a gradient when perturbing the registration parameters. For this reason, it is important to determine a suitable starting point for the optimisation, which can be achieved by pre-processing the image. The pyramidal approach to optimisation uses a coarse-to-fine hierarchical strategy to narrow down on the optimal solution [1]. With this approach, the consequences of making an error in the coarse levels have a large impact on the final solution. It is therefore important to perform a consistency check at each level.

II. Method

Before the template can be registered with the scene, it is necessary to select suitable starting points for the optimisation routine. This is important because all of the practically useful optimisation methods described above can only find local optima, which is problematic for reliable image registration. An image may have many areas where the template fits approximately, especially if it can be scaled, rotated or non-uniformly transformed. A pre-processing stage to find acceptable starting points is therefore critical for image registration using non-evolutionary algorithms.

A. Finding the Starting Points

A simple approach is taken to find regions in the scene that are likely to contain the template. First a histogram of the template is calculated and the top n bins with the most counts are used to categorise the template. The colours corresponding to these bins ($\mathbf{c}_{max}$) are then calculated and a threshold is applied to the scene image to show only the pixels that match $\mathbf{c}_{max}$. The starting points for the optimisation algorithm are chosen randomly from the pixel positions having those colours. Similar to the random search, this method is bound to find the global optimum if enough samples are taken. The thresholding is used to substantially decrease the number of possible starting points. This method assumes that the intensity of the template is the same as that of the scene image.

B. Objective Function

In order to perform optimisation, it is necessary to define an objective function that captures the essence of the problem at hand. In image registration, one wants to maximise correspondence between the scene image and the template posed at its current position. The correspondence can be measured as the sum-squared difference between the intensities of overlapping pixels. This can be expressed as an error function where a value of zero represents a perfect match. The parameters to be optimised in this problem are horizontal translation (x), vertical translation (y) and uniform scaling (s). The objective function used when there are overlapping pixels between the template and the scene image is given by:

$$error = \sum (A-T)^2 \Big/ numel(A) \qquad (3)$$

where A is the scene image, T is the template, the ^{2} refers to element-wise squaring and the summation is over each element of the resultant matrix. This error is then normalised with the number of pixels that are overlapping between both images. The gradient of this function can only be calculated numerically and a central difference approximation is used. Due to the discrete nature of images, it is important to use large perturbations when calculating the partial derivatives. A value of 2 pixels is used for the differences, meaning that translational perturbations are 2 pixels across and scaling perturbations change the image size by 2 pixels. It is also important to interpolate sub-pixel values for the optimisation algorithm to be able to function correctly. If the perturbation size is smaller than a pixel then inaccurate or highly discontinuous gradients can be calculated.

It is necessary to penalise the error function for the case when there are no overlapping pixels between the two images. The error function then becomes:

$$error = \frac{|x|}{T_w} + \frac{|y|}{T_h} \qquad (4)$$

where |x| and |y| are the absolute values of the positional parameters, T_w is the width and T_h is the height of the template in pixels. This has the effect of constraining the x and y parameters back into range when the images no longer overlap. It is necessary to hard-limit the scale parameter because the optimisation algorithms might try ridiculously high values which require extremely large amounts of memory. Very seldom does a template match a scene at very high scaling values. Similarly, if the template gets scaled to one pixel in size, then a fit can be found nearly anywhere in the scene.

C. Test Image

The image registration problem described in this paper has three parameters that would require 4-Dimensional plots (or volumetric renderings) to be able visualise the objective function. It is however possible to choose a sample image that effectively eliminates the one dimension in order to simplify visualisation. Such an image is given in Figure 1. The image is a linear gradient function with intensities varying uniformly between 0 and 1. The template is 'cut out' from the middle of the image and the global optimum is non-unique (any y position will give the minimum value). This image is suitable for understanding and observing how each algorithm performs. The error function values shown in the contour plot vary from 0 to 0.0322. The function is smooth and continuous but it is valley-like near to the optimum.

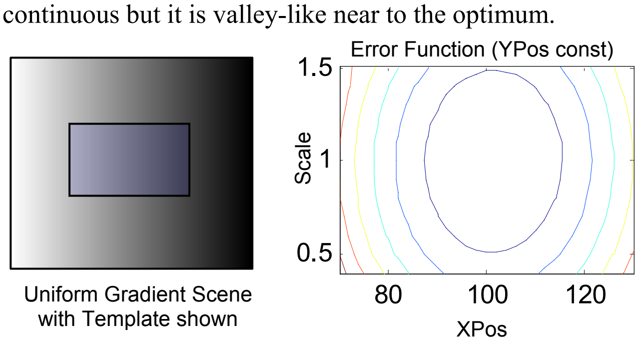

Figure 1 Y Position Invariant Test Function

D. Optimisation

Two different optimisation toolboxes are used to perform the automatic registration in Matlab. The Matlab Optimization toolbox [15] is used for its Simplex and Quasi-Newton (BFGS) methods. The Netlab [12] toolbox is used to compare Steepest Descent, Conjugate Gradient, Scaled Conjugate Gradient and Quasi-Newton methods.

E. Automatic Registration

The automatic registration process is shown in Figure 2.

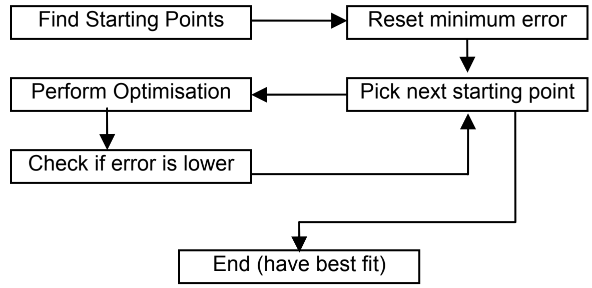

Figure 2 Automatic Registration Process

The algorithm begins by finding all the suitable starting points as described previously. A variable is created to keep track of the minimum error that has been calculated (smallest local optimum). The next starting point is then randomly selected from the set of possible values and it is flagged so that it is not chosen again. The optimisation is performed and if the error value is lower than any previously calculated value, then the position and scale parameters are saved. The minimum error variable is updated to this newly found value. The process repeats until either the minimum error is below some threshold or the complete set of starting points have been traversed. The minimum error variable will then contain the best guess at how the template should be registered to the scene.

III. RESULTS

A. Starting Points

Since the most common pixel colours are used to categorise the template, it is possible for the image to have substantial variability when compared to the template, as long as the most common colours stays the same. This makes the algorithm robust. It does however generate a large number of starting points, but experimentation has shown that on average 15 starting points are required before the template is matched.

B. Optimisation

A comparison of the results for the various methods is given in Table 1. The rows that appear greyed did not converge within the maximum number of iterations. It is important to remember that these results are in the context of image registration and the algorithms usually behave differently for more common objective functions. The nature of the test function is that it has a very shallow trough near the optimum and the numerical partial derivatives are expected to be badly scaled. Artefacts in these partial derivatives from the transformation and descretisation process cause havoc with some of the algorithms, especially the ones that calculate the Hessian matrix (such as SCG).

Table 1 Performance results for various methods

Method	Iter	Func	Grad	Time(s)	Err
Simplex	58	106	-	2.016	2.9e-12
Step. D.	!200!	201	200	22.375	1.6e-3
Conj. Gr.	5	137	5	3.078	5.8e-34
SCG	!200!	201	300	30.547	4.4e-3
Q.Newt	12	84	12	2.937	9.2e-14

1) Simplex Method

This method is especially robust for functions that have peculiar gradients or that are discontinuous, as is common in image registration. The robustness can be attributed to the fact that the simplex does not depend directly on gradients for its direction of travel. The method of travel by the simplex acts as a pseudo-gradient that plays a similar role as in the gradient methods. The typical disadvantage of this method is that it converges slowly but this method proved to give the most stable and reliable results. From the test results, it is clear that the simplex method took the fewest function evaluations (which are very computationally expensive), especially if it is taken into account that each gradient evaluation requires 6 function calls. The running time is therefore a good indication of the algorithm performance in this case. The behaviour of each algorithm is shown in Figure 3.

2) Steepest Descent

In image registration, it is common to have objective function contours that are elongated ellipses, such as in the test image. The Steepest Descent method does not perform well in these situations because it is extremely slow to converge [14]. This is clearly seen from the results in Table 1 and Figure 3. To remedy this, the gradient values returned to the algorithm are artificially scaled by a factor of 1000 and a momentum factor of 0.9 is used.

3) Conjugate Gradient

This method performs very well on the test image, but it tends to cause parameter explosions, where exceptionally large values are generated, when performing the line search. This is seen in the total number of function calls taken by the method. The parameter explosion often produces values that are in the penalise-able constraints. When tested on real-world images, the algorithm gets confused by the complex gradient information present in typical images.

4) Scaled Conjugate Gradient

Making the step size depend on the gradient is normally valid for well behaved objective functions. This is not always the case for registering a template to the image, where it is common to have steep partial derivatives close to the optimum due to transformation and descretisation artefacts. Since this method uses the Hessian matrix instead of a line search for its next step, the misbehaving gradient errors compound themselves and produce poor results.

5) Quasi-Newton

This method performs well on the test image and can be attributed to the objective function behaving very much like a quadratic smooth surface. When testing the algorithm on real-world images, it performs worse because of the misleading information that the gradients of the image give to the algorithm. This normally sends the parameter values into infeasible regions.

C. Other Sample Data

Appendix A shows the successful results of tests on other scenes and templates. The algorithm managed to register the template of a face to a scene of the entire person. The randomised choosing of starting points gave generally fast matching times, requiring and average of 15 starting points in order to achieve an accurate registration. The algorithm also proved to be robust in terms of its ability to register templates that are similar but substantially corrupted by noise. The only requirement was to adjust the stopping threshold on the minimum error value to suite the given template. The various modifications to the templates that were made include, Gaussian blurring, the addition of Gaussian noise, vortex rotation, smudging and non-uniform stretching.

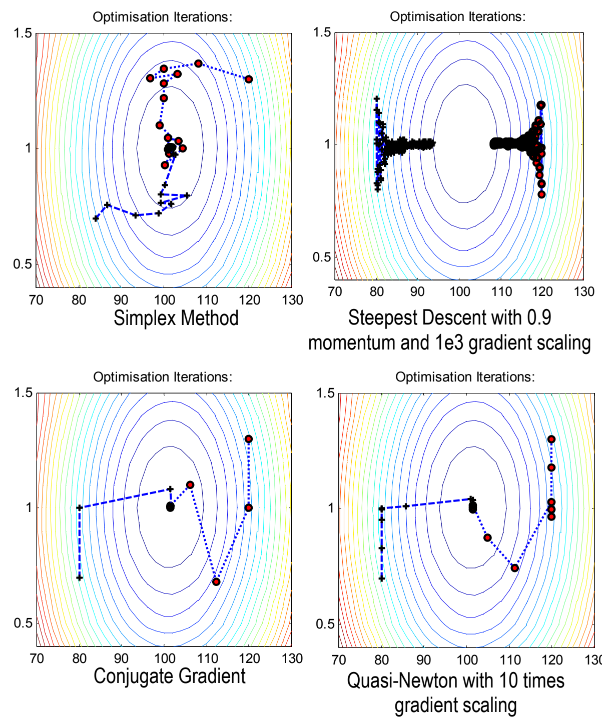

Figure 3 Iteration steps of the various methods.

IV. CONCLUSION

The poor performance of the gradient-based methods can be attributed to the numerical error in calculating partial derivatives and the form of the objective function for real world images. This discontinuous nature makes the use of gradient based methods computationally expensive and un-productive for image registration. It was demonstrated that the direct search method gives overall better results and does not suffer from parameter explosions. The simple technique employed to generate feasible starting points is merely required to overcome the problem of multiple local minima in the objective function. It is therefore expected that evolutionary techniques or particle swarm systems should be able to find the template matching parameters more reliably and predictably. This paper has shown that template based image registration using non-evolutionary techniques that do not evaluate gradient information are reliable and perform acceptably.

V. Appendix A: Successful Registrations

The following scene and templates images were successfully registered automatically using the simplex (Nelder-Mead) method and the choice of starting points described by the algorithm in this paper.

Image Scene:

Templates Matched Automatically:

References

[1] B. Zitova, J. Flusser. "Image registration methods: a survey"; Elsevier B.V.; Image and Vision Computing 21 (2003) 977–1000; June 2003.

[2] X. Peng, M. Ding, C. Zhou, Q. Ma. "A practical two-step image registration method for two-dimensional images"; Elsevier B.V.; Information Fusion; Article in press; 2004.

[3] S. Pereira, T. Pun. "An iterative template matching algorithm using the Chirp-Z transform for digital image watermarking"; Elsevier B.V.; Pattern Recognition 33 (2000) 173-175; 1999.

[4] I. Kumazawa. "Target tracking by matching a shape represented by a tree of sigmoid functions"; Elsevier B.V.; Pattern Recognition Letters 21 (2000) 661-675; 2000.

[5] Y. Bogomolov, G. Dror, S. Lapchev, E. Rivlin, M. Rudzsky. "Classification of Moving Targets Based on Motion and Appearance"; Proceedings of the British Machine Vision Conference; 2003. http://www.bmva.ac.uk/bmvc/2003/table_of_contents.html Last Accessed: 05/03/2004

[6] S. Chien, S. Sung. "Adaptive window method with sizing vectors for reliable correlation-based target tracking"; Elsevier B.V.; Pattern Recognition 33 (2000) 237}249; 1999.

[7] R. Sharma, M. Pavel. "Multisensor image registration"; Proceedings of the Society for Information Display XXVIII (1997) 951–954; 1997.

[8] P. Viola, W. Wells. "Alignment by maximization of mutual information"; International Journal of Computer Vision 24 (1997) 137–154; 1997.

[9] D. Tsai, C. Lin. "Fast normalized cross correlation for defect detection"; Elsevier B.V.; Pattern Recognition Letters 24 (2003) 2625–2631; 2003.

[10] H. Lim, S. Hossein Cheraghi. "An optimization approach to shape matching and recognition"; Computers & Electrical Engineering 24 (1998) 183-200; 1998.

[11] E. Chong, S. Zak. "An Introduction to Optimization, 2nd Ed."; Wiley; 2001.

[12] I. T. Nabney. "Netlab: Algorithms for Pattern Recognition"; Springer; UK; 2002.

[13] S. Chapra, R. Canale. "Numerical Methods for Engineers, 4th Ed."; McGraw-Hill; USA; 2002.

[14] T. Marwala. "ELEN 5015 Course Notes: Optimisation"; School of Electrical and Information Engineering; University of Witwatersrand; 2004.

[15] "Optimization Toolbox User's Guide V2"; The Mathworks Inc.; 2002.

[16] "Neural Network Toolbox User's Guide V4"; The Mathworks Inc.; 2002.

[17] T. Marwala. "Fault Identification using Neural Networks and Vibration Data"; PhD Thesis; University of Cambridge; 2000.