

Image template matching using Mutual Information and NP-Windows

NDH Dowson*, R Bowden* and T Kadir†

*Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK

†Siemens Molecular Imaging Advanced Applications, 23-38 Hyth Bridge Street, Oxford, OX1 2EP, UK

{n.dowson;r.bowden}@surrey.ac.uk; t.kadir@robots.ox.ac.uk

Abstract

A non-parametric (NP) sampling method is introduced for obtaining the joint distribution of a pair of images. This method is called NP windowing and is equivalent to sampling the images at infinite resolution. Unlike existing methods, arbitrary selection of kernels is not required and the spatial structure of images is used. NP windowing is applied to a registration application where the mutual information (MI) between a reference image and a warped template is maximised with respect to the warp parameters. In comparisons against the current state of the art MI registration methods NP windowing yielded excellent results with lower bias and improved convergence rates.

1. Introduction

NP windowing is a new method for obtaining the joint distribution of a pair of images, which is equivalent to sampling the images at infinite resolution. Unlike existing methods, arbitrary selection of a kernel is not required, and the spatial structure inherent to images specified as a lattice of pixel values is used. To demonstrate its effectiveness, we apply NP windowing in a registration application relying on the maximisation of Mutual Information (MI). For registration, MI measures the similarity of a template and a corresponding region within a reference image. The corresponding region varies with a set of warp parameters. The actual MI value is measured from an estimate of the joint probability distribution function (PDF) or joint histogram.

MI is widely used in registration applications, due to its robustness to occlusion and noise, and invariance to non-linear intensity relationships. The latter is particularly useful for registering multi-modal images in medical imaging [1] and tracking objects in rapidly changing lighting conditions [2], to name just two applications. Moreover, MI is only slightly more expensive to compute than the (widely used) sum of squared differences measure.

Despite its good performance the registration accuracy for MI may still be improved upon, since only a limited

number of samples are available. This limits the resolution of the PDF estimate, because overly sparse histograms are statistically meaningless [9]. Also, the position of the image lattice points relative to the world is arbitrary as are the histogram bin boundaries relative to the light spectrum. So small changes in position result in unpredictable changes in the PDF and instability in the MI value. Using each pixel as a single sample assumes independence and identical distribution. The fact that these assumptions are often incorrect, is generally ignored.

Parzen windowing [9] and Partial Volume Estimation (PVE) [4, 1] have been proposed to improve the estimates and improve stability. Parzen windowing accounts for uncertainty in intensity by blurring the histogram slightly. PVE explicitly smooths MI values for small shifts by using a boxcar to measure the relative effects of nearby pixels. However both require sufficient samples to populate the histogram and may suffer from artifacts and bias [7]. Parzen windowing requires a kernel to be specified beforehand, and both require the kernel size and histogram bin size to be specified.

A method without these shortcomings was proposed by Kadir and Brady, where the statistics in arbitrary regions of interest in single images are estimated in [3]. Notably the ordering of the samples is used in their method. This is the foundation of this work. However we apply it to the more complicated case of a pair of 2D images. In addition substantial simplifications to the theory are made using Green's Theorem and standard polygon rendering methods. This avoids consideration of multiple geometric cases.

For test purposes the registration accuracy of this method was compared to the current state-of-the-art methods using MI. Testing was performed on eight real data sets and results are given in terms of convergence and bias. Excellent results were obtained. Importantly, the implementation of this method is rapid enough for practical use and has been made freely available on the web at www.ee.surrey.ac.uk/personal/n.dowson.

The remainder of the paper is organised as follows. The details of the proposed NP windowing method are presented

in Section 2. Section 3 describes the experiments and discussed the results obtained, before concluding in Section 4.

2. MI from NP windowed PDFs

This section describes the proposed NP windowing method used to obtain a PDF, and hence an MI value. To begin with, we recall from standard probability theory [6] that a function of a random variable creates another random variable whose distribution may be determined by the *transformation formula* between the two.

To illustrate, consider a group of adjacent pixel values in a template image (e.g. $c_1, a_1 + c_1$ etc. in Fig. 1d). In a registration application, the pixel values have corresponding values in a second reference image (e.g. $c_2, a_2 + c_2$ etc. in Fig. 1a). The combined groups are referred to as a *neighbourhood*.

Intensity y at any point $\mathbf{x} = (x_1, x_2)$ may be computed using an interpolation method. In Fig. 1 colour is used to indicate the half bi-linear interpolated intensity. The neighbourhood size varies with the interpolation method e.g. the size for bi-cubic interpolation is 2 images x 16 points and for full bi-linear interpolation is 2 images x 4 points.

The position of the lattice points relative to the world is arbitrary, so \mathbf{x} is treated as a random variable with bounded uniform distribution: $0 \leq x_1, x_2 < 1$, as shown in Fig. 1c. The bounds arise naturally from the lattice spacing. This implies that \mathbf{y} is also a random variable, with a distribution that may be found using a transformation formula. The bounds on f_x are also transformed to form bounds on f_y as shown in Fig. 1b. The bounded region of intensity probabilities is called the *intensity polygon*.

This leads to an algorithm which in outline consists of four steps. 1. Obtain all neighbourhoods in image 1 and 2 for a given warp and interpolation method. 2. Calculate the intensity polygon for each set of samples. 3. Render these to a histogram, or alternatively locate all polygon intersections and generate a new list of intensity polygons with no overlap. 4. Calculate the MI.

Step 1 is standard and is not discussed further except to mention that interpolation methods will in general only approximate the true band-limited signal and hence the true PDF. Better approximations may be obtained by prior knowledge of the sampling pre-filter or using empirical estimation [10]. For nearest neighbour interpolation this method breaks down to standard sampling.

For illustrative purposes step 2 is first discussed for the case of a single linear interpolated 1D signal. The neighbourhood is of size two. Each pair of adjacent samples uniquely defines a straight line, denoted as

$$y = ax + b \quad | \quad 0 \leq x < 1 \quad (1)$$

where x and y are simply the scalar forms of spatial position \mathbf{x} and intensity \mathbf{y} .

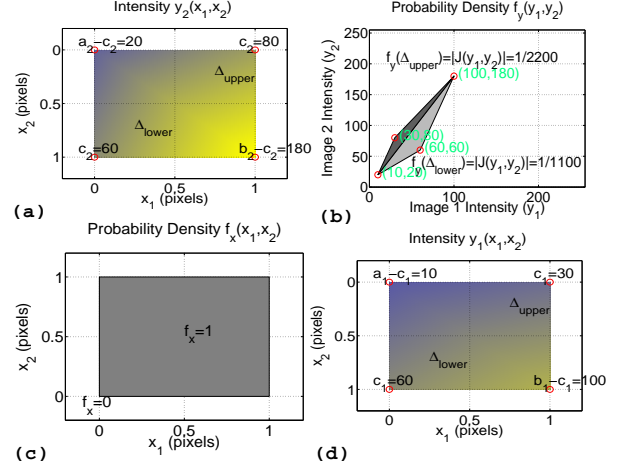


Figure 1. (d) & (a) Half Bi-linear interpolated intensities for two images: respectively $y_1(\mathbf{x})$, and $y_2(\mathbf{x})$. (c) The position of the lattice points is random but bounded with distribution f_x . (b) Using a transformation formula we obtain the distribution of intensities f_y . Note how the vertices of the two triangles in (b) correspond to pixel values in images 1 and 2 respectively in (a) and (d).

As discussed x is treated as a uniformly distributed random variable: $f_x(x) = 1$, $0 \leq x < 1$. The PDF of y , $f_y(y)$ is simply a transformation of f_x using the inverse function $x(y) = \frac{y-b}{a}$.

$$f_y(y) = |J(y)| f_x(x(y)) \quad (2)$$

where $|J| = \left| \frac{\partial x}{\partial y} \right|$ i.e. the determinant of the Jacobian of x with respect to y . Conveniently in this case $|J(y)| = |a|$, which is constant, so

$$f_y(y) = \begin{cases} \frac{1}{|a|} & y \in [a; a+b) \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

f_x acts as a bounding function, giving the y bounds shown.

Compared to single 1D case above, the derivation of $|J|$ and the subsequent evaluation of the PDF is rather more complex for pairs of 2D images. Initially the simplest useful 2D interpolation method is considered: half bi-linear interpolation (BLI). Half-BLI uses neighbourhoods of three values from each image: interpolating within two triangular areas. Fig. 1 shows two such neighbourhoods where Half-BLI is used. This yields equations of the form $y_i = a_i x_1 + b_i x_2 + c_i$, where i is the index of each image. The pair of inverse functions is:

$$x_1 = \frac{c_1 b_2 - b_1 c_2 - b_2 y_1 + b_1 y_2}{b_1 a_2 - a_1 b_2} \quad x_2 = \frac{c_1 a_2 - a_1 c_2 - a_2 y_1 + a_1 y_2}{-b_1 a_2 + a_1 b_2} \quad (4)$$

$$|J| = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} \end{vmatrix} = \begin{vmatrix} \frac{b_2}{a_1 b_2 - b_1 a_2} & \frac{-a_2}{a_1 b_2 - b_1 a_2} \\ \frac{-b_1}{a_1 b_2 - b_1 a_2} & \frac{a_1}{a_1 b_2 - b_1 a_2} \end{vmatrix} = \frac{1}{a_1 b_2 - b_1 a_2} \quad (5)$$

is the result, which is twice the inverse of the area of the intensity polygon (triangle) defined by the points $(c_1 + a_1, c_2 + a_2)$, $(c_1 + b_1, c_2 + b_2)$, (c_1, c_2) . This is exactly what is expected for a constant $|J|$, since each triangle represents half a sample and each sample integrates to one.

Having obtained the intensity polygon for each neighbourhood, we now proceed to step 3: obtaining the PDF estimate for the entire image. This simply consists of summing the distributions for each neighbourhood together:

$$\rho_{\mathbf{y}}(\mathbf{y}) = \frac{1}{N_n} \sum_{n=1}^{N_n} \frac{1}{|J_n(\mathbf{y})|} f_{x_n}(\mathbf{x}_n(\mathbf{y})) \quad (6)$$

where n indexes each neighbourhood. N_n is the number of neighbourhoods and is used to normalise the values so that the the PDF integrates to one.

Two methods for evaluating (6) were considered. First, we considered intersecting and summing the list of polygons to obtain a new list of non-overlapping polygons. Even in the relatively simple cases of constant distributions for half-BLI this is time consuming. In the case of varying distributions this becomes an intractable problem.

The second option, favoured here, is to use existing polygon drawing methods to “render” each polygon to an approximate PDF: i.e. a histogram. The drawing routine sums rather than replaces “pixel” values. The loss in accuracy is only slight if sufficient bins are used. Distributions that vary within a polygon may be represented as “texture”.

Special cases where polygons collapse to lines or points are easily detectable because their area is zero. Points also have a perimeter of zero. Points are modelled as unit impulses in the PDF. Since the intensity distribution is constant, lines are normalised by their length and split into segments wherever they intersect bin boundaries. Bins are incremented by the proportion of each line’s length they contain. Non-varying distributions use the same procedure but require explicit integration over each segment.

The cost of evaluating the histogram is dependent on the number of neighbourhoods and the average image gradient, i.e. approximately $O(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} N_n)$. The costs are relatively constant since intensity polygons are added to the histogram using a standard polygon drawing routine with flood filling at the polygon centre and (precise intersection based) anti-aliasing at the edges.

MI may be obtained directly from the PDF estimate using the standard equation:

$$I = \int_{\forall \mathbf{y}} \rho_{\mathbf{y}}(\mathbf{y}) \log \left(\frac{\rho_{\mathbf{y}}(\mathbf{y})}{\rho_{y_1}(\mathbf{y}) \rho_{y_2}(\mathbf{y})} \right) d\mathbf{y} \quad (7)$$

where $\rho_{y_1} = \int \rho_{\mathbf{y}} dy_2$ and $\rho_{y_2} = \int \rho_{\mathbf{y}} dy_1$. Of course in this case since the PDF estimate is actually a histogram, the integrals convert to sums.

Due to the finite size of the bins and the limited number of samples, MI is generally under-estimated. The residual error was found by Moddemeier to be approximately proportional to the square of the bin size [5] and inversely proportional to the number of samples. The error due to finite number of samples is eliminated by NP windowing. Accurate PDFs may be obtained even when there are few samples e.g. for small image patches.

The residual error due to bin size may also be eliminated by estimating MI directly from $\rho_{\mathbf{y}}$ i.e. without constructing a histogram. However as already discussed this would require a polygon intersection routine. Polygon intersection was implemented and works for simple cases, but became intractable for even small numbers of triangles (50).

2.1. Varying distributions

As alluded to earlier, for some kinds of interpolation the distributions in histogram space are *not* constant within each polygon. One such example is considered here: full bi-linear interpolated images. These distributions are more difficult to deal with, since they require explicit integration over their area to obtain a PDF.

Considering the example just given, the equations for a bi-linearly interpolated pair of images have the following shape: $y_i = a_i x_1 x_2 + b_i x_1 + c_i x_2 + d_i$, where i indicates the image index (i.e 1 or 2). This yields two sets of inverse functions of form $x_i(y_1, y_2) = P(y_1, y_2, 1) + \sqrt{P(y_1, y_2, 2)}$, where $P(\dots, d)$ indicates a polynomial of degree d . We use this concise form because of the limited space available. Both sets of solutions give the same solution for $|J|$:

$$|J(y_1, y_2)| = \left[(b_2 c_1 - b_1 c_2 + a_2 (d_1 - y_1) - a_1 (d_2 - y_2))^2 - 4(a_2 c_1 - a_1 c_2)(b_2 (d_1 - y_1) - b_1 (d_2 - y_2)) \right]^{-\frac{1}{2}} \quad (8)$$

As for the half bi-linear case, the valid region is bounded by a polygon with vertices corresponding to the pixel values of the neighbourhood: (d_1, d_2) , $(d_1 + b_1, d_2 + b_2)$, $(d_1 + c_1, d_2 + c_2)$ and $(d_1 + c_1 + b_1 + a_1, d_2 + c_2 + b_2 + a_2)$.

Evaluating $\int |J| dy_2 dy_1$ between such complicated boundaries is difficult, because multiple geometric cases must be considered. This can be avoided by splitting the quadrilateral into two simple polygons (triangles) and using Green’s Theorem to re-parameterise the problem.

Green’s Theorem states $\int_R \left(\frac{\partial J'_1}{\partial y_1} - \frac{\partial J'_2}{\partial y_2} \right) dA = \oint_C J'_2 dy_1 + J'_1 dy_2$, where C defines the curve around a region R . Letting $J'_2 = 0$ and $J'_1 = \int |J(y_1, y_2)|^{-1} dy_2$ yields:

$$\rho = \oint_C J'_1(y_1, y_2) ds \quad (9)$$

We now substitute $\mathbf{y}'(s) = (m_j s + c_j, s)$ for $\mathbf{y} = (y_1, y_2)$, where m_j and c_j are respectively the gradient and

y-intercept defining line segment j of the polygon. Applying the rules of integration by substitution (9) becomes:

$$\begin{aligned} \rho &= \oint_C J'_1(y_1, y_2) ds = \int_{s_0}^{s_1} J'_1(y'(s)) \left| \frac{dy'(s)}{ds} \right| ds \\ &= \int_{s_0}^{s_1} J'_1(m_j s + c_j, s) \sqrt{1 + m_j^2} ds \end{aligned} \quad (10)$$

The result of the above (double) integration is an equation containing more than 100 terms and factors. For the sake of interest the form of the equation is displayed here.

$$\begin{aligned} \rho &= k_1 \sqrt{P(s, 2)} + P(s)(k_2 + \log[P(s) + k_3 \sqrt{P(s, 2)}] \\ &+ k_4 \log[P(s) + k_5 \sqrt{P(s, 2)}] + k_6 \log\left[\frac{P(s) + k_7 \sqrt{P(s, 2)}}{P(s)}\right] \end{aligned} \quad (11)$$

Clearly the implementation of (11) although giving a precise result would be too slow for practical use. Several hundred operations would be required just to initialise the constants for each polygon. In addition a few tens of operations would be required to calculate *every* bin overlapped by the polygon. Contrast this with half bi-linear interpolation. Nine operations (4 subtractions, 2 multiplies, 3 additions) are required to calculate the area of each triangle. A flood fill routine (one add per bin) is used over most of the overlapped region, i.e. Half BLI is about two orders of magnitude faster than Full BLI. Since the potential improvement in accuracy comes at such a great cost, an implementation for full BLI is left for future work.

Using bi-cubic interpolation (BCI) in an NP windowing framework turns out to be even more problematic. The intensity equations for BCI have form $y_i = P(x_1, x_2, 3)$, which are not invertible.

3. Experiments and Results

To demonstrate the superiority of NP windowing when estimating MI it was compared to the current state of the art MI registration methods. In all, four methods were compared: NP windowing, standard sampling (One sample per pixel), third order partial volume estimation [1] and second order B-spline Parzen windowing [9]. Since derivatives were not directly available in some cases, optimisation was performed using Powell's Direction Set method [8]. In all cases MI was maximised for translation warps only. BLI was used to obtain reference image values corresponding to template lattice points.

Testing was performed using eight sets of high resolution data (2560x1920): shown in Fig. 2. In each case the full image was used as a reference and two small sub-regions were selected as templates to be matched to the original. All the images were then blurred with a normalised 12x12 top hat function and down-sampled by 12 times. The two templates extracted from each image differed in that the first template



Figure 2. The eight sets of data used for testing

was chosen to align exactly with the lattice of the down-sampled reference, while the second template was offset from the first by $(+\frac{1}{3}, +\frac{1}{3})$ of a (down-sampled) pixel.

The methods were compared using two error measures: bias and convergence. It is well known that different sampling methods create artefacts in the cost function surface of MI [7]. These artefacts can cause the maximum to shift away from its "true" position. This is referred to as bias. Bias was measured by performing a hierarchical brute-force search to locate the position of the maximum and computing its distance from the ground truth.

In addition, *local* maxima also exist, which optimisation methods may erroneously converge to. Convergence was measured by finding the mean distance from the points of convergence to the *biased* global maximum. The tests were performed by initiating convergence from 1000 random starting positions at 5 preset distances. The starting points were offset by the same amount from the global maximum for all the methods tested. The distance to the biased global maximum, rather than the ground truth, was used to decouple the effects of bias and local maxima.

Two sets of analyses were performed. Firstly, performance was measured as the number of bins in the joint histogram was varied from 16^2 to 256^2 bins in multiples of four. Secondly, the size of the template was varied in size from 9^2 to 17^2 pixels.

The results for bias while varying the number of bins is given for both lattice aligned and non-lattice aligned templates in Fig. 3a and b, respectively. Not unexpectedly the bias was very low for the lattice aligned templates (if non-zero). Overall NP windowing performed the best in this case. However, lattice alignment is unusual and these results are given to contrast them with results from a non-lattice aligned template. As demonstrated there is a large difference in bias in these two cases.

In the non-lattice aligned case notice that NP windowing is again the best performer and unlike the other methods is almost completely unaffected by the number of bins in the joint histogram. For the other methods performance steadily drops with increasing histogram sizes due to the lack of samples. Surprisingly, PVE performed the worst in terms of bias. This could be due to data blur induced by the large spatial support. Blur would effectively lower the

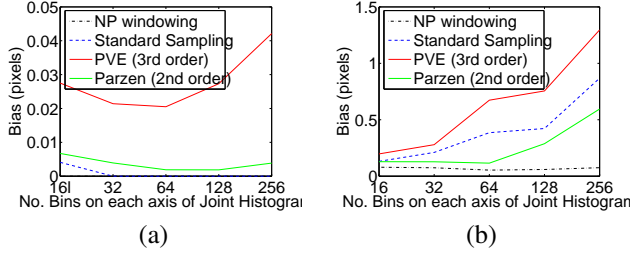


Figure 3. Bias for (a) Lattice & (b) Non-Lattice Aligned Templates for varying no. bins. Bias is low for lattice aligned templates but lattice alignment seldom occurs in real applications. In (b) performance degrades as the number of bins increases due to lack of samples. The exception is NP windowing, which is unaffected by the no. bins.

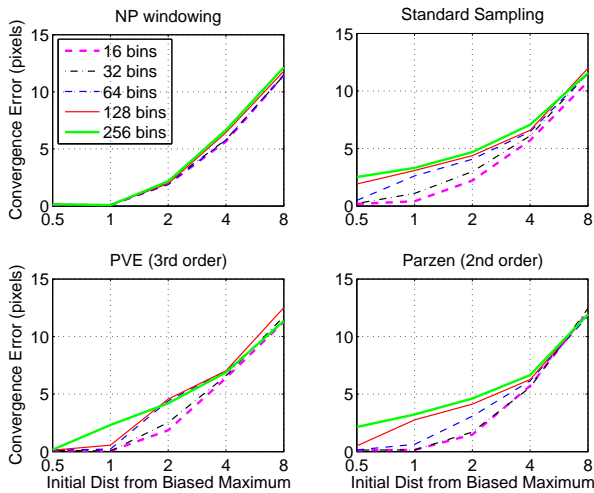


Figure 4. Convergence Error for Lattice Aligned Templates for varying no. bins.

significance of each sample speeding up the onset of performance loss due to limited samples.

Similar conclusions may also be drawn from the convergence results in Fig. 4 and 5. The number of bins has much less influence on NP windowing than other methods. For the other methods, at nearby starting points (to the biased global maximum) there is similarly good performance whatever the histogram bin-size. Likewise at distant starting points the performance is similarly poor. At intermediate distances however, there is generally a range of performance. It is at these distances where success or failure are similarly likely, that the influence of the algorithm's parameters are clearest.

PVE was the second best performer in terms of convergence, because of a larger basin size arising from extended spatial support. Apart from PVE though, it appears that the basin of convergence is neither affected by the number of histogram bins, nor by the sampling method used.

The results for bias while varying template size is given

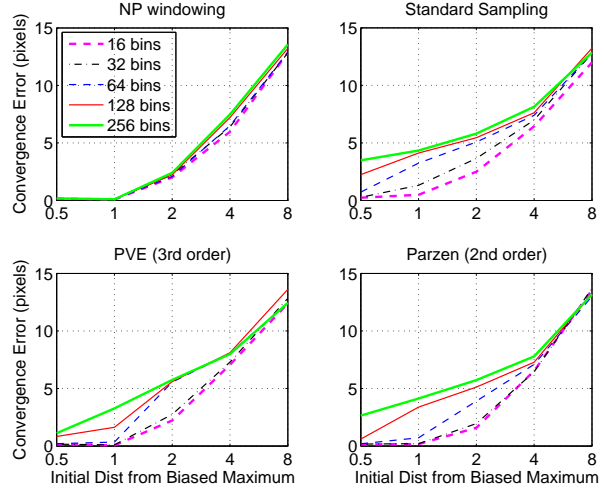


Figure 5. Convergence Error for Non-Lattice Aligned Templates for varying no. bins. NP windowing gives the best performance and is virtually unaffected by no. bins. Generally, at nearby and distant starting points convergence is similarly likely and unlikely, respectively. At intermediate points a threshold of success/failure exists and algorithm parameters have the most influence.

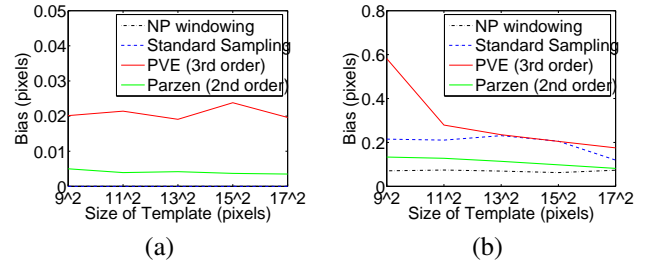


Figure 6. Bias for (a) Lattice & (b) Non-Lattice Aligned Templates for varying template size. In general performance improves as the number of pixels (samples) increases. The exception is NP windowing, which maximises the use of available data and is almost unaffected by no. samples. (On B/W printouts (a) appears to have 2 lines, since standard sampling and NP windowing had zero bias.)

for both lattice aligned and non-lattice aligned templates in Fig. 6a and b, respectively. Only the convergence results for non-lattice aligned template are given in Fig. 7, since lattice aligned solutions are unusual and space is limited.

In Fig. 6 the bias generally decreases as the number of available samples increases, because the statistics represent the data better and tends to dominate over the local blurring effects of the kernel (if one is used). This is particularly evident for PVE. The decrease in bias is negligible for NP windowing, which it makes maximal use of available information. Additional pixels add mainly redundant data.

In Fig. 7 showing the convergence tests the point of inflexion shifts rightwards as the template size increases. This indicates that additional data can increase the size of the basin of convergence despite adding only redundant information. Parzen windowing seems to be least affected by the

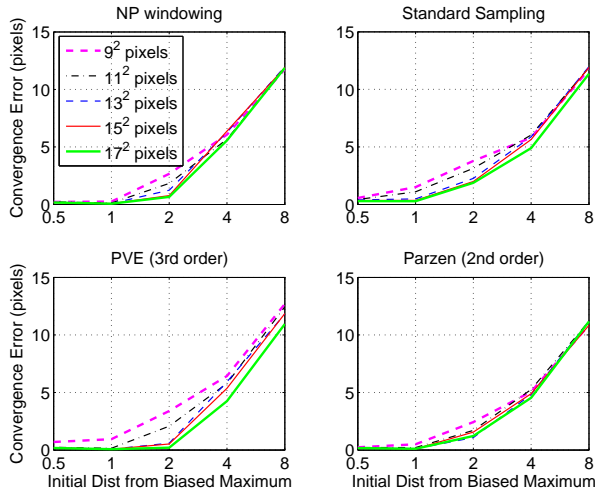


Figure 7. Convergence Error for Non-Lattice Aligned Templates when varying template size. Again a spread of performance at intermediate starting distances is observed. Parzen windowing has the lowest range at intermediate points and PVE has the best convergence for 17x17 templates. But NP windowing has the best mean performance at intermediate values.

number of samples (lowest range at intermediate values), and PVE gives best performance for the 17x17 template due to its wide spatial support. However, NP windowing gives a better mean performance than either method and performs the best in the most difficult case (9x9 template).

4. Conclusion

In this paper the NP windowing method for obtaining the joint distribution of a pair of images was introduced. Unlike existing methods the proposed approach does not assume sample independence or identical distribution. Moreover, the method does not require arbitrary parameters like kernel size to be chosen. NP windowing effectively eliminated the error caused by overly sparse sampling.

NP windowing for half bilinear and bi-linear interpolation schemes were developed theoretically. Because of its simplicity and speed the half bi-linear approach was implemented. The implementation utilised standard polygon rendering routines to generate joint histograms.

The histograms obtained using NP windowing were used in registration applications to maximise the Mutual Information between a reference image and a template with respect to some warp parameters. The registration accuracy was compared to existing state of the art methods using MI. Performance was measured in terms of bias and convergence using eight test sets.

In general NP windowing had less bias than any of the methods it was compared to (> 40% less than nearest competitor in some cases). In addition the number of histogram bins and sample size hardly affected NP window-

ing, implying that it makes maximal use of available data. NP windowing usually had the best convergence properties, although the increased spatial support of 3rd order PVE gave it a slightly improved convergence properties over NP windowing. However NP windowing tolerates decreasing sample size better. The radius of the basin of convergence increases with sample size, but is unaffected by histogram bin-size. The test data and binaries used for this work have been made freely available on the web at www.ee.surrey.ac.uk/personal/n.dowson.

In general NP windowing is recommended for applications where bin-size should be a non-critical choice and where the number of samples is limited or unknown. Moreover in domains where the bias (shift in true global maximum) must be kept to a minimum we would recommend the use of NP windowing.

In future work NP windowing using more sophisticated interpolation methods, with varying distributions, will be implemented. In addition the authors are looking at ways to eliminate the intermediate step of forming a histogram and instead evaluate Mutual Information directly. Finally, extensions to 3D voxel data sets should be investigated.

References

- [1] H. Chen and P. Varshney. Mutual information-based CT-MR brain image registration using generalised partial volume joint histogram estimation. *IEEE. Trans. Medical Imaging*, 22(9):1111–1119, September 2003. 1, 4
- [2] N. Dowson and R. Bowden. Simultaneous modelling and tracking (smat) of feature sets. volume 2, pages 99–105, San Diego, CA, USA, June 2005. 1
- [3] T. Kadir and M. Brady. Estimating statistics in arbitrary regions of interest. In *Proc. British Machine Vision Conf.*, volume 2, pages 589–598, Oxford, 2005, September 2005. 1
- [4] F. Maes, D. Vandermeulen, and P. Suetens. Medical image registration using mutual information. *Proc. of the IEEE*, 2003. 1
- [5] R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16:233–248, 1989. 3
- [6] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Inc., 3rd edition, 1991. 2
- [7] J. Pluim, J. Maintz, and M. Viergever. Interpolation artefacts in mutual information-based image registration. *Computer Vision And Image Understanding*, 77:211–232, 2000. 1, 4
- [8] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992. 4
- [9] P. Thevenaz and M. Unser. Optimization of mutual information for multi-resolution image registration. *IEEE Trans. On Image Processing*, 9(12):2083–2099, December 2000. 1, 4
- [10] B. Triggs. Empirical filter estimation for subpixel interpolation and matching. In *Proc. Int. Conf. on Computer Vision*, pages 550–557, 2001. 2