A Unifying Framework for Mutual Information methods for use in Non-linear Optimisation

Nicholas Dowson and Richard Bowden

Centre for Vision Speed and Signal Processing
University of Surrey, Guildford, GU2 7JW, UK
{n.dowson;r.bowden}@surrey.ac.uk
http://www.ee.surrey.ac.uk/personal/n.dowson

Abstract. Many variants of MI exist in the literature. These vary primarily in how the joint histogram is populated. This paper places the four main variants of MI: Standard sampling, Partial Volume Estimation (PVE), In-Parzen Windowing and Post-Parzen Windowing into a single mathematical framework. Jacobians and Hessians are derived in each case. A particular contribution is that the non-linearities implicit to standard sampling and post-Parzen windowing are explicitly dealt with. These non-linearities are a barrier to their use in optimisation. Side-by-side comparison of the MI variants is made using eight diverse data-sets, considering computational expense and convergence. In the experiments, PVE was generally the best performer, although standard sampling often performed nearly as well (if a higher sample rate was used). The widely used sum of squared differences metric performed as well as MI unless large occlusions and non-linear intensity relationships occurred. The binaries and scripts used for testing are available online.

1 Introduction

Our aim is to place the common variants of Mutual Information (MI) into a single mathematical framework, and provide their analytic derivatives for use in non-linear optimisation methods. Furthermore an evaluation of the MI variants is provided allowing other researchers to choose a particular variant in an informed manner. We demonstrate that the four most commonly used variants, namely: standard sampling, Partial Volume Estimation, In-Parzen Windowing and Post-Parzen Windowing; vary primarily in how the joint histogram is sampled. The Jacobians and Hessians are derived for all these methods using an approach similar to that of Thevenaz and Unser [1], but who considered In-Parzen Windowing only. In the cases of standard sampling, post-Parzen window estimation and higher order partial volume estimation, this is novel. Using the established framework, the methods are compared in terms of computational cost, and convergence to the ground truth for eight data sets.

Generally papers using MI choose the method reported to best suit their application, without the scope to consider other methods. The obvious exceptions are several papers that discuss artefacts on the MI cost function surface

[2, 3] and optimisation strategies [4]. However, only empirical analyses are made (unlike the analytic comparisons here). Also, neither artefacts nor optimisation strategies are the focus here. Rather, we focus on providing a common framework and side by side comparison of MI methods, which to the authors' knowledge is currently unpublished elsewhere. The binaries and scripts used for testing are available online.

Registration, or aligning one image (template) relative to another image (reference), is a common problem in many machine vision applications: *e.g.* tracking, image mosaicking, object matching, and multi-modal registrations in medical imaging. A widespread strategy for registration is to minimise (or maximise) a similarity metric between a template and the region of overlap in a reference image using an optimisation algorithm. MI has proved to be superior to many other metrics. Since its concurrent introduction and popularisation by Viola and Wells [5], Studholme *et al.*[6] and Collignon *et al.*[7] it has been widely adopted.

Shannon proposed MI [8] in his theory of information as a measure of entropy of the information shared between two signals, with quantised *amplitudes* over a period of *time*. It is a simple extension to consider 2D or 3D images rather than 1D signals, which consist of quantised *intensities* over a 2D/3D *space*.

MI has been applied using many different optimisation methods with varying degrees of success, including the simplex algorithm [9], Powell's method [7, 10], Gradient Descent [11], hierarchical brute-force searches [12] hierarchical approaches [1, 13]. Pluim *et al.*'s survey [14] cites many more examples. Several optimisation methods were systematically compared by Maes in [15]. Due to space constraints, such a comparison is beyond the scope of this paper, but all the optimisation methods in [16] (Ch.10) and the Levenberg-Marquardt algorithm [17] have been implemented.

The advantages of MI include an invariance to changes in lighting conditions, robustness to noise, sharp maxima and computational simplicity [18]. In a comparative study of registration methods, an MI based algorithm outperformed 15 other algorithms [19]. However, MI is a non-linear function and is prone to artefacts in its cost function surface. To overcome this, other forms of MI have been developed. One approach, used by Wells *et al.*[11] is to convolve the histogram with a Parzen window [20], to account for uncertainty in the intensity values. Thevenaz and Unser have a more sophisticated Parzen windowing method using B-splines [1], which are applied during the construction of the histogram, giving more accurate results. In addition, Partial Volume Interpolation was introduced by Maes *et al.*[10], which increments several histogram bins for each sample based on the distance of the sample point from the surrounding pixels. Chen and Varshney extended this concept to Generalised Partial Volume Estimation, which uses extended spatial support [21].

The remainder of the paper is organised as follows. Section 2 reviews MI along with the four common sampling methods. After this, the first and second derivatives are derived in Section 3 and some analysis is performed in Section 4. Next, two example applications are discussed with a corresponding set of experiments in Section 5. The conclusion follows in Section 6.

2 Mutual Information

2.1 Registration

To start a brief formalisation of the registration process is required. Let f_R represent a reference image, and let f_T represent a template image. Both images are functions of position $\mathbf{x} \in \mathbb{R}^2$, although only trivial changes in the analysis below are required if the dimension of $\mathbf{x}$ is altered to represent volumetric data. Since f_R and f_T are represented as lattices of values at integral positions for $\mathbf{x}$, interpolation is used to obtain values at non-integral positions. There is insufficient space to discuss the choice of an interpolation method, but this is an important design issue. The interested reader is referred to a survey by Amidror [22].

For convenience and computational efficiency f_R is treated as infinite in extent and sampling is only performed within bounds of the lattice of f_T. Regions outside of the defined lattice of f_R are defined as 0. Hence f_T is considered constant with respect to any warp, and expensive boundary checking is avoided.

The registration process aims to align f_R and f_T, by minimising a distance function D for some warp function $\mathbf{w}$ with parameters $\mathbf{v}$: $\mathbf{v}_{reg} = \arg_{\mathbf{v}} \min D[f_R(\mathbf{x}), f_T(\mathbf{w}(\mathbf{x}, \mathbf{v}))]$. For computational reasons (because f_T is usually a smaller data set than f_R) it is easier to reformulate the problem as one of applying an inverse warp. Also, the function being minimised is MI, denoted by convention as I:

$$\mathbf{v}_{reg} = \arg_{\mathbf{v}} \min -I[f_R(\mathbf{w}^{-1}(\mathbf{x}, \mathbf{v})), f_T(\mathbf{x})]$$

To maintain notational clarity $\mathbf{w}^{-1}(\mathbf{x}, \mathbf{v})$ is referred to hereafter as $\mathbf{x}_{\mathbf{w}}$. The negative sign is required because MI has larger values for better matches, and we wish to maintain the convention of referring to function minimisation.

2.2 Histogram Estimation

A measure of the information mutual to f_T and the corresponding region in f_R is obtained from the joint intensity histogram $h(r, t, \mathbf{v})$ of the two images. Here $r \in [0; r_{mx}] \in \mathbb{Z}$ and $t \in [0; t_{mx}] \in \mathbb{Z}$ index the intensities that f_R and f_T respectively consist of ($\mathbb{Z}$ is the set of integers). The histogram may be normalised to give an approximation of the probability distribution function (PDF) of intensities, i.e. $p(r, t, \mathbf{v}) = \frac{1}{N_{\mathbf{x}}} h(r, t, \mathbf{v})$, where $N_{\mathbf{x}}$ is the number of samples in the histogram. MI is defined here in terms of p rather than h for clarity, and the dependence on $\mathbf{v}$ is explicitly indicated:

$$I(\mathbf{v}) = \sum_{r,t} p_{rt}(r, t, \mathbf{v}) \log \left(\frac{p_{rt}(r, t, \mathbf{v})}{p_r(r, \mathbf{v}) p_t(t)} \right) \tag{1}$$

A more common form of (1) has three entropy terms: $I = H_r + H_t - H_{rt}$. These are exactly the same and the more condensed form above is used for conciseness.

The PDF's p_r and p_t are easily obtained from the joint PDF, since $p_r = \sum_t p_{rt}$ and $p_t = \sum_r p_{rt}$. Note the treatment of r and t as discrete variables (or

indices), indicating the finite bin-size of the histogram h from which p is derived. MI is *not* invariant to the bin-size Δi, which limits its bounds, as does the number of sample points: $I \leq \log(\min(\frac{r_{mx}}{\Delta i}, \frac{t_{mx}}{\Delta i}, k_{mx} N_{\mathbf{x}}))$, where k_{mx} indicates the number of histogram bins populated per sample. The joint histogram is defined in terms of two window functions $\psi()$, which act as membership functions:

$$h(r, t, \mathbf{v}) = \sum_{\mathbf{x}} \psi\left(r - \frac{f_R(\mathbf{x_w})}{\Delta i}\right) \psi\left(t - \frac{f_T(\mathbf{x})}{\Delta i}\right) \tag{2}$$

Where each sample taken from f_R and f_T is added to one histogram bin:

$$\psi(\epsilon) = \beta_0^-(\epsilon) = \begin{cases} 1 & 0 < \epsilon < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This kind of sampling is referred to as *standard sampling*. The $\beta()$ function in the above equation comes from the B-spline family of functions, and a brief digression describing these is now made.

2.3 B-splines

B-spline functions are a family of functions with several useful properties, a brief description of which is given here. A more detailed description of B-spline functions and their numerical computation is given by Unser *et al.*in [23]. Firstly, the sum of a B-spline function for all integral distances from a real value is one, i.e. it has a portion of unity. This means that no renormalisation is required when histogramming. Secondly, the integral of a B-spline is one. Thirdly, order n B-splines are the convolution of any set of B-splines whose order sums to n. Lastly, the derivative of an order n B-spline is a function of two order $n-1$ B-splines. These properties are summarised below.

$$\begin{aligned} \sum_{a \in \mathbf{Z}} \beta(\epsilon + a) &= 1 \qquad \epsilon \in \mathbb{R} \\ \int_{\epsilon \in \mathbf{R}} \beta(\epsilon) &= 1 \\ \beta_n(\epsilon) &= \beta_{n-1}(\epsilon) * \beta_0(\epsilon) \\ \frac{\partial \beta_n}{\partial \epsilon} &= \beta_{n-1}(\epsilon + \frac{1}{2}) - \beta_{n-1}(\epsilon - \frac{1}{2}) \end{aligned}$$

The $0th$ order B-spline β_0 is simply a top hat function, centred about 0, i.e. $\beta_0(\epsilon) = 1$ when $|\epsilon| \leq \frac{1}{2}$ and 0 otherwise. We also define offset top-hat functions $\beta_0^- = \beta_0(\epsilon - \frac{1}{2})$ and $\beta_0^+ = \beta_0(\epsilon + \frac{1}{2})$.

2.4 Different Sampling Methods

For Standard Sampling (STD) we can see from (2) and (3) that for each of the $N_{\mathbf{x}}$ lattice points in f_T, each histogram in h_{rt} is incremented once. For reference

the windowing function for STD is restated here in (4a), where f is an image, i is an intensity index, Δi is the bin size of the histogram, and $\mathbf{x}$ is a sample point. An explanation of each functions below follows.

$$\psi_{std}(i - f(\mathbf{x})) = \beta_0^-(i - \frac{f(\mathbf{x})}{\Delta i}) \tag{4a}$$

$$\psi_{pve}^{(n)}(i) = \sum_{\mathbf{y}\in\mathbb{Z}^2} \beta_n(\mathbf{x} - \mathbf{y})\beta_0^-(i - \frac{f(\mathbf{y})}{\Delta i}) \tag{4b}$$

$$\psi_{ppz}^{(n)}(i) = \beta_n^-(i - \frac{\text{floor}(f(\mathbf{x}))}{\Delta i}) \tag{4c}$$

$$\psi_{ipz}^{(n)}(i) = \beta_n^-(i - \frac{f(\mathbf{x})}{\Delta i}) \tag{4d}$$

Partial Volume Estimation (PVE), introduced by Maes as Partial Volume Interpolation in [10], aimed at making shifts between histogram bins smooth as the parameters of $\mathbf{v}$ varied. PVE has the added advantage of not adding any (possibly false) information other than the given data. The method involves populating the intensity histogram bins of the four lattice points surrounding each sample by a weighted amount. The weighting is proportional to the area of overlap between the square regions around the sample and lattice points. This is equivalent to integrating the region of each intensity for nearest neighbour interpolation: i.e. $h(i) = \int_{\mathbf{x}} \beta_0^-(i - f_{nn}(\mathbf{x}))$. Chen and Varshney extended partial volume interpolation to generalised Partial Volume Estimation (PVE) by using higher order B-splines to weight a larger region of pixels [21]. Although PVE has been treated as alternative *interpolation* methods, strictly speaking they are alternative *sampling* methods. Hence the term "PVE" being used.

The windowing function for PVE is given in (4b), where $\mathbf{y}$ are the coordinates of all lattice points in the image and n indicates the order in the sampling family. Note that (4b) collapses to (4a) with nearest neighbour interpolation for $n = 0$, since in that case only one valid value for $\mathbf{y}$ exists, that of the lattice point nearest to $\mathbf{x}$. For notational clarity, the first window function in (4b) is shown to take a vector as an input. This is simply a product of two window functions, one for each dimension of the vector, i.e. $\beta_n(\mathbf{x}) = \prod_{k=1}^{K} \beta_n(x_k)$, where K is the number of components in the vector $\mathbf{x}$.

The advantages of PVE are that it does not add information not explicitly given in the image, it is relatively inexpensive and has a smooth surface. The disadvantage is that for orders below 2, PVE is only C_1 smooth with cusps at points of $\mathbf{v}$ where grid-alignment between f_t and f_r occur. A strong bias towards these cusped positions exists. Also, a nearest neighbour model of the world ignores much of the information implicit to the image.

Two types of Parzen windowing routines exist: *Post-Parzen* Windowing (PPZ) and *In-Parzen* windowing (IPZ). In PPZ, the histogram is constructed before convolution with a Parzen window. In IPZ, each sample is convolved during histogram construction. This takes advantage of the information sample's intensity value before the information loss implicit to discretisation occurs.

The window equation for post-Parzen windowing is given in (4c), where floor($f(\mathbf{x})$) indicates reduction to the first integer value below $f(\mathbf{x})$. (4c) shows an nth order B-spline window function. In fact any window function may be used. B-splines were used here since they are inexpensive and their derivatives are easily obtainable [23]. The advantages of post-Parzen windowing are that it improves on the basic sampling method using a computationally cheap operation: $O(i_{\max}^2 w^2)$. However, there is information loss due to blurring of the histogram, and the function is not necessarily smooth. In-Parzen windowing differs slightly, in that it lacks the implicit discretisation of intensity values as shown in (4d). As a result, In-Parzen windowing has a guaranteed C_{n-1} smooth cost function surface and a more accurate histogram. Again some information loss occurs due to blurring of the histogram, and the method is comparatively expensive. Both (4c) and (4d) collapse to (4a) for $n = 0$.

3 Jacobians and Hessians

The Jacobian of MI may now be found by applying the product and chain rules to (1) and collecting the terms:

$$\frac{\partial I}{\partial \mathbf{v}} = \sum_{r,t} \frac{\partial p_{rt}}{\partial \mathbf{v}} \left(1 + \log\left(\frac{p_{rt}}{p_r}\right) - \log(p_t)\right) - \frac{p_{rt}}{p_r}\frac{\partial p_r}{\partial \mathbf{v}}$$

A more general definition of the above equation has been given by Thevenaz [1], where a non-constant $N_{\mathbf{x}}$ was accounted for. However their approach constructs the problem such that $N_{\mathbf{x}}$ is constant anyway, and making this assumption early on simplifies the following derivation considerably.

The summations in the fourth (last) term may be split to give $\sum_r \frac{1}{p_r} p'_r \cdot \sum_t p_{rt}$, since p_r and p'_r are not dependent on t. However $\sum_t p_{rt} = p_r$ since it is a sum of a joint histogram. So the fourth term becomes $\sum_r p'_r$, because p_r^{-1} and p_r cancel. However, because $N_{\mathbf{x}}$ is constant and p is based on a histogram, $\sum p$ always equals one, and therefore $\sum p'$ always equals zero, so this term disappears.

Also if the third term ($\sum_{r,t} p'_{rt} \log(p_t)$) is separated out, the summations may again be split to get $\sum_t \log(p_t) \sum_r p'_{rt}$. But $\sum_r p'_{rt} = p'_t$, which is zero as the template is constant. So the third term also disappears.

The remaining two terms are combined and the derivative of MI becomes:

$$\frac{\partial I}{\partial \mathbf{v}} = \sum_{r,t} \frac{\partial p_{rt}}{\partial \mathbf{v}} \log\left(\frac{e p_{rt}}{p_r}\right) \tag{5}$$

3.1 Derivative of histogram function

The derivative of the histogram function may be obtained using the chain rule:

$$\frac{\partial p_{rt}}{\partial \mathbf{v}} = \frac{\partial}{\partial \mathbf{v}} \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \psi[t - \frac{f_T(\mathbf{x})}{\Delta i}]\psi[r - \frac{f_R(\mathbf{x_w})}{\Delta i}] = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \psi_T[t - \frac{f_T(\mathbf{x})}{\Delta i}] \frac{\partial}{\partial \mathbf{v}} \psi_R[r - \frac{f_R(\mathbf{x_w})}{\Delta i}]$$

$$= \frac{1}{N_{\mathbf{x}} \Delta i} \sum_{\mathbf{x}} \psi_T \frac{\partial \psi_R}{\partial \epsilon} \frac{\partial \epsilon}{\partial f_R} \frac{\partial f_R}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{v}} = -\frac{1}{N_{\mathbf{x}} \Delta i} \sum_{\mathbf{x}} \psi_T \frac{\partial \psi_R}{\partial \epsilon} \nabla f_R \frac{\partial \mathbf{w}}{\partial \mathbf{v}}$$

The derivatives for the reference window functions differ for each sampling method. Here the intensities are indicated by r since only derivative for the reference image is required:

$$\frac{\partial \psi_{std}}{\partial \epsilon}(r) = \delta(r - \frac{f_R(\mathbf{x_w})}{\Delta i}) - \delta(r - 1 - \frac{f_R(\mathbf{x_w})}{\Delta i}) \tag{6a}$$

$$\frac{\partial \psi^{(n)}_{pve}}{\partial \epsilon}(r) = \sum_{\mathbf{y}} \left(\beta^+_{n-1}(\mathbf{x_w} - \mathbf{y}) - \beta^-_{n-1}(\mathbf{x_w} - \mathbf{y})\right) \beta^-_0(r - \frac{f_R(\mathbf{y})}{\Delta i}) \tag{6b}$$

$$\frac{\partial \psi^{(n)}_{ppz}}{\partial \epsilon}(r) = \left(\beta^+_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i}) - \beta^-_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i})\right) \sum_{m \in \mathbb{Z}} \delta(r - m) \tag{6c}$$

$$\frac{\partial \psi^{(n)}_{ipz}}{\partial \epsilon}(r) = \left(\beta^+_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i}) - \beta^-_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i})\right) \tag{6d}$$

It should also be noted that for PVE the ∇f factor should be removed, since ψ_{pve} does not depend on $f_R(\mathbf{x_w})$, but on $\mathbf{x_w}$. Apart from this difference, note how similar the structure of all these equations are, showing their relationship. Note also how the δ functions in $\partial_\epsilon \psi_{std}$ and $\partial_\epsilon \psi_{ppz}$ imply that the gradient is constant, except at certain $\mathbf{v}$ positions on the cost function surface where a step change occurs. This exactly mirrors reality.

In these cases (STD and PPZ) the derivative function surface is a zero plane populated by impulse functions, the analytic derivative supplies almost no information to the optimisation function and convergence will fail. Hence it is better to use the approximate derivative $\frac{\partial \psi_R}{\partial \epsilon} \approx \frac{\Delta \psi_R}{\Delta \epsilon}$:

$$\frac{\partial \psi_{std}}{\partial \epsilon}(r) \approx \beta^-_0(r - \frac{f_R(\mathbf{x_w})}{\Delta i}) - \beta^-_0(r - 1 - \frac{f_R(\mathbf{x_w})}{\Delta i}) \tag{7a}$$

$$\frac{\partial \psi^{(n)}_{ppz}}{\partial \epsilon}(r) \approx \left(\beta^+_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i}) - \beta^-_{n-1}(r - \frac{f_R(\mathbf{x_w})}{\Delta i})\right) \sum_{m \in \mathbb{Z}} \beta^-_0(r-m) - \beta^-_0(r-m-1) \tag{7b}$$

3.2 MI Hessian

The MI Hessian is approximated to:

$$\begin{aligned} \frac{\partial I^2}{\partial v_1 \partial v_2} &= \sum_{r,t} \left(\frac{\partial p_{rt}}{\partial v_1} \frac{\partial p_{rt}}{\partial v_2} \frac{1}{p_{rt}} - \frac{\partial p_r}{\partial v_2} \frac{\partial p_{rt}}{\partial v_1} \frac{1}{p_r} + \frac{\partial p^2_{rt}}{\partial v_1 \partial v_2} \log\left(\frac{e p_{rt}}{p_r}\right) \right) \\ &= \sum_{r,t} \left(\frac{\partial p_{rt}}{\partial v_1} \frac{\partial p_{rt}}{\partial v_2} \left(\frac{1}{p_{rt}} - \frac{1}{p_r} \right) \right) \end{aligned} \tag{8}$$

because in the second term $\sum_t \frac{\partial p_{rt}}{\partial v_2} \frac{\partial p_r}{\partial v_1} \frac{1}{p_r} = \sum_t \frac{\partial p_{rt}}{\partial v_2} \frac{\partial p_{rt}}{\partial v_1} \frac{1}{p_r}$. The third term is approximately zero near the minimum. Its use only improves the speed of optimisation slightly at great computational expense.

3.3 Warp functions, their Jacobians and Hessians

Here we consider four types of warp functions $\mathbf{w}(\mathbf{x}, \mathbf{v})$: translation, Euclidean, similarity and affine. The equations for these warps are:

$$\mathbf{w}_{tx}(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} x + v_1 \\ y + v_2 \end{pmatrix} \qquad \mathbf{w}_{eu}(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} +x\cos v_3 + y\sin v_3 + v_1 \\ -x\sin v_3 + y\cos v_3 + v_2 \end{pmatrix}$$

$$\mathbf{w}_{af}(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} xv_1 + yv_3 + v_5 \\ xv_2 + yv_4 + v_6 \end{pmatrix} \quad \mathbf{w}_{si}(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} +xv_4\cos(v_3) + yv_4\sin(v_3) + v_1 \\ -xv_4\sin(v_3) + yv_4\cos(v_3) + v_2 \end{pmatrix}$$

The Jacobians of each of these warps are:

$$\begin{aligned}
\nabla f \frac{\partial \mathbf{w}_{tx}}{\partial \mathbf{v}} &= \begin{pmatrix} f_x & f_y \end{pmatrix} \\
\nabla f \frac{\partial \mathbf{w}_{eu}}{\partial \mathbf{v}} &= \begin{pmatrix} f_x & f_y & (f_x\, f_y) R'(x\, y)^T \end{pmatrix} \\
\nabla f \frac{\partial \mathbf{w}_{si}}{\partial \mathbf{v}} &= \begin{pmatrix} f_x & f_y & (f_x\, f_y) v_4 R'(x\, y)^T & (f_x\, f_y) R(x\, y)^T \end{pmatrix} \\
\nabla f \frac{\partial \mathbf{w}_{af}}{\partial \mathbf{v}} &= \begin{pmatrix} f_x x & f_y x & f_x y & f_y y & f_x & f_y \end{pmatrix}
\end{aligned}$$

where R is the standard rotation matrix. The Hessians for these warps are trivial to derive and are not shown here. Hereafter, warps are sometimes referred to by their Degrees of Freedom (DoF): e.g. 3DoF warp instead of Euclidean warp.

4 Analysis

4.1 Computational Costs

The computational costs of sampling methods are important when selecting which one to use for a particular application. Also, MI is sometimes regarded as an expensive option compared to say sum of square differences (SSD) or normalised correlation. This subsection shows that this is not necessarily true.

The SSD operation is $O(N_{\mathbf{x}})$: each operation requiring a warp, a template pixel access and multiple reference pixel accesses for interpolation. For MI, the only additional cost is to access each histogram bin after constructing it, i.e. MI is $O(N_{\mathbf{x}} + t_{mx} r_{mx})$. More sophisticated MI methods require multiple bin updates per sample, which can also increase computational cost. Theoretical estimates of costs for each function are given in Table 4.1.

The costs of calculating the Jacobian would appear to be substantially higher, since one histogram per warp parameter is required. However, there is some redundancy between the gradient and function evaluations, so this increase is not substantial. Likewise for the Hessian.

Some empirical tests were performed to verify the predictions of computational cost, the results of which are given in Fig. 1. As expected there is some overhead to the functions, which is indicated by an initial decrease in the cost per sample versus the number of samples before a steady state is reached.

Table 1. Computational complexity of various similarity methods.

Function	Order	Interp. Method	Reads of f_t+f_r	Writes updates	Ancillary
SSD	n/a	NNI (Nearest Neighbour)	$N_{\mathbf{x}}(1+1)$	$N_{\mathbf{x}}$	
SSD	n/a	BLI (Bi-Linear)	$N_{\mathbf{x}}(1+4)$	$N_{\mathbf{x}}$	
SSD	n/a	BCI (Bi-Cubic)	$N_{\mathbf{x}}(1+16)$	$N_{\mathbf{x}}$	
MI(std)	n/a	NNI	$N_{\mathbf{x}}(1+1)$	$N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(std)	n/a	BLI	$N_{\mathbf{x}}(1+4)$	$N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(std)	n/a	BCI	$N_{\mathbf{x}}(1+16)$	$N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(pve)	1st	n/a	$N_{\mathbf{x}}(1+4)$	$4N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(pve)	2nd	n/a	$N_{\mathbf{x}}(1+9)$	$9N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(pve)	3rd	n/a	$N_{\mathbf{x}}(1+16)$	$16N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(ipz)	1st	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$4N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(ipz)	2nd	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$9N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(ipz)	3rd	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$16N_{\mathbf{x}}$	$t_{mx}r_{mx}$
MI(ppz)	1st	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$N_{\mathbf{x}}$	$4t_{mx}r_{mx}$
MI(ppz)	2nd	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$N_{\mathbf{x}}$	$9t_{mx}r_{mx}$
MI(ppz)	3rd	BLI,BCI	$N_{\mathbf{x}}(1+(4,16))$	$N_{\mathbf{x}}$	$16t_{mx}r_{mx}$

4.2 Artefacts

Although artefacts in the cost function surface of MI are beyond the scope of this paper, a brief mention is necessary since they can affect convergence of an optimisation algorithm to the correct minimum. The use of interpolation results in the appearance of artefacts in the cost function surface. Artefacts occur for all similarity functions, not just MI and there are two types of artefact, named for their appearance: *hiss* and periodic *glitches*.

Hiss appears as random high frequency shifts in the cost function surface. These random shifts are generally small compared to the overall value at a each position. The cause of hiss is non-linearities in the function, which cause discrete shifts as the warp parameters $\mathbf{v}$ vary. This behaviour is essentially random, since it depends on the numerous local shifts in value for each sample point.

Glitches are a periodic pattern in the cost function surface. They have a larger amplitude than hiss, although this is generally still smaller than the signal value. Glitches also have the more insidious effect of shifting global maxima to new positions or *bias*. Glitches are generally caused by a combination of synchronisation of sample positions in the reference and template image combined with biases caused by local correlations in the two sets of data. An example of this is the cusped pattern seen for first order PVE.

Of the MI families discussed, STD and PPZ are particularly prone to hiss due to their implicit non-linear floor functions. This is less of a problem than it might seem, since optimisation functions sample the cost function surface quite sparsely and the local trends in surface are not strongly affected by hiss. STD and first order PVE are somewhat prone to bias due to glitches [2], which can be more serious. The effect of glitches is seen in some results, but further discussion is not possible here due to space constraints.

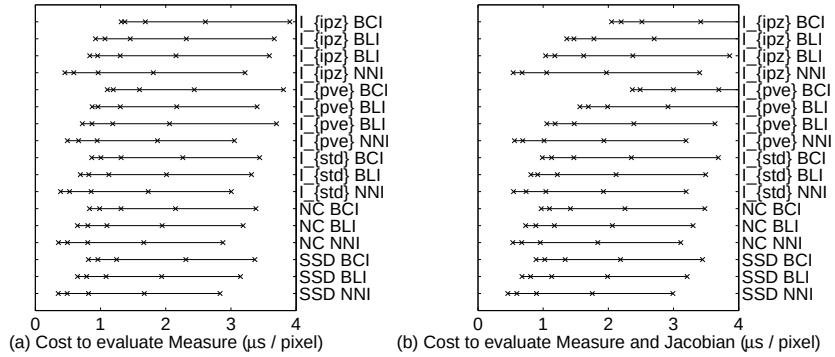

Fig. 1. Computational cost when evaluating (a) similarity functions and (b) their Jacobians as well. Efficiency increases with template size, reaching the minimum shown.

5 Experiments

A series of experiments was performed to evaluate the ability of the MI families presented here to converge to a ground truth position. To provide a baseline measurement the same set of experiments was performed for sum of squared differences (SSD) and normalised correlation (NC). In all 13 functions were compared: SSD at 3 sample rates, NC, I_{std} at 3 sample rates, $I^{(o)}_{pve}$ (o=1,2,3), $I^{(o)}_{ipz}$ (o=1,2,3), where o denotes B-spline order. In all cases bi-linear interpolation was used. Similar results were obtained for bi-cubic interpolation, so these results are not shown. In general 1 sample/pixel was made. For SSD and I_{std} rates of 2 and 3 samples per pixel were tested as well to see if an increase to the equivalent computational expense of PVE and IPZ would give comparable results. PPZ was not tested because at the time of writing it was not yet implemented.

Eight reference and template image pairs were used to cover a variety of applications and not bias towards any particular method. Data-set 1 (**Brain**) used two simulated images of the same brain obtained using two different processes. The template was fairly large (71x89) and the intensities have different underlying functions. In addition the reference image was rotated by 5° and up-scaled by 3%. Similarity warps were allowed (i.e. 4 degrees of freedom DoF). Data-set 2 (**Satellite**) is an overhead image of an airport obtained from Google-earth^{TM}. The template was extracted directly from the image and is 41x41 pixels. Affine warps were allowed (i.e. 6DoF). Data-set 3 (**Hyena**) was taken from a noisy infra-red image of a Hyena. The image was shrunk by 75% without smoothing. The template was offset such that the ground truth is 0.25 off grid alignment. The template was also 41x41 pixels. Euclidean Warps were used for registration (3Dof). Data-set 4 (**Walk**) was extracted from a video supplied by the CAVIAR project (`http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`) of Fisher *et al.*. There are 15 frames between the image and template, and the relationship between intensities is highly non-linear. The template was 19x37 pixels in size and 2DoF were used.

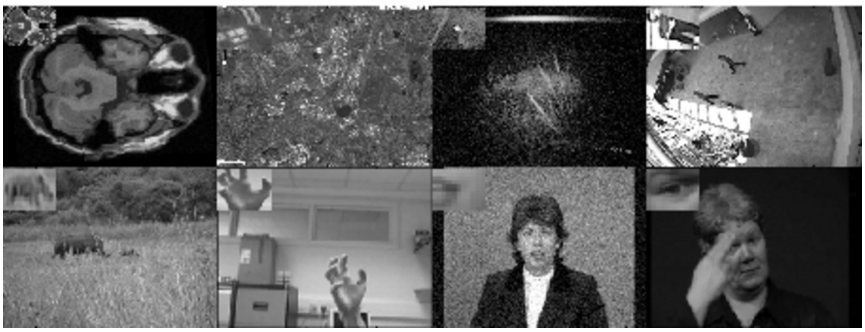

Fig. 2. The eight data sets used for testing the similarity metrics (numbered left to right top to bottom), with corresponding templates in the upper left corners.

Data set 5 (**Rhino**) a baby rhinoceros is extracted 10 frames before the frame it is registered to. The baby rhino is occluded in many places by grass in the foreground, and the sequence is particularly noisy as it was taken at low resolution and highly compressed. The template was 17x33 pixels and 2DoF were used. Data-set 6 (**Hand**) was of a hand that changes in shape and in intensity over a large region of the template. The template was 33x33 pixels in size and 2DoF were used. Data-set 7 (**Claire**) was extracted from a clean motion sequence of a newscaster. The template was 11x11 and extracted from the preceeding frame. The 5% Gaussian noise was added to the image. 2DoF were used. Data-set 8 (**Sign**) was extracted from a sequence of a lady communicating using sign. Six frames separated the image and template and the (her) right eye was used as a feature. This data-set is notable for the large amount of occlusion. The template was 17x17 pixels in size and 2DoF were used.

These images were chosen for the large amounts of noise (Claire, Hyena), large occlusions (Sign, Rhino), nearby distractors (Satellite), and highly nonlinear relationships in intensity or structural variations (Walk, Brain, Hand). The data sets used are shown in Fig. 2.

Table 5 shows the mean error (μ), standard deviation (σ), and number of convergences to within 10% of the lower template dimension (N) for each data set and each similarity metric. In these tests, only the (x, y) positions were considered, since the other warp parameters are small compared to the (x, y) position. For the test-set, the ground truth was obtained using a brute force search of the cost function surface to an accuracy of 0.01 pixels. One thousand positions were randomly chosen uniformly from a region surrounding the ground truth. The region on each side of the ground truth was 30% of the minor template dimension, and where respectively relevant for rotation, scale and affine parameters: 15°, 10% and 0.1. Due to space constraints, only results using Levenberg-Marquardt are shown, but the other optimisation methods gave comparable results.

Since MI makes no assumptions about the template and reference intensities we expected it to perform somewhat better than SSD for many of the data-sets. Particularly in the Hyena, Claire and Hand data-sets, SSD proved remarkably

Table 2. Convergence to best match for 8 data sets and 13 similarity measures.

Measure		SSD	SSD	SSD	NC	I_{std}	I_{std}	I_{std}	$I^{(1)}_{pve}$	$I^{(2)}_{pve}$	$I^{(3)}_{pve}$	$I^{(1)}_{ipz}$	$I^{(2)}_{ipz}$	$I^{(3)}_{ipz}$
Sample Rate		1	0.5	0.33	1	1	0.5	0.33	1	1	1	1	1	1
1 Brain	μ	14.43	13.21	12.81	n/a	15.08	14.91	14.60	14.82	14.36	14.08	15.02	14.89	14.79
	σ	5.67	5.52	5.41	n/a	5.89	6.18	6.57	6.27	6.78	6.99	6.02	6.18	6.29
	N	0	0	0	0	7	15	26	22	42	49	13	16	21
2 Satellite	μ	13.83	8.73	7.41	9.16	8.56	6.71	5.91	6.87	5.11	4.78	7.69	6.70	6.28
	σ	162.48	28.52	15.16	3.51	4.35	5.65	6.04	5.56	6.12	6.23	5.12	5.69	5.90
	N	211	279	307	008	132	358	462	355	566	607	249	362	413
3 Hyena	μ	8.69	7.80	7.45	9.17	9.14	9.06	8.87	8.92	8.47	8.12	9.13	9.07	9.02
	σ	3.91	4.27	4.37	3.52	3.54	3.59	3.68	3.73	4.03	4.16	3.54	3.56	3.58
	N	57	119	146	8	10	11	16	25	53	68	8	8	15
4 Walk	μ	4.22	4.37	4.03	3.79	3.77	3.71	3.50	3.73	2.98	2.78	3.72	3.63	3.59
	σ	12.12	3.29	2.74	1.44	1.45	1.50	1.66	1.53	2.54	2.79	1.49	1.54	1.59
	N	94	146	129	37	41	56	114	37	379	450	50	66	69
5 Rhino	μ	2.49	2.40	2.22	3.77	2.93	1.84	0.81	2.80	0.62	0.60	2.56	2.39	2.32
	σ	1.42	1.32	1.14	1.37	1.73	1.74	1.42	1.84	1.38	1.42	1.56	1.60	1.70
	N	247	195	185	29	242	514	890	347	960	972	275	300	335
6 Hand	μ	7.43	6.71	6.14	7.76	7.65	7.22	7.15	7.41	6.86	6.39	7.41	7.19	7.07
	σ	3.25	4.34	4.82	2.75	2.91	3.60	3.88	3.35	4.79	5.06	3.25	3.63	3.72
	N	27	226	304	7	14	86	85	40	324	396	58	95	104
7 Claire	μ	2.17	2.10	1.74	2.33	2.26	2.06	2.03	2.31	2.34	2.41	2.17	2.11	2.03
	σ	0.93	0.91	1.01	0.85	0.96	1.28	1.49	1.04	2.08	2.13	1.09	1.21	1.27
	N	135	169	341	75	121	303	368	154	470	461	204	249	308
8 Sign	μ	7.29	7.46	7.41	3.76	4.37	5.02	5.08	4.14	5.60	5.53	4.92	5.30	5.52
	σ	0.66	0.70	0.65	1.43	1.44	1.57	1.50	1.47	1.75	1.75	1.51	1.56	1.60
	N	0	0	0	33	16	5	4	20	0	0	6	4	1

tolerant of noise and structural changes. Predictably, increasing the sampling rate only improved the results where there was high level detail or large amounts of noise. Normalised Correlation was generally the worst performer, except in the Sign and Walk sequences, where its tolerance of non-linear intensity relationships gave it an edge over SSD.

I_{std} performed better than SSD in about half the cases: where intensity relationships were highly non-linear. This could be due to a generally narrower basin of convergence than SSD and large amounts of hiss in the function surface of MI. Increasing the sampling rate usually improved performance substantially, since this decreases the amount of hiss in the surface. The exception was the Sign data-set, where the large occlusion created a large basin of convergence nearby.

Overall I_{pve} was the best performer when the order was above 2. Order 1 I_{pve} does not perform well due to the large number of glitches which often create local minima at points of grid alignment. This good performance was particularly noticeable in the Satellite, Walk, Hand and Rhino data sets, which either had much high frequency information or non-linear intensity relationships. This is probably due to the smooth function surface and wide but steep basin of convergence that PVE exhibits.

Surprisingly in most cases I_{ipz}, only outperformed I_{std} when the sample-rate was 1/pixel. Considering that sampling at 2 samples/pixel is equivalent in cost to first order IPZ, computational cycles would generally be better spent on using I_{pve} at higher sample rates for I_{std}.

In summary, in cases where the images have few occlusions, and lighting conditions do not change rapidly, SSD probably gives the best results per computational unit. SSD also proved surprisingly resilient to noise. Where lighting conditions vary and occlusions occur either I_{pve} (for $n \geq 2$) or I_{std} would be the methods of choice. A choice between these is difficult since the a higher sampling rate is necessary for I_{std} to work as well as I_{pve}, so the computational saving is not great. It is possible that I_{std} may outperform I_{pve} where the scales of the image and template are very different. This is left for future work.

6 Conclusion

This paper has introduced a single framework for the four main families of MI, namely: Standard sampling, Partial Volume Estimation, In-Parzen Windowing and Post Parzen Windowing. The analytic Jacobians and Hessians of these methods were also derived. A computational cost analysis was performed, which shows that STD MI is not much more expensive to compute than Sum of Squared Differences. The implementation was used to test the convergence of various image metrics using the Levenberg-Marquardt Method on a diverse array of images.

Despite its simplicity, SSD is the method of choice where the image is not occluded and the intensities of the template and reference image are linearly related. Where this does not occur I_{pve} (for $n \leq 2$) or I_{std} would be recommended.

Similarity functions (with Jacobians and Hessians) have been implemented in C++ for SSD, normalised correlation, and Mutual Information using standard sampling, partial volume estimation and in-Parzen windowing. The binaries and scripts used for testing are available online at the authors'URL.

Acknowledgements

Financial support from the CVSSP at Surrey University, the Department of Education and Skills, UK, and the EU FP6 Project "COSPAL", IST-2003-2.3.2.4, is gratefully acknowledged.

References

1. Thevenaz, P., Unser, M.: Optimization of mutual information for multi-resolution image registration. IEEE Trans. On Image Processing **9** (2000) 2083–2099
2. Pluim, J., Maintz, J., Viergever, M.: Interpolation artefacts in mutual information-based image registration. Computer Vision And Image Understanding **77** (2000) 211–232
3. Tsao, J.: Interpolation artifacts in multimodality image registration based on maximisation of mutual information. IEEE. Trans. Medical Imaging **22** (2003) 854–864

4. Maes, F.: Segmentation and registration of multimodal medical images: From theory, implementation and validation to a useful tool in clinical practice. PhD thesis, Dept. Elect. Eng. (ESAT/PSI), KU Leuven, Leuven, Belgium (1998)
5. Viola, P., Wells, W.: Alignment by maximization of mutual information. In: Proc. Int'l Conf. on Computer Vision, Boston, MA, USA (1995) 16–23
6. Studholme, C., Hill, D., Hawkes, D.: Automated 3d registration of truncated mr and ct images of the head. In: Proc. British Machine Vision Conference. (1995) 27–36
7. Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., Marchal, G.: Automated Multi-modality image registration based on information theory. In: Information Processing in Medical Imaging. Kluwer Academic (1995) 263–374
8. Shannon, C.: A mathematical theory of communication. The Bell System Technical Journal **27** (1948) 379–423, 623–656
9. Meyer, C., Boes, J., Kim, B., Bland, R., Wahl, P., Zasadny, K., Kison, P., Koral, K., Frey, K.: Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin plate spline warped geometric deformations. Medical Image Analysis **1** (1997) 195–206
10. Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximisation of mutual information. IEEE. Trans. On Medical Imaging **16** (1997) 187–198
11. Wells, W.I., Viola, P., Atsumi, H., Nakajima, S., Kikinis, R.: Multi-modal volume registration by maximization of mutual information. Medical Image Analysis **1** (1996) 35–51
12. Studholme, C., Hill, D., Hawkes, D.: Automated 3-d registration of mr and ct images of the head. Medical Image Analysis **1** (1996) 163–175
13. Jenkinson, M., Smith, S.: A global optimisation method for robust affine registration of brain images. Medical Image Analysis **5** (2001) 143–156
14. Pluim, J., Maintz, J., Viergever, M.: Mutual-information-based registration of medical images: A survey. IEEE Trans. Medical Imaging **22** (2003) 986–1003
15. Maes, F., Vandermeulen, D., Seutens, P.: Comparative evaluation of multiresolution optimization strategies for multimodalitiy image registration by maximization of mutual information. Medical Image Analysis **3** (1999) 272–286
16. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. 2nd edn. Cambridge University Press (1992)
17. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics **11** (1963) 431–441
18. Viola, P., Wells, W.: Alignment by maximization of mutual information. Int'l Journal of Computer Vision **24** (1997) 137–154
19. West, J., Fitzpatrick, J.M., et.al., M.Y.W.: Comparison and evaluation of retrospective intermodality brain image registration techniques. J. Comput. Assisted Tomography **21** (1997) 554–566
20. Parzen, E.: On estimation of a probability density function and mode. The Annals of Mathematical Statistics **33** (1962) 1065–1076
21. Chen, H., Varshney, P.: Mutual information-based CT-MR brain image registration using generalised partial volume joint histogram estimation. IEEE. Trans. Medical Imaging **22** (2003) 1111–1119
22. Amidror, I.: Scattered data interpolation methods for electronic systems: A survey. Journal of Electronic Imaging **11** (2002) 157–176
23. Unser, M., Aldroubi, A., Eden, M.: B-spline signal processing: Part i–theory. IEEE. Trans. Signal Processing **41** (1993) 821–833