Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation

Nicholas Dowson, *Member*, *IEEE*, and
Richard Bowden, *Senior Member*, *IEEE*

Abstract—Mutual Information (MI) is popular for registration via function optimization. This work proposes an inverse compositional formulation of MI for Levenberg-Marquardt optimization. This yields a constant Hessian, which may be precomputed. Speed improvements of 15 percent were obtained, with convergence accuracies similar those of the standard formulation.

Index Terms—Mutual Information, registration, Newton optimization, tracking.

1 Introduction

An inverse compositional formulation for aligning a template and a reference image using *mutual information* is derived in this paper. The alignment or registration of a pair of images is an operation required in many applications such as image mosaicking [16], simultaneous localization and tracking [6], and multimodal image alignment [13]. In many applications, numerous registration operations are required. So, any improvements in the speed have a large effect on application performance as a whole.

Lucas and Kanade made one of the earliest practical attempts to efficiently align a template image to a reference image [9], by minimizing the Sum of Squared Difference similarity function. Processing was limited by using a Newton-Raphson method to traverse the space of warp parameters. In Newton-Raphson optimization, iterative parameter updates to alignment parameters are obtained by multiplying the Jacobian by the inverse Hessian of the similarity function. Lucas and Kanade mainly considered translations, but they demonstrated that any linear transformation could be used.

Later research considered more complex transforms and attempted to reformulate the similarity function, allowing precomputation of some terms. In particular, Hager and Belhumeur [8] proposed inverting the roles of the reference and template at a strategic point in the derivation, and Shum and Szeliski [16] constructed the warp as a composition of two nested warps. In a general treatise on Lucas-Kanade (LK) techniques [1], Baker and Matthews combined these methods to formulate the inverse-compositional method.

Sum of Squared Differences (SSD) has several advantages as a similarly function: it is fast, it is simple to implement, it has a wide basin of convergence (making convergence easy), its gradient is simple to derive, and it is well understood. SSD's disadvantages include limited robustness to noise and variations in lighting conditions. Its wide basin of convergence can also make the result ambiguous. However, tracking multiple features and the use of models of appearance and structure can significantly improve robustness [5].

Mutual Information (MI) is only slightly more expensive than SSD to compute and has several advantages. MI tolerates nonlinear relationships between the intensities in images and is robust to noise. MI has a sharp peak, giving a precise result. However, a starting point near to the solution is required. In the medical image registration field, MI is now widely used after its concurrent introduction and popularization by Viola and Wells [20], Studholme et al. [17], and Collignon et al. [4].

Numerous MI implementations exist [13], but few use an analytic derivative, limiting the optimization methods that may be used. The analytic derivative is difficult to obtain because of the nonlinear flooring functions implicit in the histogramming process used in calculating MI. Notable exceptions are an analytic derivative of MI using Partial Volume Interpolation by Maes et al. [10]; and a derivative for MI using B-spline Parzen windowing by Thevenaz and Unser [18]. More recently, a general derivation of the four common types of MI was published by Dowson and Bowden [7]. The availability of a general analytic derivative for MI allows its use in the so-called Lucas-Kanade (LK) framework. This has implications to applications in both the computer vision and medical imaging communities.

The contribution of this work is to develop an inverse compositional formulation for MI. This uses two techniques: First, the alignment function is recomposed as a function of a base warp and a warp variation and, second, the roles of template and reference image are inverted or exchanged. This is difficult in the case of MI because the template and reference values are not separable into two terms. But, with some limited assumptions of constancy, speed-ups are still obtainable while maintaining the same accuracy as the conventional forwards-additive approach. Brooks and Arbel have also explored reformulating functions [3]. However, they use a BFGS optimizer, a bracketing and line-minimization method. BFGS only requires a Jacobian to be supplied and iteratively constructs a Hessian during optimization. In contrast, our approach evaluates a Hessian directly and uses this in a Levenberg-Marquardt (LM) algorithm, a Newton-type method. A direct comparison would really be considering two optimization philosophies rather than two formulations and is hence beyond the scope of this work.

The remainder of the paper is arranged as follows: After a background to image alignment in Section 2, the inverse compositional formulation of MI is presented in Section 3. The derivation obtained is compared to existing methods in terms of convergence and speed in Section 4 before the conclusions are given in Section 5.

2 Background

To begin with a brief formalization of the registration process is required. Let I_r represent a reference image and let I_t represent a template image. The images are functions of 2D coordinate $\mathbf{x} \in \mathbb{R}^2$. Some trivial changes to the formalization allow volumetric data $\mathbb{R}^3$ to be represented as well. Since I_r and I_t are represented as lattices of values at integral positions for $\mathbf{x}$, interpolation is used to obtain values at nonintegral $\mathbf{x}$ values.

The registration process aims to locate the region in I_r that most resembles I_t by minimizing a distance function, f, which measures the similarity of the two regions. The position of I_t relative to I_r is specified by a warp function $\mathbf{w}$ with parameters $\mathbf{v}$.

$$\mathbf{v}_{reg} = \arg_{\mathbf{v}} \min f[I_r(\mathbf{w}(\mathbf{x}, \mathbf{v})), I_t(\mathbf{x})]. \tag{1}$$

The position of greatest similarity is found using an optimization method. f can be any similarity measure, e.g., SSD or MI. MI increases with greater similarity, so, to maintain convention, we minimize negative MI. For convenience and computational efficiency, I_r is treated as infinite in extent and sampling to measure f is always performed within bounds of the defined region of I_t. Regions outside the defined region of I_r are defined as 0. Hence, I_t is constant with respect to the warp parameters and computationally expensive boundary checking is avoided.

Many optimization algorithms exist, but LK methods use a particular group of these: The so-called Newton-type methods, i.e.,

- N. Dowson is with Siemens Molecular Imaging, 28-36 Hythe Bridge Str, Oxford, OX1 2EP UK. E-mail: nicholas.dowson@siemens.com.
- R. Bowden is with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH UK.
E-mail: r.bowden@surrey.ac.uk.

Manuscript received 4 Sept. 2006; revised 30 Apr. 2007; accepted 16 July 2007; published online 8 Aug. 2007.
Recommended for acceptance by S. Baker.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0640-0906.
Digital Object Identifier no. 10.1109/TPAMI.2007.70757.

0162-8828/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

Name	Update
(Full) Newton Descent	$\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} - H^{-1}G$
Quasi-Newton Descent	$\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} - \tilde{H}^{-1}G$
Steepest Descent	$\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} - \lambda G$
Levenberg Marquardt	$\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} - ((\mathbf{1} + \lambda \mathbf{I})\tilde{\mathbf{H}})^{-\mathbf{1}}\mathbf{G}$

Fig. 1. Updates for four Newton-type optimization Methods. ($\mathbf{1}$ is a matrix of ones.) Although not explicitly indicated, several λ values may be tested.

methods which assume locally parabolic topology and "jump" to the minimum using gradient information: $\mathbf{v}^{(k+1)} \leftarrow \mathbf{v}^{(k)} - H^{-1}(\mathbf{v}^{(k)}) G(\mathbf{v}^{(k)})$. Here, H is the Hessian of f, $\frac{\partial^2 f}{\partial \mathbf{v}^2}$, G is the Jacobian of f, $\frac{\partial f}{\partial \mathbf{v}}$, and k indexes the iteration number. Newton methods should be contrasted with methods that choose a direction, bracket the minimum, and minimize along the line using Brent's algorithm [2], e.g., Powell's Method or Variable Metric Methods [14]. Bracketing methods are more stable than Newton methods, but somewhat slower since more function evaluations are performed.

Minima in tracking and registration problems are frequently numerous and closely spaced, so the robustness of bracketing yields little advantage. Speed improvements, on the other hand, make multiple initializations practical, which can improve performance.

Generally, LK type methods apply Quasi-Newton optimization, i.e., an approximate Hessian, $\tilde{H}$, is used. In general, Newton and Quasi-Newton only perform well when near to the minimum. Steepest Descent methods, which ignore local curvature and instead multiply G by a scalar *step-size* value λ, perform better when further from the minimum. The Levenberg-Marquardt [11] method combines these two methods for optimal performance. A summary of these methods is supplied in Fig. 1.

2.1 Lucas-Kanade Framework

The Lucas-Kanade (LK) framework uses the sum of squared differences function in a forwards-additive formulation to use the terminology of [1]. In this formulation, a base warp, $\mathbf{v}$, and a warp variation, $\Delta\mathbf{v}$, are used together to parameterize the relative positions of I_r and I_t:

$$f_{SSD}(\mathbf{v} + \Delta\mathbf{v}) = \sum_{\mathbf{x}} [I_r(\mathbf{w}(\mathbf{x}, \mathbf{v} + \Delta\mathbf{v})) - I_t(\mathbf{x})]^2. \tag{2}$$

A first order Taylor expansion is applied to the function within the brackets (not to the function as a whole):

$$f_{SSD}(\mathbf{v} + \Delta\mathbf{v}) = \sum_{\mathbf{x}} \left[I_r(\mathbf{w}(\mathbf{x}, \mathbf{v})) + \nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \Delta\mathbf{v} - I_t(\mathbf{x}) \right]^2, \tag{3}$$

where ∇I_r is the gradient of the image I_r with respect to its coordinates. A partial derivative with respect to $\Delta\mathbf{v}$ is then obtained:

$$\frac{\partial f_{SSD}}{\partial \Delta\mathbf{v}} = 2 \sum_{\mathbf{x}} \left[\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \right]^T \left[I_r(\mathbf{w}(\mathbf{x})) + \nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \Delta\mathbf{v} - I_t(\mathbf{x}) \right]. \tag{4}$$

Assuming a locally parabolic shape and setting the gradient to zero gives a closed form solution for updating $\mathbf{v}$, which takes the form: $\Delta\mathbf{v} = \tilde{H}^{-1}G$, where

$$G = \sum_{\mathbf{x}} \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \right)^T (I_t(\mathbf{x}) - I_r(\mathbf{w}(\mathbf{x}, \mathbf{v}))), \tag{5}$$

$$\tilde{H} = \sum_{\mathbf{x}} \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \right)^T \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{v}} \right). \tag{6}$$

Of course, a true parabolic surface seldom occurs, so the warp parameter must be iteratively computed and updated until the variation in parameters or function values becomes sufficiently small. The computational cost of each update is $O(N_{\mathbf{x}} N_{\mathbf{v}})$ for G and $O(N_{\mathbf{x}} N_{\mathbf{v}}^2)$ for H, where $N_{\mathbf{x}}$ is the number of pixels and $N_{\mathbf{v}}$ is the number of warp components.

The Hessian is denoted with a tilde because of an early hidden approximation made in the Taylor expansion, which neglects some of the second order information. A full second order expansion applied to the the entire f_{SSD} function yields the full Hessian:

$$\begin{aligned} H = \sum_{\mathbf{x}} \Bigg\{ &\left[\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{x}} \right]^T \left[\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{x}} \right] + \\ &+ (I_r - I_t) \left[[\frac{\partial \mathbf{w}}{\partial \mathbf{v}}]^T (\nabla \cdot \nabla I_r) [\frac{\partial \mathbf{w}}{\partial \mathbf{v}}] + \nabla I_r \frac{\partial^2 \mathbf{w}}{\partial \mathbf{v}^2} \right] \Bigg\}. \end{aligned} \tag{7}$$

In a full Newton derivation, (7) would replace (6). Apart from the second term in H being computationally expensive to compute $O(16 N_{\mathbf{x}} N_{\mathbf{v}}^2)$, it is often marginal compared to the first term, especially near the minimum and has little effect on convergence.

2.2 The Inverse-Compositional Method

Baker and Matthews presented a reformulation of the SSD distance function and update method called the inverse compositional method in [1]. The warp function was recomposed as a function of two warps $\mathbf{w}(\mathbf{x}, \mathbf{v})$ and $\mathbf{w}(\mathbf{x}, \Delta\mathbf{v})$ with the roles of I_r and I_t inverted:

$$f_{SSD}(\mathbf{v}, \Delta\mathbf{v}) = \sum_{\mathbf{x}} (I_t(\mathbf{w}(\mathbf{x}, \Delta\mathbf{v})) - I_r(\mathbf{w}(\mathbf{x}, \mathbf{v})))^2. \tag{8}$$

Following the steps in Section 2.1, using this formulation yields the following approximation of the Hessian: $\tilde{H} = (\nabla I_t \frac{\partial \mathbf{w}}{\partial \mathbf{v}})^T (\nabla I_t \frac{\partial \mathbf{w}}{\partial \mathbf{v}})$. This depends solely on the template and is therefore constant with respect to $\mathbf{v}$. In other words, the Hessian may be precomputed, decreasing the overall complexity of each iterative update to $\mathbf{v}$ from $O(N_{\mathbf{x}} N_{\mathbf{v}}^2)$ to $O(N_{\mathbf{x}} N_{\mathbf{v}})$, reducing the time to register I_r and I_t.

3 Mutual Information in an LK Framework (MILK)

Mutual Information was originally presented by Shannon [15] as a measure of the information shared between two signals. This is calculated using the joint probability distribution function (PDF) of the intensities (amplitudes) of the two images (signals)

$$f_{MI} = \sum_{r,t} p_{rt}(r, t, \mathbf{v}) \log \left(\frac{p_{rt}(r, t, \mathbf{v})}{p_r(r, \mathbf{v}) p_t(t, \mathbf{v})} \right), \tag{9}$$

where $r \in [0; N_r - 1] \in \mathbb{Z}$ and $t \in [0; N_t - 1] \in \mathbb{Z}$ are, respectively, the range of allowed intensity values in I_r and I_t. The joint PDF is estimated from the joint histogram $p_{rt} = N_{\mathbf{x}}^{-1} h_{rt}$. The marginal probabilities are simply obtained by summing along one axis of the PDF, i.e., $p_r = \sum_t p_{rt}$ and $p_t = \sum_r p_{rt}$. As discussed in [7], several methods to measure MI exist, with the primary variation being how the image is sampled and the histogram is populated. But, in all cases, (9) is used.

The membership function of the histogram, ψ, illustrates the relationship between p and $\mathbf{v}$ more clearly than (9)

$$p_{rt}(r, t, \mathbf{v}) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \psi[r - I_r(\mathbf{w}(\mathbf{x}, \mathbf{v}))] \cdot \psi[t - I_t(\mathbf{x})]. \tag{10}$$

For this work, the in-Parzen windowing formulation of MI was used, where the window function is a B-spline: $\psi = \beta_n(\cdot)$. This formulation, originally proposed by Thevenaz and Unser [18], individually convolves each intensity sample with the Parzen window *before* the information loss associated with binning occurs. This is important because interpolated intensities can take noninteger values, the fractional part of which is usually thrown away. The result is a piecewise constant function as $\mathbf{v}$ varies, for which many bracketing and Newton-type optimization methods do not perform well.

Unlike the standard-sampling approach, where $\psi(\epsilon)$ is top-hat function, where ψ is 1 for $0 \leq \epsilon < 1$ and 0 otherwise, or the Post-Parzen windowing approach, where the histogram is convolved with a Parzen window after construction, the cost function for in-Parzen windowing is smooth. This improves convergence, especially at positions close to the global maximum. In this work, a third order B-spline window was used.

The Jacobian of MI is found by applying the product and chain rules and some simplifications [7] to obtain a Jacobian and a Hessian:

$$G = \sum_{r,t} \frac{\partial p_{rt}}{\partial \mathbf{v}} \log\left(\frac{p_{rt}}{p_r}\right), \tag{11}$$

$$H = \sum_{r,t} \left\{ \frac{\partial p_{rt}}{\partial \mathbf{v}}^T \frac{\partial p_{rt}}{\partial \mathbf{v}} \left(\frac{1}{p_{rt}} - \frac{1}{p_r}\right) + \frac{\partial^2 p_{rt}}{\partial \mathbf{v}^2} \log\left(\frac{p_{rt}}{p_r}\right) \right\}. \tag{12}$$

The derivatives of ψ are easily calculated from the calculus of B-splines [19] since $\partial_\epsilon B_n(\epsilon) = B_{n-1}(\epsilon + \frac{1}{2}) - B_{n-1}(\epsilon - \frac{1}{2})$. The second derivative is obtained in a similar manner.

The last term in (12) is usually neglected because it is expensive to obtain and does not affect convergence overly once the solution is near the local minimum. This is the analog of neglecting the second order terms for SSD in (7).

3.1 Inverse-Compositional MILK

The inverse compositional derivation for MI may now be obtained in the same manner as for SSD by splitting the warp into a function of two parameters, $\mathbf{v}$ and $\Delta\mathbf{v}$:

$$f_{MI}(\mathbf{v}, \Delta\mathbf{v}) = \sum_{r,t} p_{rt}(\mathbf{v}, \Delta\mathbf{v}) \log\left(\frac{p_{rt}(\mathbf{v}, \Delta\mathbf{v})}{p_r(\mathbf{v}) p_t(\Delta\mathbf{v})}\right). \tag{13}$$

Hereafter, to save space, the function parameters are not shown. Using the same approach as Section 2.2 for MI, the following gradient function is obtained:

$$\begin{aligned} G &= \sum_{r,t} \left\{ -\frac{p_{rt}}{p_t} \frac{\partial p_t}{\partial \Delta\mathbf{v}} + \log\left(\frac{p_{rt}}{p_t}\right) \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} \right\} \\ &= \sum_{r,t} \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} \log\left(\frac{p_{rt}}{p_t}\right). \end{aligned} \tag{14}$$

The first term in (14) cancels out, using reasoning similar to that used in [7] to obtain (11). In $\sum_{r,t} \frac{p_{rt}}{p_t} \frac{\partial p_t}{\partial \Delta\mathbf{v}}$, because p_t is independent of r, the summations may be separated to form $\sum_t \frac{1}{p_t} \frac{\partial p_t}{\partial \Delta\mathbf{v}} \cdot \sum_r p_{rt}$. But, $\sum_r p_{rt} = p_t$, which cancels with $\frac{1}{p_t}$, so the whole term becomes $\sum_t \frac{\partial p_t}{\partial \Delta\mathbf{v}}$. This summation is zero because the window function is a B-spline, with the property: $\sum_{\forall \epsilon \in \mathbb{Z}} \partial_\epsilon \psi(\epsilon) = \sum_{\forall \epsilon \in \mathbb{Z}} \partial_\epsilon (B_2(\epsilon + \frac{1}{2}) - B_2(\epsilon - \frac{1}{2})) = 0$. The same reasoning was also used to eliminate p_r in the log function parameter, although this is not shown in (14).

The Hessian is obtained using chain and product rules, before applying some simplification,

$$\begin{aligned} H = &\sum_{r,t} \left\{ -\frac{\partial p_t}{\partial \Delta\mathbf{v}}^T \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} \frac{1}{p_t} + \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}}^T \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} \frac{1}{p_{rt}} \right. \\ &\left. + \frac{\partial^2 p_{rt}}{\partial \Delta\mathbf{v}^2} \log\left(\frac{p_{rt}}{p_t}\right) \right\}. \end{aligned} \tag{15}$$

In the first term of (15), only one factor is dependent on r so it may be separated out to form: $\sum_t \frac{1}{p_t} \frac{\partial p_t}{\partial \Delta\mathbf{v}}^T \sum_r \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} = \sum_t \frac{1}{p_t} \frac{\partial p_t}{\partial \Delta\mathbf{v}}^T \frac{\partial p_t}{\partial \Delta\mathbf{v}}$ because r is independent of $\Delta\mathbf{v}$. Hence, the first term becomes dependent on t only.

$$\begin{aligned} H = &-\sum_t \frac{\partial p_t}{\partial \Delta\mathbf{v}}^T \frac{\partial p_t}{\partial \Delta\mathbf{v}} \frac{1}{p_t} + \sum_{r,t} \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}}^T \frac{\partial p_{rt}}{\partial \Delta\mathbf{v}} \frac{1}{p_{rt}} \\ &+ \sum_{r,t} \frac{\partial^2 p_{rt}}{\partial \Delta\mathbf{v}^2} \log\left(\frac{p_{rt}}{p_t}\right). \end{aligned} \tag{16}$$

In (16), there are three summations, the first of which is first order and independent of I_r. The second sum is first order but dependent on I_r and the third is second order.

The expensive second order term $(O(N_\mathbf{x} N_\mathbf{v}^2 + N_\mathbf{v}^2 N_t^2 N_r^2))$ can be neglected as it was for SSD and forwards-additive MI since it is marginal at positions close to the minimum. However, term two presents more of a problem since its dependence on I_r requires it to be recomputed every time $\mathbf{v}$ is updated. Its computational cost of $O(N_\mathbf{x} N_\mathbf{v} + N_\mathbf{v}^2 N_t^2 N_r^2)$ is significant relative to that of the first sum $(O(N_\mathbf{x} N_\mathbf{v} + N_\mathbf{v}^2 N_t^2))$, but this sum cannot simply be neglected since, of term one and term two, it forms a larger proportion of H.

Unlike SSD, in MI the influence of I_r and I_t cannot be wholly split into separate terms that are combined additively for any formulation of the MI function. Hence, for MI, H is always at least partially dependent on $\mathbf{v}$. MI is not the only such function. Normalized Correlation (NC) suffers from the same disadvantage since one of the components of NC is a sum of $I_r I_t$ products, which is not separable either.

This cost of reevaluating the second term in (16) every iteration is overcome by assuming its constancy anyway. This assumption is reasonable so long as the changes to $\mathbf{v}$ are small. Although, under large changes, this assumption becomes inaccurate, so does the assumption of local linearity made by the use of a first order Taylor expansion. Hence, $\tilde{H}$ may be treated as a precomputable constant. This yielded good results.

4 Experiments

Several experiments were undertaken to demonstrate the following:

- Inverse-compositional MI converges as frequently and in the same number of iterations as forwards-compositional MI.
- Due the precomputation of the Hessian, the computational cost of registration is lower for Inverse-compositional MI than for forwards-additive MI.

Experiments were performed using three image pairs for typical applications. These are shown in the first column of Fig. 2. The first pair of images are two slices taken from a simulated MRI of a human brain using T1 and PD modes. Because the brain was simulated, the ground truth is known exactly. The different modalities also present difficulties for similarity that assume a linear intensity relationship between image pairs. In the second image pair, the template was extracted directly from the reference image of a natural scene. Although this is somewhat artificial, it allows the ground truth to be exactly known. The third example is of an indoor scene where the lighting conditions have changed substantially. The images were hand registered using key-points at a high resolution $(2,560 \times 1,920)$ before being cropped and subsampled. Hence, the ground truth is known to within less than a pixel.

Simulated registrations using a six degree of freedom (DoF) affine warp from multiple initial starting points were performed. The initial positions were generated by randomly offsetting three of the corner points of the template and computing the parameters yielding the affine transformation between the ground truth and offset positions. This is similar to the test framework used by Baker and Matthews in [1]. In total, 600 tests per image were performed. These consisted of six groups of 100 tests, where a different standard deviation was associated with each group and a normal distribution was used for the random offsets. Standard deviations of 2, 4, 6, 8, 10, and 12 pixels were used. The results over 50 outer-loop iterations of a Levenberg-Marquardt (LM) algorithm are plotted for each image in the second column of Fig. 2. LM rather than Newton optimization was used since it's use of multiple function evaluations makes it more robust to tracking failure. The number of iterations was not fixed, two termination criteria were also used: when the change in function value was too low, i.e., $|f^{(k)} - f^{(k-1)}| < 10^{-4}$) and when the maximum change in any one warp parameter was too low, i.e., $\max_n |\Delta\mathbf{v}_n^{(k)}| < 10^{-4}$, where n indexes the component of $\Delta\mathbf{v}$.

The number of outer-loop iterations is not directly related to the number of function evaluations since the number of inner-loop iterations may vary. In LM optimization, G and H are only

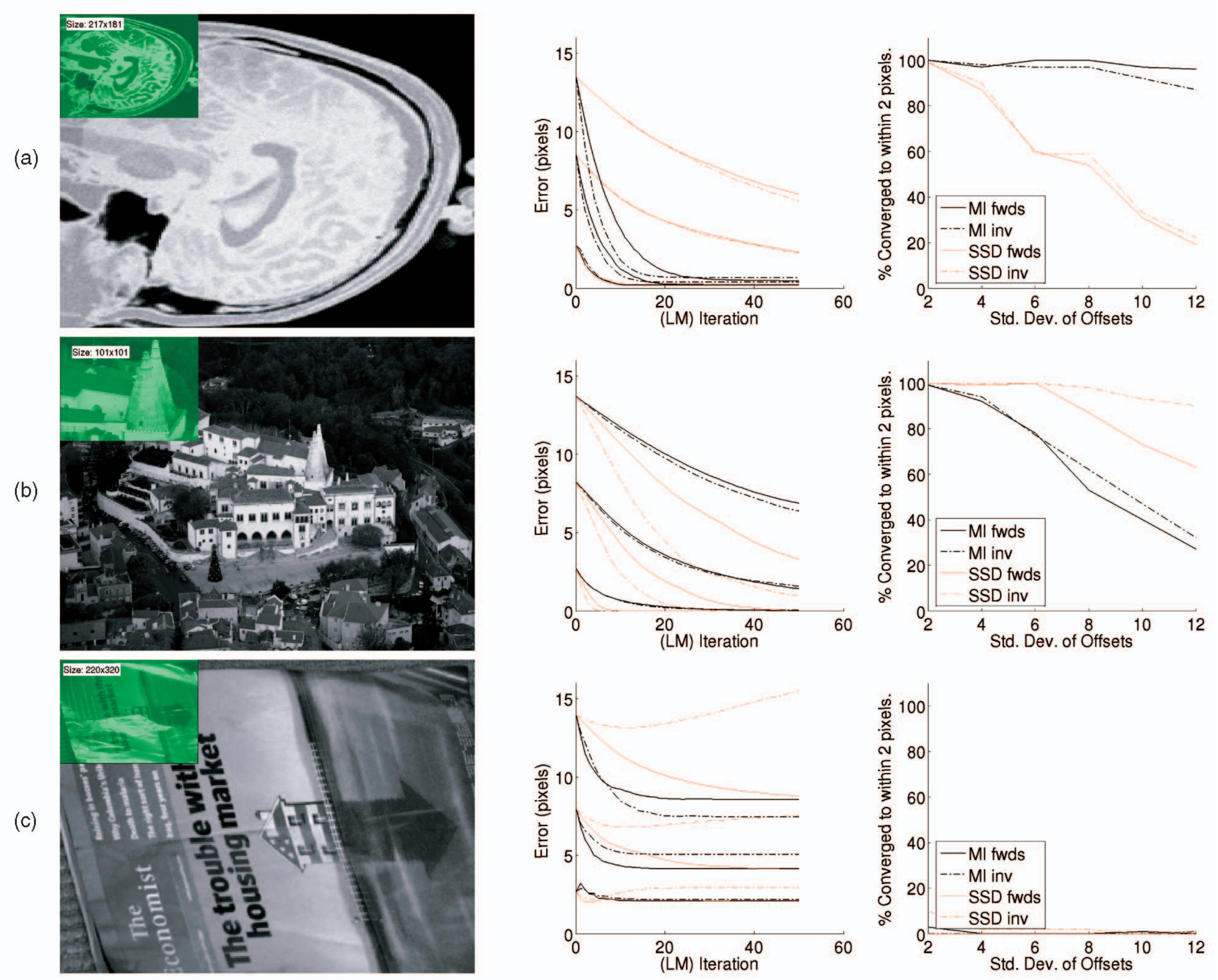

Fig. 2. Convergence rate for three typical examples of registration problems dealing with: (a) multiple modalities in medical images, (b) clutter in natural images, and (c) specularities in images. In all cases, the template used is displayed as a shaded green template in the upper left-hand corner. The size of the template is displayed. The error over 50 outer loop iterations of the Levenberg-Marquardt algorithm is shown for MI and SSD in both cases for forwards additive and inverse compositional formulations.

	Mean Time / Std. Dev.(s)				Evaluations of f / f,G & $\tilde{H}$			
Image	MI fwds	MI inv.	SSD fwds	SSD inv.	MI fwds	MI inv.	SSD fwds	SSD inv.
Image 1: Brain	2.6/1.2	2.2/0.78	2.4/1.2	2.5/1.1	36/18	35/16	39/34	38/36
Image 2: Palace	1.4/0.66	1.3/0.61	0.36/0.18	0.21/0.14	64/37	63/37	31/32	19/20
Image 3: Economist	3.6/1.5	3.6/1.6	5.1/2.3	4.3/2.4	32/13	33/14	39/33	38/27

Fig. 3. Time (mean and standard deviation) and number of evaluations (of inner loop evaluations where only f is measured and outer loop evaluations where f, G, and $\tilde{H}$ are calculated) required to converge for the three images (across all tests). Note that the inverse-compositional methods do not recalculate the Hessian, but reuse the precalculated one.

recalculated from the images once per outer-loop iteration. For inverse-compositional formulations $\tilde{H}$ is *reused*. Only f is reevaluated every inner loop iteration. Since LM optimization may terminate early, the number of inner and outer loop iterations is displayed, along with the mean time per optimization, in Fig. 3. Additional optimizations were made, by reusing computational constructs utilized to calculate f, for calculating, G and again for calculating $\tilde{H}$. This partially obscures the advantages of using a precomputed $\tilde{H}$ since the cost of evaluating $\tilde{H}$ in addition to G is low. A larger increase in cost occurs when H is calculated using second order information. This is clearly shown in Fig. 4, where the mean cost per evaluation per pixel for each method tested was measured over 20 tests.

In the first image, inverse-compositional MI managed to converge faster than forwards-additive MI. The difference arose because inverse-compositional MI utilizes an approximate Hessian as discussed in Section 3.1. The initially faster convergence was a surprising result which is believed to occur because the forwards-additive approach updates the Hessian to model local conditions, leading it to take more conservative steps than the inverse-compositional algorithm. However the constant histogram also has the side effect of a larger final error than the forwards-additive approach. The final error also increased with initial offset because the region in I_r initially overlapped had less in common with the overlapped region when the algorithm finally converged. This implies that, once convergence is reached, optimization should be restarted with an *updated* Hessian that accounts for local conditions. An experiment to verify if restarting the algorithm would improve the final error for the inverse-compositional formulation was undertaken. One hundred random offsets with a standard deviation

Values Evaluated	Time / pixel (μs)			
	MI fwds	MI inv.	SSD fwds	SSD inv.
f	.41	.42	.18	.15
f & G	2.5	2.3	.88	.77
f, G & $\tilde{H}$	2.4	2.4	1.6	1.6
f, G & H	28.	15.	11.	11.

Fig. 4. Cumulative cost of evaluating a function, its Jacobian, Hessian, and full second order Hessian. Results for both SSD and MI, for forwards-additive and inverse-compositional approaches, are shown as a time cost per pixel averaged over 20 evaluations. A $1,884 \times 1,252$ template was used, which was sufficient that the overhead of function initialization was negligible. Affine transformations were used.

of 12 pixels were used used to initialize optimization for the medical image pair shown in Fig. 2a. The results of these tests, given in Fig. 5, show that restarting the algorithm reduced the final error of the inverse-compositional method to that of the forwards-additive method. The re-evaluation of the Hessian caused by the restart came at an increased time cost of 2.9s versus 2.7s without the restart (3.3s for the forwards-additive approach). The SSD tests are of interest only in this case since the nonlinear relationships of multimodal medical images are well known to confound SSD.

In the second image pair, SSD converged faster than MI in every case. This is due to the direct match of the template and reference intensities (in the correct position) and the large basin of convergence typical of SSD. MI, on the other hand, appeared to become trapped or slowed by the local topology of function surface. Although MI is perhaps not the ideal distance function for the a problem like image 2, note how the inverse-compositional approach for MI performs as well as the forwards additive approach.

Image 3 is a difficult problem due to the radical changes to intensity induced by specularities on the object as lighting conditions change. MI's tolerance of nonlinear intensity relationships allows slightly better performance than SSD in that convergence is always toward ground truth, but convergence failure occurs in most cases. Inverse-compositional MI performs comparably to forwards-additive MI, although it too becomes trapped in local minima in many cases.

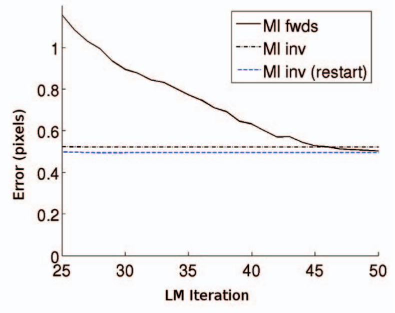

Fig. 5. Experiment showing the improvement in final accuracy for the inverse-compositional formulation when restarting optimization. Only the last 25 iterations are shown to make the improvement visible.

For MI, the mean time to convergence was faster for the inverse-compositional formulation in most cases, despite the thriftiness of the LM method with Hessian evaluations. Time savings up to 15 percent were made. Approximately the same number of iterations were performed in both cases. For SSD, the inverse-compositional approach was almost twice the speed of the forwards-additive approach for image 2, agreeing with previous work [1]. The speed improvement was due partly to the efficiency of the inverse-compositional formulation and partly to the fewer iterations required to converge on average. This pattern was not repeated for images 1 and 3 for two reasons: first, LM's thriftiness with Hessian evaluations and second, the cost of Hessian evaluation was reduced by reusing parts of the function and Jacobian evaluations.

The speed of SSD relative to MI indicated that, for many applications, it is still the method of choice. Standard sampled MI

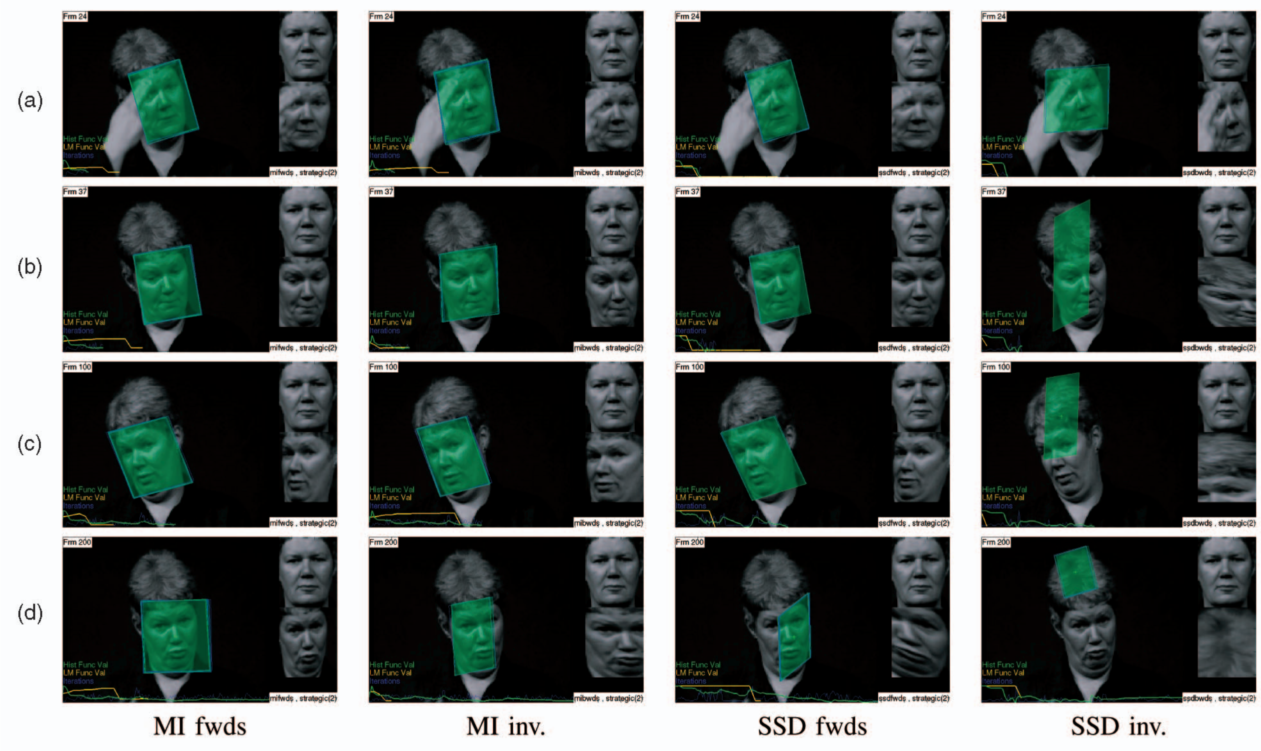

Fig. 6. Tracking a face over a video sequence using various algorithms at (a) Frame 24, (b) Frame 37, (c) Frame 100, and (d) Frame 200. Four algorithms were used, namely, the forwards-additive and inverse-compositional formulations for MI and SSD. All of the methods performed comparably, except in the case of SSD an occlusion halfway through the sequence pulled the SSD trackers of target. MI supposedly deals with occlusions better. In our experience, this is not always the case.

can compete with SSD in speed [7], but, due to its noisy function surface, does not perform well for Newton-type optimization methods. It does perform well for random sampling optimization methods and with the simplex algorithm, but these algorithms do not use the Hessian, so an inverse-compositional approach holds no advantages.

In addition to the tests above, a tracking algorithm was implemented using the strategic update approach of Matthews et al. [12]. In it, the tracking performance of forwards-additive MI and inverse-compositional MI were compared on a video sequence. For comparable real-time performance to SSD, standard sampled MI was used rather than the in-Parzen windowing approach used in the tests above. The results are shown in Fig. 6. As shown, the inverse-compositional approach performed just as well as forwards additive. Results using SSD are also shown for interest. SSD also tracked quite well, but a large occlusion by the hand pulled the tracker off target.

5 Conclusion

In this paper, an inverse compositional formulation for Mutual Information was introduced. This reformulates the MI function to yield an approximate Hessian that is dependent only upon the template image values and is therefore constant. The Hessian is approximate because cross terms between the reference and template intensities exist, which cannot be separated and vary with the warp parameter. These can be assumed to be constant too because the variation warp parameter is small. Inverse Compositional SSD on the other hand has an exact Hessian (to first order) because the effects of the two images are entirely separable.

The result of several experiments showed that inverse-compositional MI could compete with forwards-additive MI in terms of registration accuracy and demonstrated computational savings of up to 15 percent. This improvement occurred despite two confounding factors. First, Levenberg-Marquardt optimization does not require a Hessian to be evaluated every time a function is calculated. Second, the Hessian computation was optimized by reusing components utilized to calculate the function value. For methods where the Hessian is required every iteration, like the Newton method, the computational improvement would be greater. In testing, SSD and MI each performed best in different applications, suggesting that neither function is ideal in all circumstances.

The source code and test harness of this work have been made available at the authors' Web site. In future work, the effect of using more accurate joint-histogram approximations estimated from the marginal histograms will be investigated. Testing with nonrigid 3D data transformations will also be examined, as will reformulations of other functions like Normalized Correlation.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments and suggestions and they would also like to thank T. Kadir for all his helpful advice.

References

[1] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *Int'l J. Computer Vision,* vol. 56, no. 3, pp. 221-255, Mar. 2004.

[2] R.P. Brent, "An Algorithm with Guaranteed Convergence for Finding a Zero of a Function," *Computer J.,* vol. 14, no. 4, pp. 422-425, 1971.

[3] R. Brooks and T. Arbel, "Generalizing Inverse Compositional Image Alignment," *Proc. Int'l Conf. Pattern Recognition,* vol. II, pp. 1200-1203, Aug. 2006.

[4] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated Multi-Modality Image Registration Based on Information Theory," *Information Processing in Medical Imaging,* pp. 263-374, Kluwer Academic, 1995.

[5] T. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 681-685, June 2001.

[6] A. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 1403-1410, Oct. 2003.

[7] N. Dowson and R. Bowden, "A Unifying Framework for Mutual Information Methods for Use in Non-Linear Optimisation," *Proc. Ninth European Conf. Computer Vision,* A. Leonardis, H. Bischof, and A. Prinz, eds., vol. 1, pp. 365-378, May 2006.

[8] G. Hager and P. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, pp. 1025-1039, Oct. 1998.

[9] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, Aug. 1981.

[10] F. Maes, D. Vandermeulen, and P. Seutens, "Comparative Evaluation of Multiresolution Optimization Strategies for Multimodalitiy Image Registration by Maximization of Mutual Information," *Medical Image Analysis,* vol. 3, no. 4, pp. 272-286, Apr. 1999.

[11] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. SIAM,* vol. 11, no. 2, pp. 431-441, June 1963.

[12] I. Matthews, T. Ishikawa, and S. Baker, "The Template Update Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 810-815, June 2004.

[13] J. Pluim, J. Maintz, and M. Viergever, "Mutual-Information-Based Registration of Medical Images: A Survey," *IEEE Trans. Medical Imaging,* vol. 22, no. 8, pp. 986-1003, Aug. 2003.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C,* second ed. Cambridge Univ. Press, 1992.

[15] C. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical J.,* vol. 27, pp. 379-423, 623-656, July-Oct. 1948.

[16] H.-Y. Shum and R. Szeliski, "Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment," *Int'l J. Computer Vision,* vol. 36, no. 2, pp. 63-84, 2000.

[17] C. Studholme, D. Hill, and D. Hawkes, "Automated 3D Registration of Truncated MR and CT Images of the Head," *Proc. British Machine Vision Conf.,* pp. 27-36, Sept. 1995.

[18] P. Thevenaz and M. Unser, "Optimization of Mutual Information for Multi-Resolution Image Registration," *IEEE Trans. Image Processing,* vol. 9, no. 12, pp. 2083-2099, Dec. 2000.

[19] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part I—Theory," *IEEE. Trans. Signal Processing,* vol. 41, no. 2, pp. 821-833, Feb. 1993.

[20] P. Viola and W. Wells, "Alignment by Maximization of Mutual Information," *Proc. Int'l Conf. Computer Vision,* pp. 16-23, June 1995.