

Accelerated 3D Image Registration

Martin Vester-Christensen, Søren G. Erbou, Sune Darkner and Rasmus Larsen

Informatics and Mathematical Modelling, Technical University of Denmark,
Richard Petersens plads, DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

Image registration is an important task in most medical imaging applications. Numerous algorithms have been proposed and some are widely used. However, due to the vast amount of data collected by eg. a computed tomography (CT) scanner, most registration algorithms are very slow and memory consuming. This is a huge problem especially in atlas building, where potentially hundreds of registrations are performed. This paper describes an approach for accelerated image registration. A grid-based warp function proposed by Cootes and Twining, parameterized by the displacement of the grid-nodes, is used. Using a coarse-to-fine approach, the composition of small diffeomorphic warps, results in a final diffeomorphic warp. Normally the registration is done using a standard gradient-based optimizer, but to obtain a fast algorithm the optimization is formulated in the inverse compositional framework proposed by Baker and Matthews. By switching the roles of the target and the input volume, the Jacobian and the Hessian can be pre-calculated resulting in a very efficient optimization algorithm. By exploiting the local nature of the grid-based warp, the storage requirements of the Jacobian and the Hessian can be minimized. Furthermore, it is shown that additional constraints on the registration, such as the location of markers, are easily embedded in the optimization. The method is applied on volumes built from CT-scans of pig-carcasses, and results show a two-fold increase in speed using the inverse compositional approach versus the traditional gradient-based method.

Keywords: Image registration, Lucas-Kanade algorithm, Inverse Compositional algorithm

1. INTRODUCTION

Registration of images is an important and actively researched area of medical imaging. It is the task of transforming the geometry of two or more images such that their corresponding regions are aligned. The need may arise from comparison of images from different imaging modalities, from images obtained at different times, from different patients, or from comparison with a patient atlas. Registration is needed in a wide variety of medical applications, eg. for diagnostic purposes, for pre-surgery planning or for treatment estimation. The medical imaging hardware of today produce images of high resolution, and as a consequence a huge amount of data need processing in order to solve the registration problem. The requirements on the processing hardware are very high in terms of storage capability, CPU speed, and maybe most importantly in memory capacity.

A range of different registration algorithms have been produced in the recent years. The algorithm described in this paper, belongs to the class of parameterized methods. Hence the registration can be described by a set of parameters. Existing methods includes Rueckert¹ *et al.* using B-splines on a grid to define the warp-field, Cootes² *et al.* uses bounded diffeomorphisms, warping pixels inside a unit sphere based on the displacement of the sphere center. An example of the non-parametric approach is found in Christensen³ *et al.* which solve partial differential equations for fluid motions to align images.

This paper presents the acceleration of an image registration algorithm⁴ by Cootes *et al.*. An inverse compositional⁵ optimization scheme, proposed by Baker and Matthews, is used. It is a Gauss-Newton approach, but in which the Jacobian and the estimated Hessian can be precomputed. However, to be tractable memory wise, this requires exploitation of the properties of the registration algorithm. An additional benefit of the Gauss-Newton approach is the ease of which addition of soft constraints on the registration can be added. Comparison is made with optimization using the Lucas-Kanade scheme.

Further author information: Send correspondence to Martin Vester-Christensen, E-mail: mvc@imm.dtu.dk, Telephone: +45 4525 5228

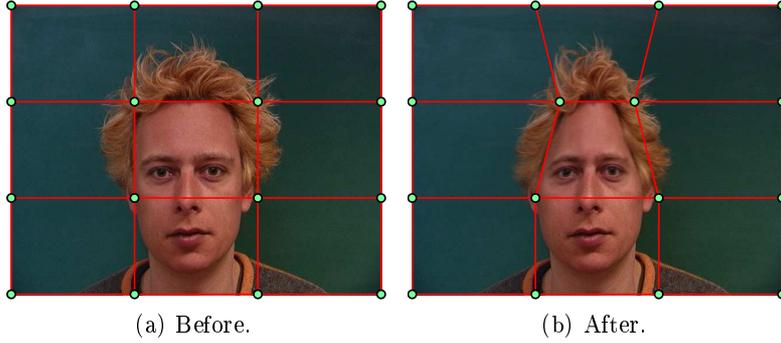


Figure 1. The displacement of a pixel is governed by the displacement of its surrounding grid nodes.⁷

The paper consist of the following. First the registration algorithm is described. In 2.1 the image registration algorithm is described, followed by outlining of the Lucas-Kanade algorithm, in 2.2, and the inverse compositional optimization algorithm, in 2.3. Addition of soft constraints is described in 2.4. Section 3 describes the results obtained by comparing the two optimization schemes.

2. METHODS

The image registration algorithm utilized in this paper is proposed by Cootes⁴ *et al.* which builds on the algorithms presented by Rueckert⁶ *et al.* The image registration is performed by composing a series of grid-based diffeomorphic warps which ensures the resulting warp being diffeomorphic.

2.1. Grid-Based Diffeomorphisms

A grid-based warp is represented by a grid of nodes*, see figure 1. The transformation \mathbf{W} of a pixel $\mathbf{x} = (x, y, z)$ is found by interpolating the displacement \mathbf{d} of its surrounding grid nodes. In 3D the interpolating scheme is as below,

$$\begin{aligned}
 \mathbf{W}(\mathbf{x}; \alpha) &= k(x-i)k(y-j)k(z-l)\mathbf{d}_{i,j,l} & + & k(i+1-x)k(y-j)k(z-l)\mathbf{d}_{i+1,j,l} \\
 &+ k(x-i)k(j+1-y)k(z-l)\mathbf{d}_{i,j+1,l} & + & k(i+1-x)k(j+1-y)k(z-l)\mathbf{d}_{i+1,j+1,l} \\
 &+ k(x-i)k(y-j)k(l+1-z)\mathbf{d}_{i,j,l+1} & + & k(i+1-x)k(y-j)k(l+1-z)\mathbf{d}_{i+1,j,l+1} \\
 &+ k(x-i)k(j+1-y)k(l+1-z)\mathbf{d}_{i,j+1,l+1} & + & k(i+1-x)k(j+1-y)k(l+1-z)\mathbf{d}_{i+1,j+1,l+1},
 \end{aligned} \tag{1}$$

where $k()$ denotes a suitable kernel function which is non-zero only for $i \leq x < i+1$, $j \leq y < j+1$ and $l \leq z < l+1$. The warp is parameterized with the components of the displacement vectors \mathbf{d} . Thus, with a $3 \times 3 \times 3$ grid in 3D, the warp consist of 81 parameters. The kernel is chosen as $k(r) = \frac{1}{2}(1 + \cos(\pi r))$ which gives a smooth and invertible mapping⁴ given that $-\frac{1}{\pi} < r < \frac{1}{\pi}$. Using the interpolating scheme the warp is regularized by the coarseness of the grid. Thus, a pixel cannot move outside the bounding box provided by the surrounding grid nodes. However, to represent a complex transformation several simple warps can be composed,

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \mathbf{W}(\mathbf{x}; \delta_1) \circ \mathbf{W}(\mathbf{x}; \delta_2) \dots \mathbf{W}(\mathbf{x}; \delta_{n-1}) \circ \mathbf{W}(\mathbf{x}; \delta_n), \tag{2}$$

where $\mathbf{W}(\mathbf{x}; \delta_1) \circ \mathbf{W}(\mathbf{x}; \delta_2) = \mathbf{W}(\mathbf{W}(\mathbf{x}; \delta_2); \delta_1)$ denotes the composition of two warps. The warps are applied in a fine to coarse manner.

2.2. Image Registration using the Lucas-Kanade Algorithm

The goal of the image registration algorithm is to align a target and a input image such that the difference is minimized. This is quantified by the minimization of the sum of squared residuals,

$$\mathbf{F}(\mathbf{p}) = \sum_{\mathbf{x}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2, \tag{3}$$

*The following is an elaboration on the paper⁴ by Cootes *et al.*, but is included here for completeness.

where I is the input image and T is the target image. This can be minimized using a Gauss-Newton optimization scheme,⁸

$$F(\mathbf{p}) = \frac{1}{2} \sum_{\mathbf{x}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta\mathbf{p}))]^2, \quad (4)$$

which by Taylor expansion and solving for $\Delta\mathbf{p}$ gives,

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} \right]^{\top} E(\mathbf{x}), \quad (5)$$

where \mathbf{H} is the Gauss-Newton approximation to the Hessian,

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} \right]^{\top} \left[\nabla I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} \right], \quad (6)$$

and the error is,

$$E(\mathbf{x}) = T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})). \quad (7)$$

The warp parameters \mathbf{p} are updated using,

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}. \quad (8)$$

The Jacobian is found to be,

$$\mathbf{J} = \sum_{\mathbf{x}} \left[\nabla I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} \right], \quad (9)$$

where $\nabla I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ is the image gradient of the input image sampled at the points $\mathbf{W}(\mathbf{x}; \mathbf{p})$, and $\frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}}$ is the derivative of the warp function with respect to the parameters.

This optimization scheme requires computation of the Jacobian \mathbf{J} and the inverse Hessian \mathbf{H}^{-1} at each iteration. For large volumes and large grids, this is very computationally demanding. However, the Hessian is symmetric and very sparse thus enabling the utilization of fast schemes for solving large sparse linear equations.⁹

2.3. Inverse Compositional Image Registration

To overcome the drawbacks of the Gauss-Newton scheme of calculating the Jacobian and the Hessian in each iteration, Baker and Matthews⁵ recently proposed the *Inverse Compositional Algorithm*, in which the Jacobian and the Hessian can be precomputed. As the name implies the algorithm consists of two innovations. The compositional part refers to the updating of the parameters and the inverse part indicates that the image and the target switches roles. The cost function in 4 is changed to,

$$F_{ic}(\mathbf{p}) = \frac{1}{2} \sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2. \quad (10)$$

Solving for $\Delta\mathbf{p}$ gives,

$$\Delta\mathbf{p} = -\mathbf{H}_{ic}^{-1} \sum_{\mathbf{x}} \left[\nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]^{\top} E(\mathbf{x}). \quad (11)$$

The update to the warp is,

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta\mathbf{p})^{-1}, \quad (12)$$

In equation 10 it can be seen that the incremental warp $\mathbf{W}(\mathbf{x}; \Delta\mathbf{p})$ applies only to the target T , and thus the Taylor expansion is around $\mathbf{p} = \mathbf{0}$, yielding the Jacobian

$$\mathbf{J}_{ic} = \sum_{\mathbf{x}} \left[\nabla T(\mathbf{W}(\mathbf{x}; \mathbf{0})) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]. \quad (13)$$

and thus the Hessian \mathbf{H}_{ic} is,

$$\mathbf{H}_{ic} = \sum_{\mathbf{x}} \left[\nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]^\top \left[\nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]. \quad (14)$$

The Jacobian is independent of \mathbf{p} and $\nabla T(\mathbf{x})$ is the image gradient of the target, thus enabling precomputation of the Jacobian and the Hessian.

Baker and Matthews⁵ proves that the update $\Delta \mathbf{p}$ calculated using the inverse compositional algorithm is equivalent, to a first order approximation, to the update calculated using the Lucas-Kanade algorithm.

2.4. Adding Constraints

Baker *et al.*¹⁰ describe how to incorporate prior information on the warp parameters. This could for instance be landmark or volume constraints formulated as an additional term in the expression to be minimized, i.e. as weighted soft constraints to equation 3,

$$\frac{1}{2} \sum_{\mathbf{x}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2 + \alpha \sum_{i=1}^K C_i^2(\mathbf{p}). \quad (15)$$

K is the number of constraints, C_i is a vector of functions containing the prior on the parameters for the i^{th} constraint and α is a weight controlling the emphasis on the prior term. In the inverse compositional framework this corresponds to equation 10,

$$\frac{1}{2} \sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})))]^2 + \alpha \sum_{i=1}^K C_i^2(\mathbf{p} + \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \Delta \mathbf{p}). \quad (16)$$

Approximating this with a first order Taylor expansion gives the following update equations for the gradient $\Delta \mathbf{p}$ and the Hessian,

$$\Delta \mathbf{p} = -\mathbf{H}_{ic, C_i}^{-1} \left[\sum_{\mathbf{x}} \left[\nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]^\top E(\mathbf{x}) + \alpha_{Gr} \sum_{i=1}^K \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right]^\top C_i(\mathbf{p}) \right] \quad (17)$$

$$\mathbf{H}_{ic, C_i} = \mathbf{H}_{ic} + \alpha_{He} \sum_{i=1}^K \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right]^\top \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right]. \quad (18)$$

The computational cost of adding priors is that the Hessian is not constant anymore. The costs is $O(nN + n^2K + n^3)$ compared to $O(nN + n^2)$ without priors. As long as the number of constraints and the number of parameters are smaller than the number of pixels/voxels ($K \ll N$ and $n \ll N$), this cost is negligible. In order to make the prior terms robust to the number of constraints, the α weights are chosen relative to the L2-norm of the term without priors,

$$\alpha_{Gr} = \alpha_{Rel} \frac{\left\| \sum_{\mathbf{x}} \left[\nabla T(\mathbf{x}) \frac{\partial \mathbf{W}(\mathbf{x}; \mathbf{0})}{\partial \mathbf{p}} \right]^\top E(\mathbf{x}) \right\|_{L2}}{\left\| \sum_{i=1}^K \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right]^\top C_i(\mathbf{p}) \right\|_{L2}} \quad (19)$$

$$\alpha_{He} = \alpha_{Rel} \frac{\left\| \mathbf{H}_{ic} \right\|_{L2}}{\left\| \sum_{i=1}^K \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right]^\top \left[\frac{\partial C_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} \right] \right\|_{L2}}, \quad (20)$$

where $\alpha_{Rel} \geq 0$ is the relative weighting between the two terms. $\alpha_{Rel} = 0$ corresponds to the inverse compositional without the prior term.

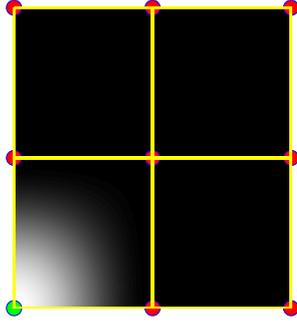


Figure 2. The derivative of the warp with respect to the node in the lower left corner. It is non-zero only in the region bounded by the neighboring nodes.

2.5. Fast Grid-Based Image Registration

The inverse compositional scheme described above requires the computation the Jacobian from equation 13 and subsequently the Hessian from equation 14. To take full advantage of the inverse compositional method, they must be precalculated and stored. However, the storage requirement of the Jacobian can be very large. In a naive implementation using a simple $3 \times 3 \times 3$ grid, space is needed for $81 \cdot N$ floating point numbers with N being the number of pixels in the image and potentially very large. So great care must be taken in the implementation in order to exploit the speed gain provided by the inverse compositional framework.

2.5.1. Calculating the Jacobian and the Hessian

The Jacobian consist of the gradient $\nabla T(\mathbf{x}) = \left[\frac{\partial T}{\partial x} \quad \frac{\partial T}{\partial y} \quad \frac{\partial T}{\partial z} \right]$ of the target image and the derivative of the warp $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$. $\nabla T(\mathbf{x})$ can be found using a simple finite difference method or more elaborate methods using B -splines or similar interpolating methods. The warp derivatives are very simple to calculate. As mentioned, the parameters of the warp are simply the $3 \cdot mnp$ ordinates of the displacements of the nodes,

$$\mathbf{p} = [d_{1,1,1}^x \quad d_{1,1,1}^y \quad d_{1,1,1}^z \quad d_{2,1,1}^x \quad d_{2,1,1}^y \quad d_{2,1,1}^z \dots d_{m,n,p}^x \quad d_{m,n,p}^y \quad d_{m,n,p}^z], \quad (21)$$

for a $m \times n \times p$ grid. This and equation 1 yields

$$\frac{\partial \mathbf{W}}{\partial d_{i,j,l}^x} = \frac{\partial \mathbf{W}}{\partial d_{i,j,l}^y} = \frac{\partial \mathbf{W}}{\partial d_{i,j,l}^z}, \quad (22)$$

meaning that the derivatives corresponding to the x, y, z components of one displacement vector are equal. Each pixel \mathbf{x} contributes to eight partial derivatives only, corresponding to the eight surrounding grid nodes, cf. figure 2 for a 2D example. Thus, for N pixels, an $N \times 8$ floating point value representation of $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is possible using a simple lookup method. However, since the kernel function $k(\cdot)$ only operates on the distance from a contributing pixel \mathbf{x} to a node, an even sparser representation of only $N_{reg} \times 8$ is possible. N_{reg} is the number of pixels surrounded by 8 neighboring grid nodes. For increasing grid sizes the space requirement goes down.

Another property of $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is seen from,

$$k(i+1-x) = 1 - k(x-i), \quad (23)$$

which holds for y and z as well. For a neighborhood of grid nodes, the derivative at $\mathbf{x} = (x, y, z)$ wrt. the

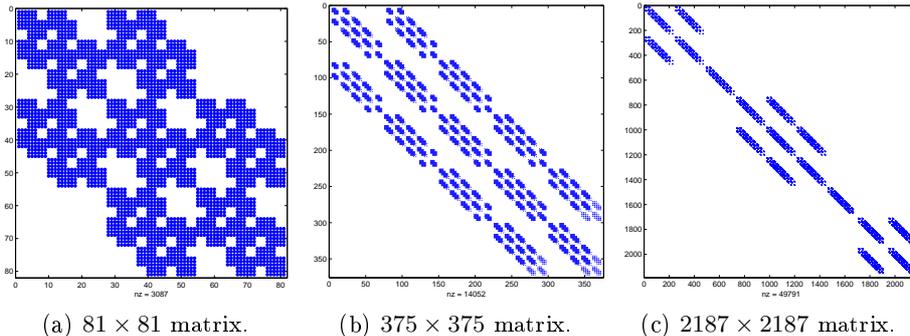


Figure 3. Plot of non-zero elements in Hessian matrices, for grid sizes of $3 \times 3 \times 3$, $5 \times 5 \times 5$ and $9 \times 9 \times 9$.

parameters of the warp has the form,

$$\begin{aligned} \frac{\partial \mathbf{W}}{\partial p_{i,j,l}} &= k(x-i)k(y-j)k(z-l) \\ \frac{\partial \mathbf{W}}{\partial p_{i+1,j,l}} &= (1-k(x-i))k(y-j)k(z-l) \\ &\vdots \\ \frac{\partial \mathbf{W}}{\partial p_{i+1,j+1,l+1}} &= (1-k(x-i))(1-k(y-j))(1-k(z-l)) \end{aligned} \quad (24)$$

Thus, to evaluate the derivative contribution for a single pixel \mathbf{x} , only three kernel function evaluations, $k(x-i)$, $k(y-j)$ and $k(z-l)$ are needed. The derivatives can then be found using simple multiplications and subtractions.

Finally the Jacobian is calculated as in equation 13, which means multiplying the warp derivatives with the image gradients. However, the Jacobian with the sparse representation mentioned above has a size of $N \times 24$ floating point values which requires a large amount of memory. Subsequently a compromise has to be made, and in this work the image gradients and the warp derivatives are stored separately, yielding space requirement for $N \times 3$ plus $N_{reg} \times 8$ numbers. This means the multiplication of the gradients and the derivatives must be performed each time the Jacobian is needed, yet precalculation of the Hessian is still possible.

The Hessian is calculated as in equation 14. The sparseness of the Jacobian is transferred into the Hessian. Figure 3 depicts the sparseness of the Hessian for three grid sizes. Solving equation 11 requires the inversion of the Hessian matrix. This usually destroys the sparseness and is very computationally demanding. Furthermore, the Hessian can be very ill-conditioned, but making use of iterative methods⁹ for solving sparse linear equations, the inversion of the Hessian can be avoided.

2.5.2. The Image Registration Algorithm

To estimate the transformation of the input image I into the target image T minimization of equation 10 with respect to \mathbf{p} is required. This is done by first applying a coarse grid, eg. $3 \times 3 \times 3$, and iteratively solving equation 11 until convergence. Subsequently a finer grid, eg. $5 \times 5 \times 5$, is applied, and so forth. This enables the estimation of small local transformations while still being diffeomorphic.⁴ A multilevel approach, using downsampled versions of the images, is adopted to avoid local minima. Consequently, for each grid size, optimization is done in a coarse-to-fine manner as well, starting the optimization on a finer downsampling level with the parameters estimated on a coarser level. See figure 4. At each iteration equation 12 must be used to update the parameters. In this work the parameters of $\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})$ is estimated with the first order approximation¹⁰ $-\Delta \mathbf{p}$. Thus the parameter update from equation 12 has the form,

$$\mathbf{p} \leftarrow \mathbf{p} - \Delta \mathbf{p}. \quad (26)$$

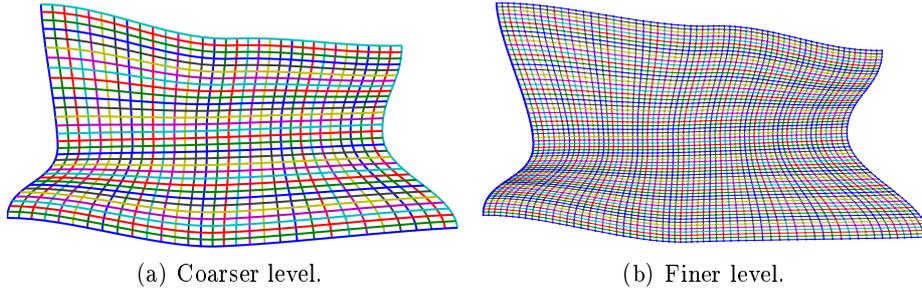


Figure 4. Plot of the warfield estimated on a coarse image level and the corresponding field in a finer level.

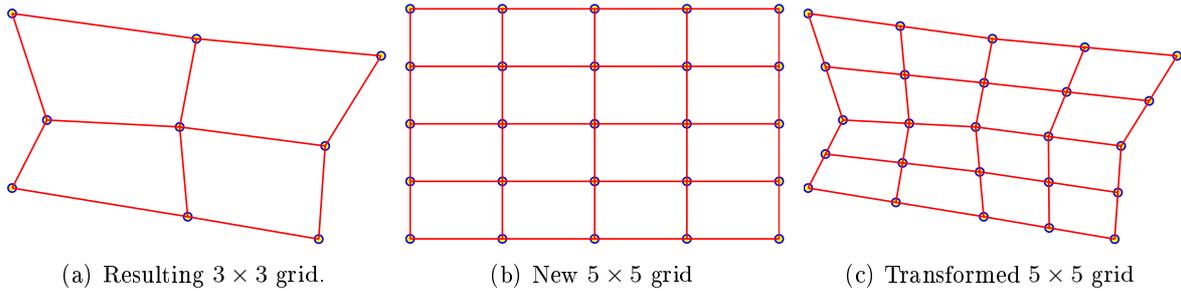


Figure 5. Composition of grids of different sizes.

When the optimization for one grid size converges, a finer grid is applied, ie. a $3 \times 3 \times 3$ is replaced with a $5 \times 5 \times 5$. Composition of warps of different grid sizes is done in a simple manner. Figure 5 depicts the scheme. The grid nodes of the higher level warp are transformed with the lower level warp. Thus, the parameters from the lower level are transported into the higher level warp.

3. RESULTS

3.1. 3D Non-Rigid Registration

5 CT-volumes of the hind part of porcine carcasses are cross-registered to compare the inverse compositional algorithm with the Lucas-Kanade algorithm. The volumes are approximately $512 \times 512 \times 170$ voxels of size $[0.67, 0.67, 2]$ mm. After rigid registration the two algorithms for non-rigid registration are applied and the speed and accuracy are compared. Due to time considerations the 20 registrations are done with images downsampled to $\frac{1}{16}$, $\frac{1}{8}$ and $\frac{1}{4}$ of the original image size for grids of size $[3, 5, 7, 9]$ per dimension. Figure 6 shows three slice planes of a volume and their corresponding error images after a typical registration. The main errors are along the border of the volume due to the large difference in value between background and volume.

Rows 1 and 2 in table 1 show the mean value and standard deviation of the number of iterations used before convergence, the final registration error and the time consumption for the two algorithms. Row 3 shows the mean improvement when using the IC algorithm and row 4 shows a paired T-test of significant differences in the mean values. There are highly significant improvements (denoted by 1) in both speed and number of iterations. The mean final registration error of the two methods are not significantly different. In this simple test it therefore shows that the inverse compositional algorithm is as accurate as the Lucas-Kanade algorithm, as expected, but is twice as fast for registration of CT-volumes. The results are obtained using a Dell Latitude 810D, with a 2.0Ghz CPU and 2 gb of ram. The implementation of the Lucas-Kanade algorithm utilizes the same sparseness properties as the implementation of the inverse compositional algorithm. Thus, the algorithms perform similar memory wise. The speed increase of the inverse compositional algorithm is due to the precomputation of the Hessian and the target image gradient. In the paper⁴ by Cootes *et al.* the optimization is done by a simple gradient descent scheme. The gradient is computed by displacement of each of the grid nodes in turn. Early

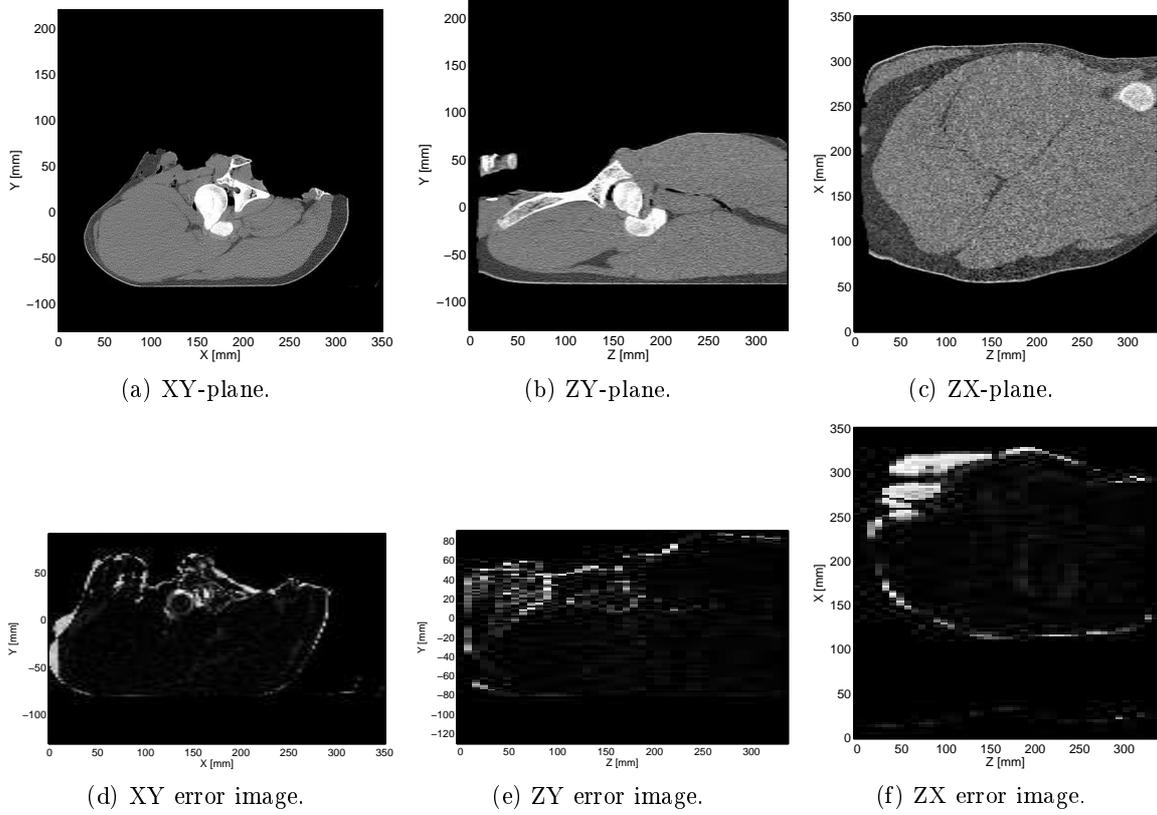


Figure 6. (a)-(c): CT-volume of a porcine carcass. (d)-(f): Error images.

Table 1. Comparison between the Lucas-Kanade algorithm and the inverse compositional algorithm based on 20 registrations.

	Iterations	Final Error	Speed [sec.]
Lucas-Kanade (mean \pm std.)	160 \pm 29	$(7.6 \pm 4.1) \cdot 10^{11}$	436 \pm 128
Inverse Compositional (mean \pm std.)	140 \pm 15	$(7.5 \pm 4.0) \cdot 10^{11}$	222 \pm 45
Mean improvement with IC (%)	13	2	49
Paired T-test for difference in mean, 1=mean values are sign. diff.	1 ($p < 0.006$)	0	1 ($p < 10^{-3}$)

experiments have shown this approach to be very slow, so no comparison is made. However, no space is required for precomputation, and as such it performs well memory wise.

3.2. Adding Constraints

If some prior knowledge is at hand before registration or if there is a need for guiding the registration, adding constraints in the optimization scheme should be considered. As an example 58 landmark constraints are applied in a 2D affine warp of image I in figure 7(a) to image T in figure 7(b). The warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ of a pixel \mathbf{x} is defined by 6 parameters in vector \mathbf{p}

$$\mathbf{W}(\mathbf{x}, \mathbf{p}) = \begin{pmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (27)$$

The first order term in the linearization of the additive update is

$$\frac{\partial \mathbf{p}'}{\partial \Delta \mathbf{p}} = - \begin{pmatrix} 1 + p_1 & p_3 & 0 & 0 & 0 & 0 \\ p_2 & 1 + p_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 + p_1 & p_3 & 0 & 0 \\ 0 & 0 & p_2 & 1 + p_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 + p_1 & p_3 \\ 0 & 0 & 0 & 0 & p_2 & 1 + p_4 \end{pmatrix}. \quad (28)$$

The error function of the prior part is defined as

$$C_i = (Lm_{T,i} - \mathbf{W}(Lm_{I,i}; \mathbf{p})), \quad (29)$$

where $Lm_{T,i}$ is the i^{th} landmark in the target image T and $Lm_{I,i}$ is the i^{th} landmark in the input image I . The Jacobian of C_i then is

$$\frac{\partial C_i}{\partial \mathbf{p}} = - \begin{pmatrix} Lm_{I,i,x} & 0 & Lm_{I,i,y} & 0 & 1 & 0 \\ 0 & Lm_{I,i,x} & 0 & Lm_{I,i,y} & 0 & 1 \end{pmatrix}, \quad (30)$$

where $Lm_{I,i,x}$ and $Lm_{I,i,y}$ are the x- and y-coordinates, respectively, of the i^{th} landmark in I.

Figure 7(c) shows the L2-norm of the intensity and prior error as α_{Rel} is increased. $\alpha_{Rel}=0$ corresponds to no constraints and $\alpha_{Rel}=1$ corresponds to the intensity and the constraints being weighted equally. The constraints improve the registration as long as the intensity error is decreasing, i.e. for α -values approximately between 0.8 and 1.6. The prior error will of course decrease with increasing α -value. How much weight to put on the constraints depends on the application but for this example weighting intensity and prior more or less equal gives the best result. Figures 7(d)-7(f) show the difference between the input image I and the target image T warped into the coordinate frame of I, for $\alpha_{Rel}=0,1$ and 2. The improvement in registration without constraints compared to the registration with the intensity and constraints weighted equally is obvious, especially in the area around the jaw.

Applying similar constraints as described in section 2.4 to the nonrigid case or in 3D is straightforward, all you need to do is to define C_i and compute the Jacobian and $\partial \mathbf{p}' / \partial \Delta \mathbf{p}$.

4. CONCLUSION

This paper has presented an algorithm for registration of 3D images. Registration is done using grid-based warps in a coarse-to-fine manner, enabling the registration of even fine structures in the images while still being diffeomorphic. Using the inverse compositional framework for optimization, the algorithm performs very fast. Exploitation of the sparseness of the grid-based warps and the properties of the interpolating kernel, enables the precomputation of the Hessian and the target image gradient. The algorithm has a two-fold increase in speed compared to a Lucas-Kanade based algorithm.

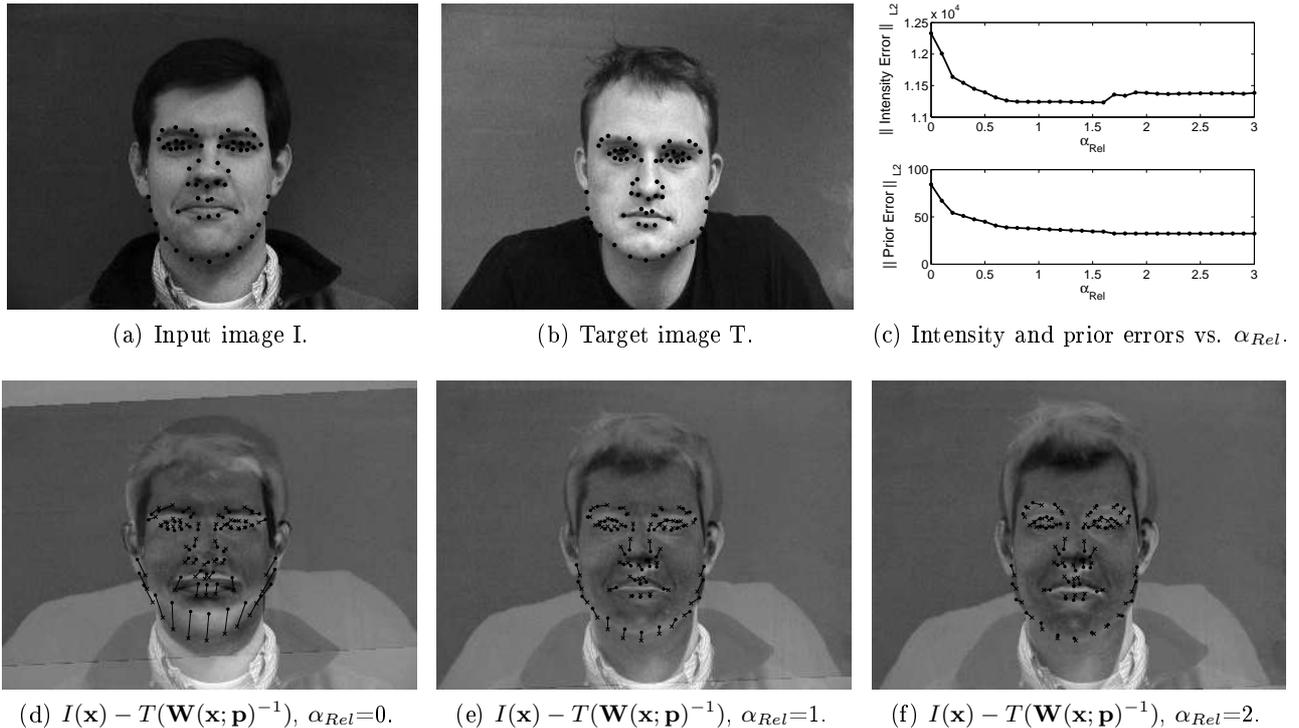


Figure 7. 2D affine registration of 2 images with increasing weight on landmark constraints.

ACKNOWLEDGMENTS

The 2D images are part of the DTU Face Database⁷ and the CT data was provided by the Danish Meat Research Institute as a part of the project "The Virtual Butcher" funded by the Danish Pig Levy Fund and the Directorate for Food, Fisheries and Agri Business. The authors would also like to thank the IMM ITMAN graduate school at the Technical University of Denmark for support.

REFERENCES

1. D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: Application to breast mr images," *IEEE Trans. Medical Imaging* **18**, pp. 712–721, August 1999.
2. T. Cootes, S. Marsland, C. Twining, K. Smith, and C. Taylor, "Groupwise diffeomorphic non-rigid registration for automatic model building," in *European Conference on Computer Vision*, pp. Vol IV: 316–327, 2004.
3. G. Christensen, *Deformable Shape Models for Anatomy*. PhD thesis, Washington University, 1994.
4. T. Cootes, C. Twining, and C. Taylor, "Diffeomorphic statistical shape models," in *British Machine Vision Conference*, pp. 447–456, 2004.
5. S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision* **56**, pp. 221–255, February 2004.
6. D. Rueckert, A. Frangi, and J. Schnabel, "Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration," *IEEE Trans. Medical Imaging* **22**, pp. 1014–1025, August 2003.
7. M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME - A Flexible Appearance Modelling Environment," *IEEE Transactions on Medical Imaging* **22**(10), pp. 1319–1331, 2003.
8. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (darpa)," in *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130, April 1981.

9. G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, USA, second ed., 1989.
10. S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework: Part 4," tech. rep., Robotics Institute, Carnegie Mellon University, 2004.