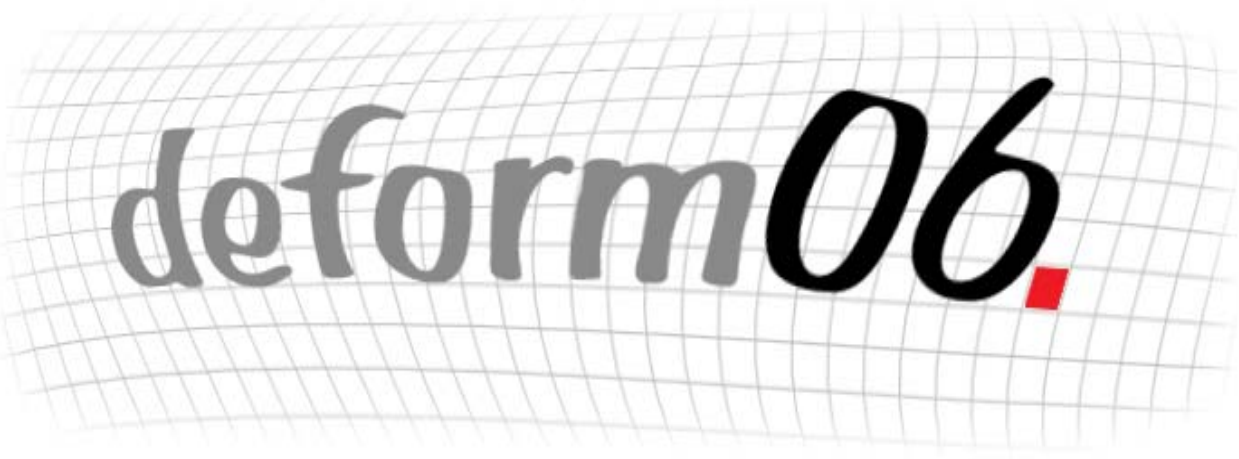
DEFORM'06 - Proceedings of the Workshop on Image Registration in Deformable Environments

Adrien Bartoli, Nassir Navab, Vincent Lepetit

Adrien Bartoli
Nassir Navab
Vincent Lepetit (Eds.)



DEFORM'06

Workshop on Image Registration in Deformable Environments
8 September 2006, Edinburgh, UK

Proceedings

DEFORM'06 is associated to BMVC'06, the 7th British Machine Vision Conference

Preface

These are the proceedings of DEFORM'06, the Workshop on Image Registration in Deformable Environments, associated to BMVC'06, the 17th British Machine Vision Conference, held in Edinburgh, UK, in September 2006.

The goal of DEFORM'06 was to bring together people from different domains having interests in deformable image registration.

In response to our Call for Papers, we received 17 submissions and selected 8 for oral presentation at the workshop. In addition to the regular papers, Andrew Fitzgibbon from Microsoft Research Cambridge gave an invited talk at the workshop.

The conference website including online proceedings remains open, see HTTP://COMSEE.UNIV-BPCLERMONT.FR/EVENTS/DEFORM06.

We would like to thank the BMVC'06 co-chairs, Mike Chantler, Manuel Trucco and especially Bob Fisher for is great help in the local arrangements, Andrew Fitzgibbon, and the Programme Committee members who provided insightful reviews of the submitted papers. Special thanks go to Marc Richetin, head of the CNRS Research Federation TIMS, which sponsored the workshop.

August 2006

Adrien Bartoli
Nassir Navab
Vincent Lepetit

Organization

General Co-Chairs

Adrien Bartoli	LASMEA – CNRS / University Clermont II, France
Nassir Navab	Technical University of Munich, Germany
Vincent Lepetit	EPFL, Lausanne, Switzerland

Programme Committee

Lourdes Agapito	Queen Mary, University of London, UK
Amir Amini	University of Louisville, USA
Fred Azar	Siemens Corporate Research, Princeton, USA
Marie-Odile Berger	INRIA, Nancy, France
Tim Cootes	University of Manchester, UK
Eric Marchand	INRIA, Rennes, France
Andrew Davison	Imperial College London, UK
Pascal Fua	EPFL, Lausanne, Switzerland
Fredrik Kahl	Lund Institute of Technology, Sweden
Ali Khamene	Siemens Corporate Research, Princeton, USA
Nikos Paragios	Ecole Centrale de Paris, France
Xavier Pennec	INRIA, Sophia-Antipolis, France
Daniel Rueckert	Imperial College London, UK
Laurent Sarry	University Clermont I, France
Peter Sturm	INRIA, Grenoble, France
Luc Van Gool	ETH Zurich, Switzerland and University of Leuven, Belgium
René Vidal	Johns Hopkins University, Baltimore, USA

Additional Reviewers

Xavier Lladó	Queen Mary, University of London, UK
Julien Pilet	EPFL, Lausanne, Switzerland
Christophe Tilmant	University Clermont II, France
Geert Willems	University of Leuven, Belgium

Website, Design and Proceedings

Vincent Gay-Bellile	LASMEA – CNRS / University Clermont II, France
Martin Horn	Technical University of Munich, Germany
Mathieu Perriollat	LASMEA – CNRS / University Clermont II, France

Sponsoring Institution

TIMS is a CNRS Research Federation
in Clermont-Ferrand, France

Table of Contents

Regular Papers

Invited Talk

Using Subdivision Surfaces for 3–D Reconstruction from Noisy Data

Slobodan Ilić
Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
Slobodan.Ilic@telekom.de

Abstract

In this work, we developed a method which effectively uses subdivision surfaces to fit deformable models to noisy 3–D data. The subdivision models provide a compact low dimensional representation of model parameter space. They also allow a free form deformation of the objects of arbitrary geometry. These two properties make them ideal for fitting deformable models to noisy data.

We tested our approach on stereo data acquired from uncalibrated monocular video sequences, then on data acquired from low quality laser scans, and compared them. We have shown that it can be successfully used to reconstruct human faces as well as the other 3D objects of arbitrary geometry.

1 Introduction

The subdivision surfaces are very popular approach for smooth surface representation. They has been extensively used in Computer Graphics for geometric modeling, and computer animation [15, 12, 4]. The problem of 3D reconstruction from unorganized data points using the subdivision surfaces has also been addressed by the graphics community [7, 9, 11, 14, 2, 8]. Those methods use noise free data produced by high quality laser scanners. In Computer Vision the 3D shapes were extracted from the images which involve very noisy information. Since this problem is highly under-constrained, the generic models were extensively used [6, 13, 1] to constrain the solution of the reconstruction task. The main challenge is to find a suitable model parameterization where the number of parameters is small compared to the overall dimensionality of the problem. On the other hand the parameterization has to support a full free form deformation of the target surface. The subdivision surfaces naturally possess those properties. They produce a smooth mesh from a coarse one as shown in Fig. 2. The initial coarse mesh, *control mesh*, is a very rough representation of the object of interest. It can be of arbitrary topology, and subdivision can be applied N times to it. The final smooth mesh, *model mesh*, entirely depends on the initial control mesh. Its shape can be modified in a free form manner. Also, the number of control mesh vertices is far smaller than the number of actual model vertices. In this paper we develop a method which relays on subdivision surface models to reconstruct 3D objects from noisy data shown in Fig. 1 as opposed to clean laser scan data used in Computer Graphics. We are not aware of any other method which uses subdivision surfaces for 3D shape recovery from nosy data. Besides valuable properties such as free form deformation and low dimensionality, subdivision models, allowed us to use the initial control mesh as a regularizer in our optimization framework.

(a)

(b)

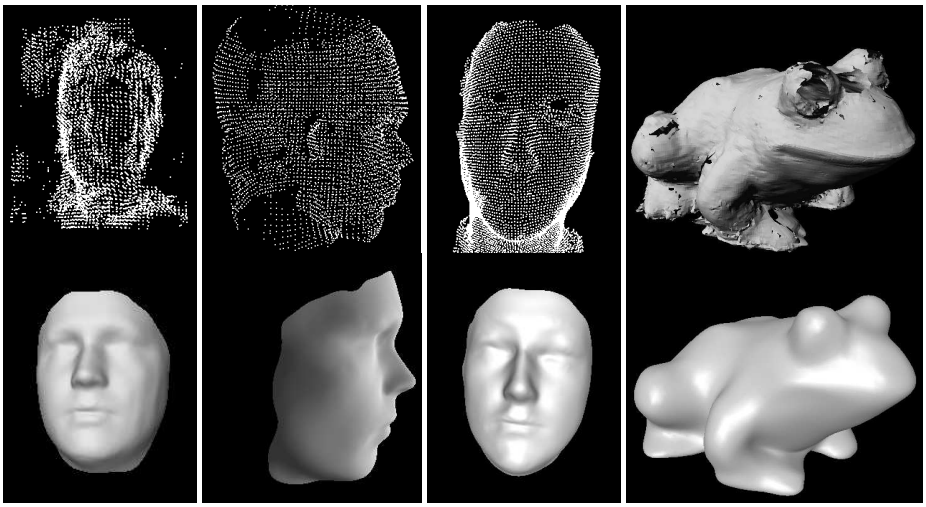


Figure 1: Reconstruction of 3D objects from noisy data: (a) 3–D data clouds corresponding to stereo and laser scans respectively. First two images on the left side are stereo data clouds obtained from uncalibrated, Fig 5, and calibrated, Fig 6, image sequences respectively, while the other two correspond to the laser scans.(b) Reconstruction results obtained by fitting generic subdivision models to the data from (a). For faces a generic face model was used, while for the frog model is extracted from the scan.

We chose to demonstrate and evaluate our technique mainly in the context of face-modeling from uncalibrated and calibrated images sequences and from the laser scan data. We compare the recovered models from images to those of the laser scans in order to verify the quality of the reconstruction. To demonstrate that our method can be applied to objects of different geometry we reconstructed a ceramic statue of the frog from its laser scan. The data we are dealing with are noisy and incompletge as it can be seen in Fig 1(a), and, yet, we obtained realistic models, whose geometry is correct, as depicted in Fig. 1(b).

According to these experiments we argue that the approach is generic and can be applied to any task for which deformable facetized models exist. We therefore view our contribution as the integration of a powerful shape representation technique into a robust least-squares framework.

In the remainder of the paper, we first describe subdivision parameterization in more details. We then introduce our least-squares optimization framework and, finally, present results using laser scanned data, calibrated and uncalibrated image sequences.

2 Subdivision Surface Parameterization

The key property of subdivision models is that the smooth high resolution mesh models are represented by very coarse initial control mesh. The control mesh consists of relatively small number of vertices and completely controls the shape of the smooth mesh model. The model mesh is obtained by subsequent subdivision of the initial coarse mesh as depicted in Fig. 2.

(a)

(b)

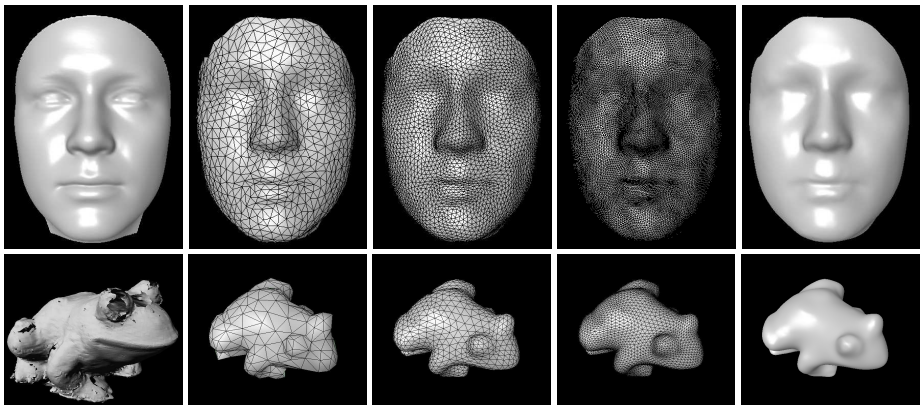


Figure 2: Generic models and their subdivisions. (a) The first image on the left corresponds to the average face model of [1]. The next image shows the actual control mesh we use. It is obtained by reducing the average face model for 98.8% decreasing number of vertices from 53235 to only 773. The following images represent model meshes corresponding to one and two levels of subdivision. The last image is just shaded version of two times subdivided control mesh. (b) Now, the first image on the left corresponds to the laser scan of the ceramic frog statue. Following images represent control mesh, model mesh with one and two levels of subdivision and two times subdivided control mesh shown as shaded.

Our method requires a generic model of the object we want to reconstruct. To quickly produce those models we take advantage of the property of the subdivision that it can be applied to meshes of arbitrary geometry and topology. Thus, the generic control mesh can be quickly extracted from already existing high quality generic models or directly form the data as discussed in Sec. 4 and depicted by Fig. 2.

In the reminder of this section we will review Loop subdivision schema for one level of subdivision and discuss the hierarchy of subdivision if multiple levels of subdivision are applied.

2.1 Loop Subdivision

The Loop scheme is a simple approximating face-split scheme applicable only to triangular meshes proposed by Charles Loop [10]. The edge splitting produces *odd vertices* while vertex splitting produces *even vertices.* The new vertices are *weighted sum* of control point neighbors from previous level of subdivision. The weights define subdivision rules and differ for interior and boundary vertices. These rules can be illustrated by a local masks called subdivision masks as shown in Fig. 3.

Because of well established rules, the computation of the coefficients and dependencies between control and model vertices is straightforward. The coefficients can be assembled into a subdivision matrix. Therefore we can parameterize our subdivided mesh by control mesh as matrix multiplication $V^{mod}_{m\times 1} = \mathbf{S}_{m\times n} V^{ctrl}_{n\times 1}$, where $S_{m\times n}$ is a subdivision matrix with m being the number of vertices of subdivided mesh and n number of vertices of control mesh.

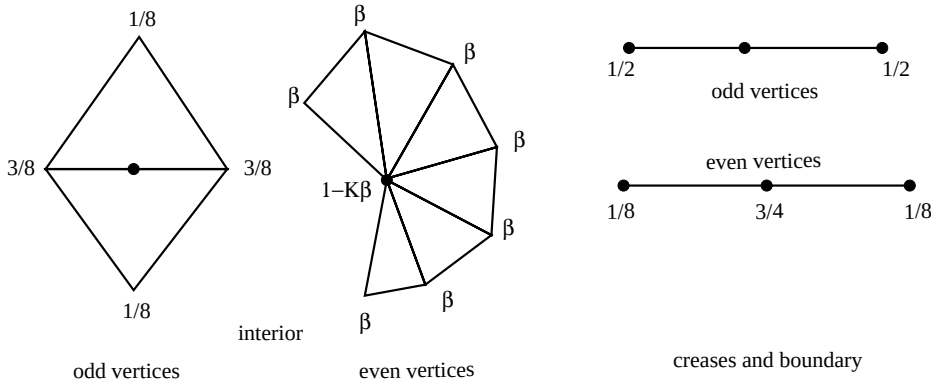


Figure 3: Loop subdivision masks. The adjacent vertices of the interior even vertices have a weight β, while the interior even vertex has a weight of $1-k\beta$ where K is the number of adjacent vertices and $\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{K}\right)\right)$.

2.2 Hierarchy of subdivision

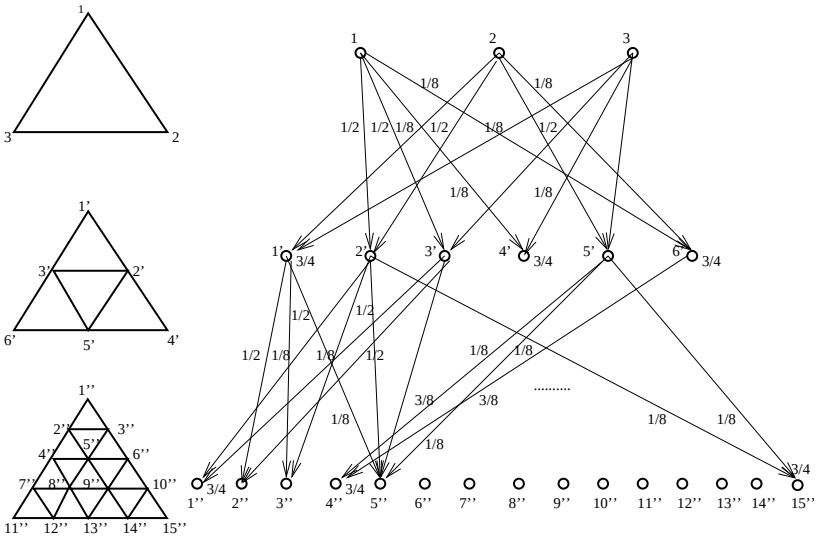


Figure 4: Schematic representation of the dependency graph. Three levels of subdivision are depicted, because of the simplicity, for a single triangle.

To parameterize the model when more levels of subdivision are used we apply a hierarchy of subdivision. In that case we do the following matrix multiplication $V^{k+1} = \mathbf{S}^k\mathbf{S}^{k-1}\cdots\mathbf{S}^1V^1$. In theory this is a compact way of representing subdivision operation, but in practice multiplication of huge matrices requires significant computation resources what is impractical. In order to avoid matrix multiplication we have to find out which vertices at any subdivision level are influenced by selected control vertex. We can compute these dependencies by using graph data structure. For that purpose we build a *dependency graph* of Fig. 4. Each layer of this graph represents a subdivision level and each edge represents an influence relation to which we attach weight coefficient from the subdivision

masks. In order to find out which vertices are influenced by the selected control vertex we should propagate all weights at each level. We recursively traverse the graph to find the influenced vertices. Note that this relation is permanent and has to be computed once in the beginning.

3 Optimization Framework for Subdivision Surface Fitting

In this section, we introduce the framework we have developed to fit subdivision surfaces to noisy image data. We fit *mesh models* such as the ones of Fig. 2 corresponding to several levels of subdivision. Our goal is to deform the surface—without changing its topology—so that it conforms to the image data. The deformation is controled by the *control mesh*, such as one of Fig. 2. In this work data is made of 3–D points computed using stereo or laser scan data. In standard least-squares fashion, for each point observation $\mathbf{x}_i$, we write an observation equation of the form $d(\mathbf{x}_i, S) = \varepsilon_i$, where S is a state vector corresponding to the vertices of the control mesh which defines the shape of the surface, d is the distance from the point to the surface and ε_i is the residual error measuring the deviation of the data from the model. In practice $d(\mathbf{x}_i, S)$ is taken to be the orthonormal distance of $\mathbf{x}_i$ to the closest surface triangulation facet. This results in *nobs* observation equations forming a vector of residuals $F(S) = [\varepsilon_1, ..., \varepsilon_i, ...]^T_{1 \le i \le nobs}$ that we minimize in the least squares sense by minizing its square norm:

$$\chi^2 = \frac{1}{2} F(S)^T W F(S) = \sum_{i=1}^{nobs} w_i d(\mathbf{x}_i, S)^2 \ . \tag{1}$$

Because some of the observations, may be spurious, we weigh them to eliminate outliers. Weighting is done as the preprocessing step, before the real fitting is started. Observation weight w_i are taken to be inversely proportional to the initial distance $d(\mathbf{x}_i, S)$ of the data point to the model surface. More specifically we compute w_i weight of the obs_i as:

$$w_i = exp(\frac{d_i}{\overline{d_i}}), 1 \le i \le nobs \tag{2}$$

where $\overline{d_i}$ is the median value of the $d(\mathbf{x}_i, S)$. In effect, we use $d(\mathbf{x}_i, S)$ as an estimate of the noise variance and we discount the influence of points that are more than a few standard deviations away. Because of the non-differentiability of the distance function we recompute the point to facet attachments before every minimization procedure.

In theory we could take the parameter vector S to be the vector of all x,y, and z coordinates of the model mesh. However, because the data we are dealing with are very noisy and the number of parameters is huge we found that the fitting process to be very brittle since we deal with very ill-conditioned problem.

3.1 Subdivision parametrization

We therefore use vertices *control mesh* such as the one of Fig. 2 to be our state vector. More precisely, in our scheme, we take the state vector S to be the vector of 3-D displacements of the control mesh vertices, which is very natural using the subdivision formalism.

For each vertex on the model mesh, obtained by several subdivisions of the initial control mesh, we extract the control vertices and their weights influencing those model vetices using dependency graph of Fig. 4. This allows us to express displacement of every model vertex i after k subdivisions as the weighted linear combination of displacements of control points from previous subdivision levels which influence them as:

$$\triangle V_i^k = Vn_i^k - V_i^k = \sum_{c_{k-1}=1}^{N_{c_{k-1}}} \sum_{c_{k-2}=1}^{N_{c_{k-2}}} \cdots \sum_{c_0=1}^{N_{c_0}} w_{c_{k-1}} w_{c_{k-2}} \cdots w_{c_0} \triangle V_{c_0}^0 \ . \tag{3}$$

where Vn_i^k is the new position after the deformation and V_i^k is the initial position of the ith model vertex obtained after k subdivisions. Indices c_{k-j} count number of infulencing control verteces on the previous subdivision level $k-j$, $j = n, 0$ where n is the number of subdivision levels. Equally, $N_{c_{k-j}}$, $j = n, 0$ is the actual number of control vertices on the subdivision level $k-j$ influencing the position of the ith model vetrex. Weights $w_{c_{k-j}}$ are associated weights from Loop's subdivision masks of the control vetices on the previous subdivision level.

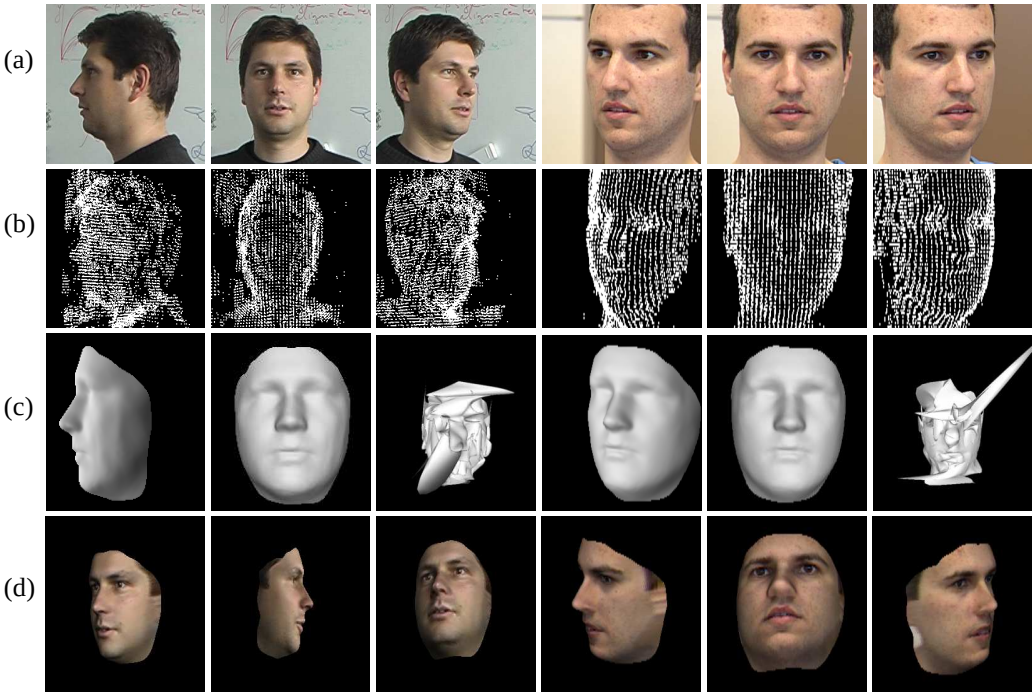


Figure 5: Reconstruction from an uncalibrated video sequences. (a) Three images chosen from two short video sequence where the total number of used images is 15 and 6 respectively. (b) Stereo data clouds shown in the same projection as the input images. (c) Reconstructed faces shown as shaded models. Note the figures where we did not use regularization during the fitting. The amount of noise seriously influenced reconstruction. (d) Texture-mapped version of the recovered 3–D models viewed from different view points.

(a)

(b)

Figure 6: Reconstruction from an calibrated video sequence. (a) Five images chosen from the calibrated video sequence, courtesy of IGP, ETH Zürich. (b) Reconstructed face shown as shaded model overlapped on the original image in the same perspective in order to highlight the quality of the reconstruction.

3.2 Regularization

Because there are both noise and potential gaps in the image data, we found it necessary to introduce a regularization term. Since we start with a generic model, we expect the deformation between the initial shape and the original one to be smooth. This can be effectively enforced by preventing deformations at neighboring vertices of the control mesh to be too different. If the control points formed a continuous surface, a natural choice would, therefore, be to take this term to be the sum of the square of the derivatives of the $\triangle V_i^0$ displacements across the initial control mesh. By treating its facets as C^0 finite elements, we can approximate regularization energy E_D as the quadratic form

$$E_D = \lambda S^T K S = \lambda \left(\Delta_x^t K \Delta_x + \Delta_y^t K \Delta_y + \Delta_z^t K \Delta_z \right) \quad (4)$$

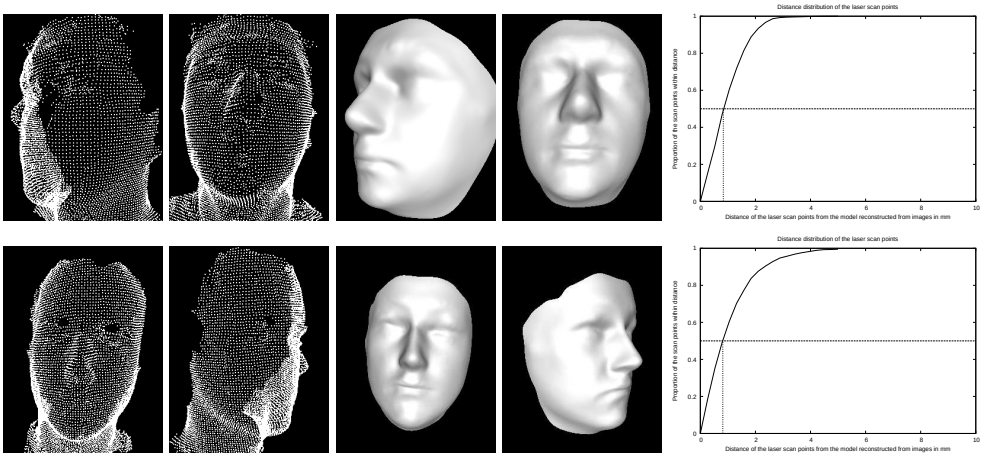
where λ is small regularization constant, K is a very sparse stiffness matrix, S is a state vector containing displacements of along the x,y or z coordinates at each point of this control surface.

4 Results

We demonstrate and evaluate our technique mainly in the context of face modeling since many tools for that are available to us. We show its flexibility also by using it to fit a generic model of the frog to the noisy scan data.

4.1 Reconstruction from Image Sequences

We recover faces from uncalibrated sequances of Fig. 5(a) and calibrated one of Fig. 6(a) sequences. In the first row of Fig. 5(a) we choose to show three images from two different uncalibrated video sequences. In both cases, we had no calibration information about the camera or its motion. We therefore used a model-driven bundle-adjustment technique [6] to compute the relative motion and, thus, register the images. We then used the technique [5] to derive the clouds of 3–D points depicted by Fig. 5(b). Because we used fewer images and an automated self-calibration procedure as opposed to a sophisticated manual



(a) (b) (c) (d) (e)

Figure 7: Reconstruction from an laser scan data and comparison with the reconstruction of the same subjects obtained from the images. (a,b) Laser scan data clouds of two persons from Fig 5 shown in different positions. (c,d) Reconstructed faces shown as shaded model. (e) Quantitative evaluation of the facial reconstructions form images, where the models reconstructed from the scans are used as a ground truth. For the subjects of Fig. 5, we show the proportion of the 3–D vertices in the models reconstructed from the laser scans that are within a given distance of the reconstructed model from the images, after they have been aligned using an affine transformation.

one, the resulting cloud of 3–D points is much noisier and harder to fit. Shaded models obtained after the fitting are depicted on Fig. 5(c). These models can also be reused to re-synthesize textured images such as the ones of Fig. 5(d). The initialization is done by finding an affine transformation between the model face and the data cloud according to 5 manually provided points on the — top of the nose and corners of the eyes and the mouth — through the least-square minimization.

We also illustrate the effectiveness of our approach using relatively clean stereo data. We use the sequence of forty 512x512 calibrated images where some are depicted by the first row of Fig. 6(a). The reconstructed face is then overlapped on the images using the same projection matrices as the once producing the images. Note that the reconstruction appears to be precise as shown in Fig. 6(b).

4.2 Reconstruction from Laser Scan Data

To evaluate our method and quantify the quality of the reconstruction from images, we have used a Minolta^{tm} laser scanner to acquire data of the subjects of Figs. 7(c,d). Those results were obtained by fitting the generic model of Fig. 2(a). The same model was used for reconstruction from images. Since the laser data are cleaner, but, yet, not perfect, the reconstruction seems to be of greater precision. Because of that we used them as the ground truth and compared the image-based reconstructions to them. We therefore compute an affine transform that brings the reconstructed image-based model to the scan-based model. This is done by least-squares minimization where we look for the affine transformation which brings two models together. This is correct since the intrinsic

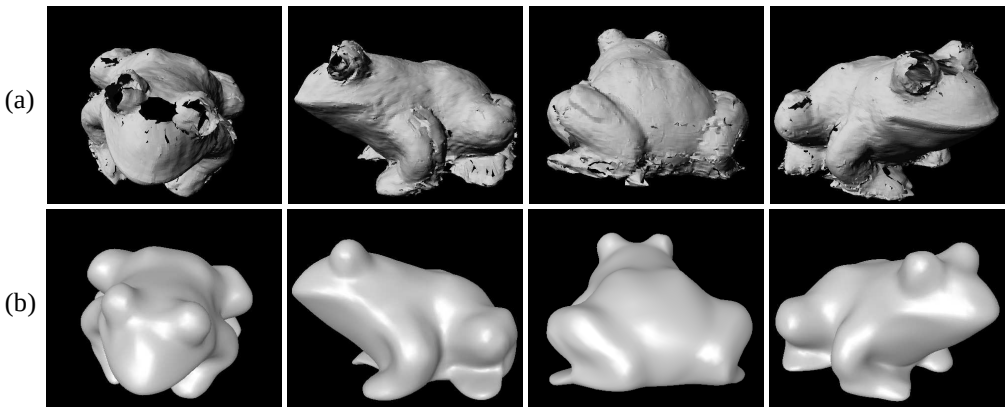


Figure 8: Reconstruction of the laser scanned frog statue. (a) Laser scan data shown in different position indicating holes and ambiguous parts coming from stitching of the individual scans. (b) Reconstructed frog model shown in similar positions as the scan. Note, that the holes are filled and the noise is smoothed out.

camera parameters are given the approximate values during bundle-adjustment procedure. This resulted in stereo reconstruction precise up to the affine transformation. In Fig. 7(e), we plot for both subjects, the proportion of the 3–D vertices of the laser-based recovered model that are within a given distance of the image-based reconstructed model, after it has been deformed using the affine transform discussed above. The median distances are 0.833858*mm* and 0.815620*mm* respectively. Comparing the reconstruction from images to the original laser scan data gives the increase of the median error to 1.94*mm* and 2.28*mm* respectively. This is normal since the original laser scan data are noisy in contrast to the model fitted to it and used as ground truth in Fig 7.

Our scanner has a theoretical precision of around 0.3*mm*. However, this does not take into account some of the artifacts it produces, such as those that occur when stitching together several scans of the target object seen from different viewpoints. A median distance between two independent reconstructions being under 1.0*mm* is therefore a good indication that our results are consistent with each other. Finally, we show the result of reconstructing the frog from the noisy laser scan. The frog statue is made of smooth ceramic material. Since our scanner is static it required the object to be turned and individual scans to be stitched. The specularities on the eyes produced holes in the scan. As it can be seen in the Fig. 8(a) the scan has quite a lot of ambiguities and missing parts what made it challenging for reconstruction. The results we obtained are smooth. The ambiguities, such as one back on the left leg are smoothed and the holes on the eyes are filled as shown in Fig. 8(b). In this way we managed to significantly improve the reconstruction from the scan, by simply extracting the model from the scan and then fitting it back to the original data.

5 Conclusion

In this work, we proposed to use the subdivision surfaces to fit generic surface models to noisy 3–D image and laser scan data. We demonstrated the effectiveness and robustness of our technique in the context of face modeling where the generic model was available. We also showed that, in fact, we can model other complex shape, such as frog statue, for

which a deformable triangulated model exists.

In future work we intend to investigate the possibility of using reverse subdivision to produce even better generic models out of the available hight quality models. We also intend to investigate the use of such models in the context of model-based bundle adjustment where both shape and camera parameters can be extracted simultaneously from image feature correspondences.

References

[1] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3–D Faces. In *Computer Graphics, SIGGRAPH Proceedings*, pages 187–194, Los Angeles, CA, August 1999.

[2] K.-S. D. Cheng, W. Wang, H. Qin, K.-Y. K. Wong, H. Yang, and Y. Liu. Fitting subdivision surfaces to unorganized point data using sdm. In *Pacific Conference on Computer Graphics and Applications*, pages 16–24, 2004.

[3] D. DeCarlo and D. Metaxas. The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996.

[4] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94, New York, NY, USA, 1998. ACM Press.

[5] P. Fua. From Multiple Stereo Views to Multiple 3–D Surfaces. *International Journal of Computer Vision*, 24(1):19–35, August 1997.

[6] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2):153–171, July 2000.

[7] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[8] W. K. Jeong and C. H. Kim. Direct reconstruction of displaced subdivision surface from unorganized points. In *Pacific Conference on Computer Graphics and Applications*, pages 160–169, October 2001.

[9] N. Litke, A. Levin, and P. Schröder. Fitting subdivision surfaces. In *Proceedings of the conference on Visualization '01*, pages 319–324. IEEE Computer Society, 2001.

[10] C. Loop. Smooth Subdivision Surfaces Based on Triangles. Master thesis, Department of Mathematics, University of Utah, 1987.

[11] V. Scheib, J. Haber, M. C. Lin, and H.-P. Seidel. Efficient fitting and rendering of large scattered data sets using subdivision surfaces. *Computer Graphics Forum*, 21(3), 2002.

[12] P. Schroder. Subdivision as a fundamental building block of digital geometry processing algorithms. *Journal of Computational and Applied Mathematics*, 149, 2002.

[13] Y. Shan, Z. Liu, and Z. Zhang. Model-Based Bundle Adjustment with Application to Face Modeling. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[14] H. Suzuki, S. Takeuchi, F. Kimura, and T. Kanai. Subdivision surface fitting to a range of points. In *Pacific Conference on Computer Graphics and Applications*, page 158. IEEE Computer Society, 1999.

[15] D. Zorin, P. Schröder, A. DeRose, L. Kobbelt, A. Levin, and W. Sweldens. Subdivision for modeling and animation. In *SIGGRAPH 2000 Course Notes*, 2000.

A Single Directrix Quasi-Minimal Model for Paper-Like Surfaces

Mathieu Perriollat **Adrien Bartoli**

LASMEA - CNRS / UBP ◇ Clermont-Ferrand, France

Mathieu.Perriollat@lasmea.univ-bpclermont.fr, Adrien.Bartoli@gmail.com

http://comsee.univ-bpclermont.fr

Abstract

We are interested in reconstructing paper-like objects from images. These objects are modeled by developable surfaces and are mathematically well-understood. They are difficult to minimally parameterize since the number of meaningful parameters is intrinsically dependent on the actual surface.

We propose a quasi-minimal model which self-adapts its set of parameters to the actual surface. More precisely, a varying number of rules is used jointly with smoothness constraints to bend a flat mesh, generating the sought-after surface.

We propose an algorithm for fitting this model to multiple images by minimizing the point-based reprojection error. Experimental results are reported, showing that our model fits real images accurately.

1 Introduction

The behaviour of the real world depends on numerous physical phenomena. This makes general-purpose computer vision a tricky task and motivates the need for prior models of the observed structures, *e.g.* [1, 4, 8, 10]. For instance, a 3D morphable face model makes it possible to recover camera pose from a single face image [1].

This paper focuses on paper-like surfaces. More precisely, we consider paper as an unstretchable surface with everywhere vanishing Gaussian curvature. This holds if smooth deformations only occurs. This is mathematically modeled by developable surfaces, a subset of ruled surfaces. Broadly speaking, there are two modeling approaches. The first one is to describe a continous surface by partial differential equations, parametric or implicit functions. The second one is to describe a mesh representing the surface with as few parameters as possible. The number of parameters must thus adapts to the actual surface. We follow the second approach since we target at computationally cheap fitting algorithms for our model.

One of the properties of paper-like surfaces is inextensibility. This is a nonlinear constraint which is not obvious to apply to meshes, as Figure 1 illustrates. For instance, Salzmann *et al.* [10] use constant length edges to generate training meshes from which a generating basis is learnt using Principal Component Analysis. The nonlinear constraints are re-injected as a penalty in the eventual fitting cost function. The main drawback of this approach is that the model does not guarantee that the generated surface is developable.

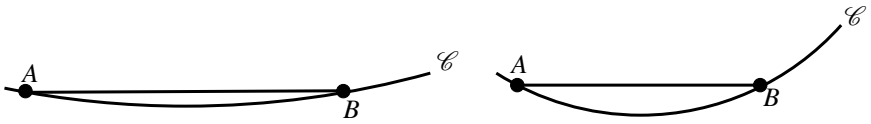


Figure 1: Inextensibility and approximation: A one dimensional example. The curve $\mathscr{C}$ represents an inextensible object, A and B are two points lying on it. Linearly approximating the arc (AB) leads to the segment AB. When $\mathscr{C}$ bowes, although the arc length (AB) remains constant, the length of the segment AB changes. A constant length edge model is thus not a valid parameterization for inextensible surfaces.

We propose a model generating a 3D mesh satisfying the above mentioned properties, namely inextensibility and vanishing Gaussian curvature at any point of the mesh. The model is based on bending a flat surface around rules together with an interpolation process leading to a smooth surface mesh. We only assume a convex object shape. The number of parameters lies very close to the minimal one. This model is suitable for image fitting applications and we describe an algorithm to recover the deformations and rigid pose of a paper-like object from multiple views.

Previous work. The concept of developable surfaces is usually chosen as the basic modeling tool. Most work uses a continuous representation of the surface [3, 4, 7, 9]. They are thus not well adapted for fast image fitting, except [4] which initializes the model parameters with a discrete system of rules. [11] constructs developable surfaces by partitioning a surface and curving each piece along a generalized cone defined by its apex and a cross-section spline. This parameterization is limited to piecewise generalized cones. [6] simulates bending and creasing of virtual paper by applying external forces on the surface. This model has a lot of parameters since external forces are defined for each vertex of the mesh. A method for undistorting paper is proposed in [8]. The generated surface is not developable due to a relaxation process that does not preserve inextensibility.

Roadmap. We present our model in §2 and its construction from multiple images in §3. Experimental results on image sequences are reported in §4. Finally, §5 gives our conclusions and some further research avenues.

2 A Quasi-Minimal Model

We present our model and its parameterization. The idea is to fold a flat mesh that we assume rectangular for sake of simplicity. We underline however that our model deals with any convex shape for the boundary.

2.1 Principle

Generating a surface mesh using our model has two main steps. First, we bend a flat mesh around 'guiding rules'. Second, we smooth its curvature using interpolated 'extra rules', as illustrated in Figure 2. The resulting mesh is piecewise planar. It is guaranteed to be admissible, in the sense that the underlying surface is developable.

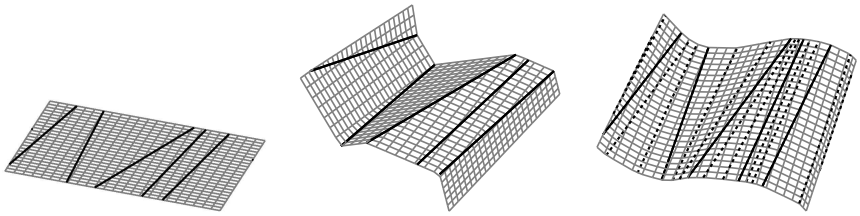
Step 1: Bending with guiding rules. A ruled surface is defined by a differentiable space curve $\alpha(t)$ and a vector field $\beta(t)$, with t in some interval I, see e.g. [11]. Points on the surface are given by:

$$X(t,v) = \alpha(t) + v\beta(t)\ , \quad t \in I \quad v \in \mathbb{R} \quad \beta(t) \neq 0. \tag{1}$$

The surface is actually generated by the line pencil $(\alpha(t), \beta(t))$. This formulation is continuous.

Since our surface is represented by a mesh, we only need a discrete system of rules (sometimes named generatrices), at most one per vertex of the mesh. Keeping all possible rules leads to a model with a high number of parameters, most of them being redundant due to surface smoothness. In order to reduce the number of parameters, we use a subset of rules: The guiding rules. Figure 2 (left) shows the flat mesh representing the surface with the selected rules. We associate an angle to each guiding rule and bend the mesh along the guiding rules accordingly. Figure 2 (middle) shows the resulting guiding mesh. The rules are choosen such that they do not to intersect each other, which corresponds to the modeling of smooth deformations.

Step 2: Smoothing with extra rules. The second step is to smooth the guiding mesh. To this end, we hallucinate extra rules from the guiding ones, thus keeping constant the number of model parameters. This is done by interpolating the guiding rules. The folding angles are then spread between the guiding rules and the extra rules, details are given in the next section. Figure 2 (right) shows the resulting mesh.



Flat mesh Guiding mesh Smoothed mesh

Figure 2: Surface mesh generation. (left) Flat mesh with guiding rules (in black). (middle) Mesh folded along the guiding rules. (right) Mesh folded along the guiding and extra rules.

2.2 Parameterization

A guiding rule i is defined by its two intersection points A_i and B_i with the mesh boundary. Points A_i and B_i thus have a single degree of freedom each. A minimal parameterization is their arc length along the boundary space curve. Since the rules do not intersect each other on the mesh, we define a 'starting point' P_s and an 'ending point' P_e such that all rules can be sorted from P_s to P_e, as shown on Figure 3 (left). Points A_i (resp. B_i) thus have an increasing (resp. decreasing) arc length parameter. The set of guiding rules is parameterized by two vectors s_A and s_B which contain the arc lengths of points A_i and B_i respectively. The non intersecting rules constraint is easily imposed by enforcing monotonicity on vectors s_A and s_B.

As explained above, the model is smoothed by adding extra rules. This is done by interpolating the guiding rules. Two piecewise cubic Hermite interpolating polynomials are computed from the two vectors s_A and s_B. They are called f_A and f_B. This interpolation function has the property of preserving monotonicity over ranges, as required. Figure 3 (right) shows these functions and the control points s_A and s_B. The bending angles are interpolated with a spline and rescaled to account for the increasing number of rules.

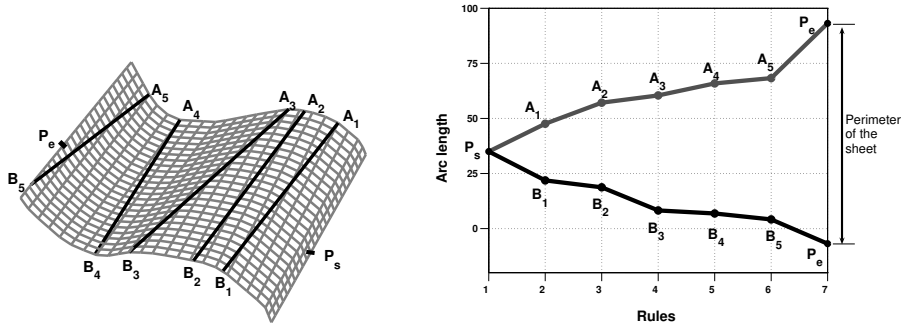


Figure 3: (left) The generated mesh with the control points (A_i, B_i). (right) Arc lengths s_A and s_B of the control points with the interpolating functions f_A and f_B.

Table 1 summarizes the model parameters. The model has $4 + S + 3n$ parameters, S being the number of parameters describing the mesh boundary (for instance, width and height in the case of a rectangular shape) and n being the number of guiding rules.

Parameters	Description	Size
n	number of guiding rules	1
n_e	number of extra rules	1
$\mathscr{S}$	mesh boundary parameters	S
P_s	arc length of the 'starting point'	1
P_e	arc length of the 'ending point'	1
s_A	arc lengths of the first point defining the guiding rules	n
s_B	arc lengths of the second point defining the guiding rules	n
θ	bending angles along the guiding rules	n

Table 1: Summary of the model parameters. (top) Discrete parameters (kept fixed during nonlinear refinement step). (bottom) Continuous parameters.

The deformation is parameterized by the guiding rules. Those are sorted from the 'starting point' to the 'ending point', making wavy the deformation.

We define a directrix as a curve on the surface that crosses some rules once. A minimal comprehensive set of directrices has the least possible number of directrices such that each rule is crossed by exactly one directrix. It is obvious that this set is reduced to a single curve for our model, linking the 'starting point' to the 'ending point'. Consequenctly surfaces requiring more than one directrix can not be generated by our model, as for example a sheet with the four corners pulled up. The model however shows to be experimentally very effective.

3 A Multiple View Fitting Algorithm

Our goal is to fit the model to multiple images. We assume that a 3D point set and camera pose have been reconstructed from image point features by some means. We use the reprojection error as an optimization criterion. As is usual for dealing with such a nonlinear criterion, we compute a suboptimal initialization that we iteratively refine.

3.1 Initialization

We begin by reconstructing a surface interpolating the given 3D points. A rule detection process is then used to infer our model parameters.

Step 1: Interpolating surface reconstruction. Details about how the 3D points are reconstructed are given in §4.1. The interpolating surface is represented by a 2D to 1D Thin-Plate Spline function [2], mapping some planar parameterization of the surface to point height. Defining a regular grid on the image thus allows us to infer the points on the 3D surface. Figure 4 and Figure 6 show two examples.

Step 2: Model initialization by rule detection. The model is initialized from the 3D surface. The side length is choosen as the size of the 3D mesh.

Guiding rules must be defined on the surface. This set of n rules must represent the surface as accurately as possible. In [3] an algorithm is proposed to find a rule on a given surface. It is a method that tries rules on several points on the surface with varying direction. We use it to compute rules along sites lying on the diagonal, the horizontal and the vertical axes. These sites are visible on Figure 4.

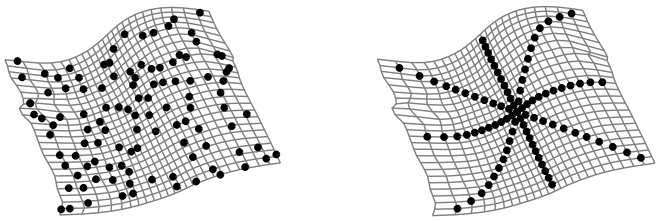


Figure 4: Model initialization. (left) Reconstructed 3D points and the interpolating surface. (right) Points where rules are sought.

The rules are described by the arc length of their intersection points with the mesh boundary. The two arc lengths defining a rule i can be interpreted as a point R_i in $\mathbb{R}^2$, as shown in Figure 5 . Our goal is now to find the vectors s_A and s_B which define the guiding rules, such that their interpolating functions f_A and f_B, defining the parametric curve (f_A, f_B) in $\mathbb{R}^2$, describe the rules. We thus compute s_A and s_B such that the distance between the curve (f_A, f_B) and the points R_i is minimized. We fix the number of guiding rules by hand, but a model selection approach could be used to determine it from the set of rules.

This gives the n guiding rules. The bending angle vector θ is obtain from the 3D surface by assuming it is planar between two consecutive rules. The initial suboptimal model we obtain is shown on Figure 6.

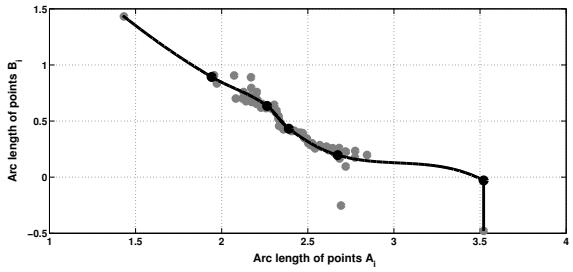


Figure 5: The points in gray represent the detected rules. The black curve is the parametric curve (f_A, f_B) and the black points are the estimated controls points that define the initial rules.

3.2 Refinement

The reprojection error describes how well the model fits the actual data, namely the image feature points. We thus introduce latent variables representing the position of each point onto the modeled mesh with two parameters. Let L be the number of images and N the number of points, the reprojection error is:

$$e = \sum_{i=1}^{N} \sum_{j=1}^{L} (m_{j,i} - \Pi(C_j, M(S, x_i, y_i)))^2. \quad (2)$$

In this equation, $m_{j,i}$ is the i-th feature point in image j, $\Pi(C,M)$ projects the 3D point M in the camera C and $M(S,x_i,y_i)$ is a parameterization of the points on the surface, with S the surface parameters. The points on the surface are initialized by computing each (x_i,y_i) such that their individual reprojection error is minimized, using initial surface model.

To minimize the reprojection error, the following parameters are tuned: The parameters of the model (the number of guiding and extra rules is fixed), see Table 1, the pose of the model (rotation and translation of the generated surface) and the 3D point parameters.

The Levenberg-Marquardt algorithm [5] is used to minimize the reprojection error. Upon convergence, the solution is the Maximum Likelihood Estimate under the assumption of an additive *i.i.d.* Gaussian noise on the image feature points.

4 Experimental Results

We demonstrate the representational power of our fitting algorithm on several sets of images. First, we present the computation of a 3D point cloud. Second, we show the results for the three objects we modeled. Third, we propose some augmented reality illustrations.

4.1 3D Points Reconstruction

The 3D point cloud is generated by triangulating point correspondences between several views. These correspondences are obtained while recovering camera calibration and pose using Structure-from-Motion [5]. Points off the object of interest and outliers are removed by hand. Figure 4 shows an example of such a reconstruction.

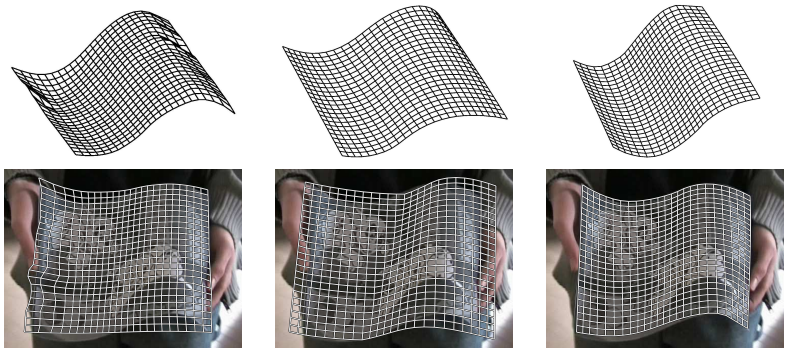


Figure 6: (top) 3D surfaces. (bottom) Reprojection into images. (left) Interpolated surface. (middle) Initialized model. (right) Refined model.

4.2 Model Fitting

Even if our algorithm deals with several views, the following results have been performed with two views. Figure 6 and Figure 7 show the 3D surfaces, their reprojection into images and the reprojection errors distribution for the paper sequence after the three main steps of our algorithm: The reconstruction (left), the initialization (middle) and the refinement (right). Although the reconstruction has the lowest reprojection error, the associated surface is not satisfying, since it is not enough regular and does not fit the borders of the sheet. The initialization makes the model more regular, but is not enough accurate to fit the boundary of the paper, so that important reprojection errors remain. At last, the refined model is visually acceptable and its reprojection error is very close to the reconstructed one. It means that our model accurately fits the image points, while being governed by a much lower number of parameters than the set of independant 3D points has. Moreover the reprojection error significantly decreases thanks to the refinement step, which validates relevance of this step.

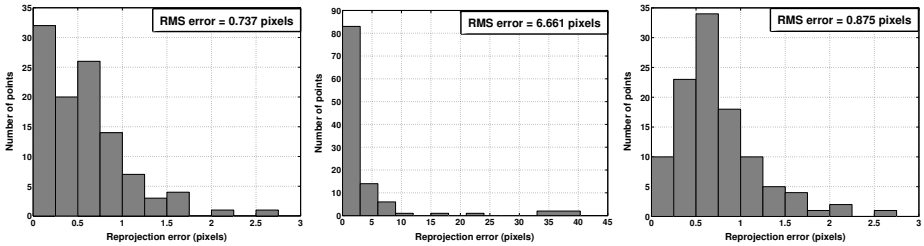


Figure 7: Reprojection errors distribution for the images shown in Figure 6. (left) 3D point cloud. (middle) Initial model. (right) Refined model.

We have tested our method on images of a poster. The results are shown in Figures 8. The reprojections of the computed model are acceptable: The reprojection error of the reconstruction is 0.35 pixels and the one for the refined model is 0.59 pixels.

At last, we fit the model to images of a rug. Such an object does not really satisfy the constraints of developable surfaces. Nevertheless, it is stiff enough to be well-

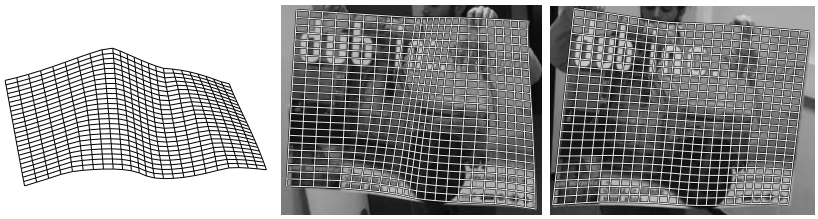


Figure 8: Poster mesh reconstruction. (left) Estimated Model. (middle) Reprojection onto the first image. (right) Reprojection onto the second image.

approximated by our model. The results are thus slightly less accurate than for the paper and the poster: The reprojection error of the reconstruction step is 0.34 pixels and the one of the final model is 1.36 pixels. Figure 9 shows the reprojection of the model onto the images used for the reconstruction.

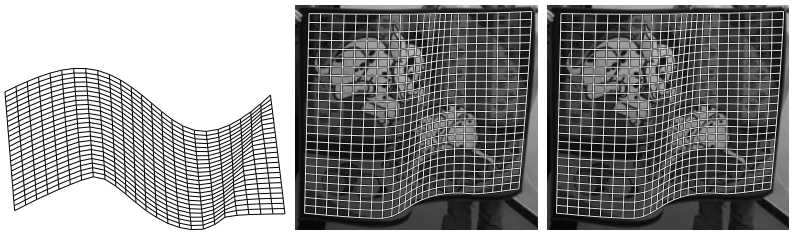


Figure 9: Rug mesh reconstruction. (left) Estimated Model. (middle) Reprojection onto the first image. (right) Reprojection onto the second image.

4.3 Applications

We demonstrate the proposed model and fitting algorithm by unwarping and augmenting images, as shown on Figures 10 and 11. Knowing where the paper is projected onto the images allows us to change the texture map or to overlay some pictures. The augmenting process is described in Table 2. Since we estimate the incoming lighting, the augmented images look realistic.

AUGMENTING IMAGES

1. Run the proposed algorithm to fit the model to images
2. Unwarp one of the images chosen as the reference one to get the texture map
3. Augment the texture map
4. For each image automatically do
   - 4.1 Estimate lighting change from the reference image
   - 4.2 Transfer the augmented texture map

Table 2: Overview of the augmenting process.

Figure 10: Some applications. (left) Unwarped texture map of the paper. (middle) Changing the whole texture map. (right) Augmented paper.

Figure 11: Augmentation. (left) Augmented unwarped texture map. (middle) Augmented texture map in the first image. (right) Synthetically generated view of the paper with the augmented texture map.

5 Conclusion and Future Work

This paper describes a quasi-minimal model for paper-like objects and its estimation from multiple images. Although there are few parameters, the generated surface is a good approximation of smoothly deformed paper-like objects. This is demonstrated on real image sequences thanks to a fitting algorithm which initializes the model first and then refines it in a bundle adjustment manner.

There are many possibilities for further research. The proposed model could be embedded in a monocular tracking framework or used to generate sample meshes for a learning-based model construction.

We currently work on alleviating the model limitations mentioned earlier, namely handling a general boundary shape and the comprehensive set of feasible deformation.

References

[1] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(9), September 2003.

[2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[3] H.-Y. Chen and H. Pottmann. Approximation by ruled surfaces. *Journal of Computational and Applied Mathematics*, 102:143–156, 1999.

[4] N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *Proceedings of the European Conference on Computer Vision*, 2004.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[6] Y. Kergosien, H. Gotoda, and T. Kunii. Bending and creasing virtual paper. IEEE *Computer Graphics & Applications*, 14(1):40–48, 1994.

[7] S. Leopoldseder and H. Pottmann. Approximation of developable surfaces with cone spline surfaces. *Computer-Aided Design*, 30:571–582, 1998.

[8] M. Pilu. Undoing page curl distortion using applicable surfaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, December 2001.

[9] H. Pottmann and J. Wallner. Approximation algorithms for developable surfaces. *Computer Aided Geometric Design*, 16:539–556, 1999.

[10] M. Salzmann, S. Ilic, and P. Fua. Physically valid shape parameterization for monocular 3-D deformable surface tracking. In *Proceedings of the British Machine Vision Conference*, 2005.

[11] M. Sun and E. Fiume. A technique for constructing developable surfaces. In *Proceedings of Graphics Interface*, pages 176–185, May 1996.

The Morphlet Transform: A Multiscale Representation for Diffeomorphisms

Jonathan R. Kaplan, David L. Donoho
Dept. of Mathematics, Statistics
Stanford University
Stanford, CA, 94305-2125, USA
jkaplan@math.stanford.edu

Abstract

We describe a multiscale representation for diffeomorphisms. Our representation allows synthesis – e.g. generate random diffeomorphisms – and analysis – e.g. identify the scales and locations where the diffeomorphism has behavior that would be unpredictable based on its coarse-scale behavior. Our representation has a forward transform with coefficients that are organized dyadically, in a way that is familiar from wavelet analysis, and an inverse transform that is nonlinear, and generates true diffeomorphisms when the underlying object satisfies a certain *sampling condition*.

Although both the forward and inverse transforms are nonlinear, it is possible to operate on the coefficients in the same way that one operates on wavelet coefficients; they can be shrunk towards zero, quantized, and can be randomized; such procedures are useful for denoising, compressing, and stochastic simulation. Observations include: (a) if a template image with edges is morphed by a complex but known transform, compressing the morphism is far more effective than compressing the morphed image. (b) One can create random morphisms with and desired self-similarity exponents by inverse transforming scaled Gaussian noise. (c) Denoising morpishms in a sense smooths the underlying level sets of the object.

1 Introduction

Temporal or spacial deformation is a common underlying source of variability in many signal and image analysis problems. This deformation may be the result of measurement distortions, as in the case of satellite imagery [1] and GC/MS data [12] or the deformation may be the actual phenomenon of study [9, 11]. In the first case, the deformation is seen as a nuisance and must be removed before further analysis. In the second case, the goal is not the removal of the deformation but rather the extraction of the deformation. Once the deformation has been extracted, understanding the phenomenon at hand consists of analyzing the deformation itself. This paper presents a novel representation for the deformation after extraction that takes advantage of smoothness and multiscale organization to both ease the computational burden of analysis and reveal geometric structure.

For the problem of deformation analysis we will limit ourselves to deformations that are diffeomorphisms–special deformations that are invertible, smooth, and have smooth inverses – this specialization allows us to limit the problem to one that is mathematically amenable. Although they are one of the basic building blocks of modern theoretical mathematics, and an everyday object in pure mathematics, diffeomorphisms are a new data type in image and signal analysis. This new data type needs a representation native to its own structures. This representation should allow for fast computation and storage while maintaining as much transparency as possible. In this paper we present one possible approach to addressing this need.

We present a novel nonlinear invertible multiscale transform on the space of diffeomorphisms that can be used to store, manipulate, and analyze the variability in a collection of signals that are all diffeomorphic to a given template. This multiscale transform is known as the *Morphlet Transform*. The use of such a transform is motivated by the success of multiscale methods in data compression [5], noise removal [4], and fast template registration [7]. Many realistic diffeomorphisms that appear in image and signal analysis are sparsely represented in the Morphlet transform space.

There is one chief obstacle to representing diffeomorphisms: the space of diffeomorphisms is not a linear function space. Diffeomorphisms are functions, and so any given diffeomorphism may be expressed in a basis expansion. But perturbations or manipulations of the expansion coefficients may produce, after reconstruction, a function that is no longer a diffeomorphism. This will be true regardless of the basis functions that are used.

There are two typical strategies for representing diffeomorphisms. The first is to restrict to a parametric family of functions like affine maps, polynomials, or thin-plate splines. Although these methods are attractive due to their simplicity, with the exception of affine maps, none of these methods can guarantee that the matching function is in fact a diffeomorphism. The second strategy is to use a deformation vector field such as in [6]. The diffeomorphism is then the unit time flow induced by the vector field. The vector field representation has three drawbacks: first, calculating features of the full diffeomorphism involves solving a first order PDE, which can be computationally intensive; second, the representation is global, so the values of the vector field outside of some region can effect the value of the diffeomorphism in the region; third, because of the global nature of the vector field the multiscale structure of the field may not correspond to the multiscale structure of the diffeomorphism. In contrast, the Morphlet transform offers a local and computationally efficient description for diffeomorphisms that ensures that the matching functions are, in fact, diffeomorphisms.

The Morphlet transform takes uniform samples of the diffeomorphism on the standard tetrahedral lattice and returns a collection of coefficients indexed by scale and location. These coefficients have the same organizational structure as the coefficients produced by the wavelet transform. Like the wavelet transform, the Morphlet coefficients reflect the deviation of the diffeomorphism at one scale and location from the behavior predicted from the next coarser scale values. Unlike the wavelet transform, a given coefficient no longer corresponds to an element of a basis set. Rather, both decomposition and reconstruction are nonlinear.

1.1 Related Work

There is now a long history of the use of multiscale methods in image registration and deformation [8]. On the question of representing diffeomorphism, D. Mumford and E. Sharon have worked on the question of the conformal fingerprint for conformal diffeomorphisms between two simply connected regions of the plane [11]. T.F. Cootes *et al.*[2] work on building a parametric statistical model on the space of diffeomorphisms. A great deal of interest in pattern analysis using diffeomorphic registration has been generated in the field of computational anatomy. See [6, 9].

2 Template Warping and Diffeomorphisms

Before we continue, let us define some terms and make precise the problem we wish to address. For our purposes, a diffeomorphism is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}^n$ which is one to one, onto, and has a smooth inverse. We shall model signals and images as real-valued functions on $\mathbb{R}^n$. The deformation of a signal will be the action of a diffeomorphism on the signal.

$$I_{\text{deform}}(x) = \phi^*(I)(x) = I \circ \phi^{-1}(x). \tag{1}$$

We assume we have a collection of signals $\{I_n\}_{n=1}^N$ and a fixed template signal I_{template} such that for each signal I_n there exists a diffeomorphism ϕ_n such that:

$$I_n = \phi_n^*(I_{\text{template}}). \tag{2}$$

We will not address the question of how to calculate ϕ_n given I_n and I_{template}. There is a large existing literature devoted to solving the diffeomorphic registration problem [10], and we postpone discussing the relationship between the morphlet transform and registration algorithms for a later publication. We will simply assume that we can calculate ϕ_n satisfying (2) or some appropriate regularization.[1]

Once the set of registering diffeomorphisms, $\{\phi_n\}_{n=1}^N$, has been obtained, all of the information contained in the sample $\{I_n\}_{n=1}^N$ is now contained (up to the accuracy of the diffeomorphic assumption) in the set of registering diffeomorphisms and the template I_{template}. If we want to study the variability of $\{I_n\}_{n=1}^N$ we need only study the variability of $\{\phi_n\}_{n=1}^N$. This is advantageous due to the smoothness of the registering diffeomorphisms.

Typically, the measured signals are not smooth. In 1-d, the signals may have jumps and spikes. In 2-d, the images have edges and textures. Representing these features can be difficult, and most of the science of image analysis focuses on ways of dealing with these non-smooth features. In contrast, the registering diffeomorphisms are frequently very smooth functions with a few localized regions of sharp transition. These regions of sharp transition in the registering diffeomorphisms are exactly the regions responsible for the variability in the collection of signals $\{I_n\}_{n=1}^N$.

Bijections that are smooth except for a few local singularities have sparse Morphlet transforms. Thus, even when analyzing images with high spatial resolution, only a small fraction of the morphlet coefficients of the resulting registering diffeomorphisms will be large. Only these large coefficients are important. Morphlet transform preserves diffeomorphisms, and has approximation properties similar to the wavelet transform. Thus, if

[1] In the presence of noise, (2) is never satisfied. Rather, registration algorithms typically search for a diffeomorphism that satisfies a regularized least squares problem.

we threshold and discard small coefficients, the reconstructed diffeomorphism will have a sparse representation and will be very close to the original diffeomorphism. We will give examples of such a reconstruction in section 5.2.

3 The Interpolating Wavelet Transforms

The Morphlet transform is a nonlinear variant of the wavelet transform. We do not have the space to give a thorough introduction to the theory of linear wavelet transforms, but because the Morphlet transform explicitly builds off of the interpolating wavelet transform [3] we will briefly describe its construction and properties. For ease of presentation we will only discuss the one dimensional case; the higher dimensional case is similar.

The linear interpolating wavelet transform is defined on the dyadic samples of a continuous real-valued function. Let f be a continuous function on $\mathbb{R}$. Fix integers J_0 and J_1 which will serve as the coarsest and finest dyadic sampling scales respectively. Sample f at $\frac{k}{2^{J_1}}$ for $k \in \mathbb{Z}$. Define β_k^j as:

$$\beta_k^j = f(\frac{k}{2^j}) \tag{3}$$

for $J_0 \leq j \leq J_1$.

Fix a positive odd integer D. And define the prediction Pred_k^j as:

$$\text{Pred}_k^j = \pi_k^j(\frac{k}{2^j} + \frac{1}{2^{j+1}}), \tag{4}$$

where π_k^j is a local interpolating polynomial of order D. Specifically,

$$\pi_k^j \text{ interpolates the values } (\frac{k+i}{2^j}, \beta_{k+i}^j) \text{ for } i = -\frac{D-1}{2}, \ldots, \frac{D+1}{2}. \tag{5}$$

We define the linear interpolating wavelet transform as the set of coefficients:

$$\left\{\beta_k^{J_0}, \alpha_k^{j,\text{linear}}\right\}_{j,k\in\mathbb{Z}} \tag{6}$$

where

$$\alpha_k^{j,\text{linear}} = \beta_{2k+1}^{j+1} - \text{Pred}_k^j. \tag{7}$$

Thus, $\alpha_k^{j,\text{linear}}$ measures the discrepancy between the true value of the function and the value predicted by using the samples at the next coarser scale. The coefficients $\{\beta_k^{J_0}\}$ are called the coarse scale coefficients and $\{\alpha_k^{j,\text{linear}}\}$ are called the detail coefficients. In regions where f is smooth, the detail coefficients decay exponentially in j. The rate of decay measures the degree of smoothness of f. If a function is smooth everywhere except for a few isolated singularities, then the fine scale coefficients of the wavelet transform will be very small away from the singularities.

To invert the transform we employ a pyramidal scheme starting with the coarsest sample $\beta_k^{J_0}$. For each scale j, we predict the values at the next finer scale $j+1$ using (4) and reconstruct the samples at scale $j+1$ using:

$$\beta_{2k+1}^{j+1} = \alpha_k^{j,\text{linear}} + \text{Pred}_k^j. \tag{8}$$

Both the forward and inverse transforms involve $O(N)$ flops, where N is the number of samples of f.

4 The Morphlet Transform

The Morphlet transform acts on the dyadic samples of a continuous diffeomorphism of $\mathbb{R}^n$. We will show the 2-d version of the transform, as the simplicity of diffeomorphisms in dimension $n = 1$ makes the 1-d transform insufficiently instructive. The high dimensional versions follow the same pattern as the 2-d transform.

4.1 The Sampling Condition

Due to a sampling condition, the Morphlet transform is only actually defined for a special sub-manifold of the space of diffeomorphisms. The idea behind the sampling condition is to ensure, at each scale, that the reconstructed function "looks like a diffeomorphism." In particular, we demand that the discrete Jacobians of the samples of the diffeomorphism are all positively oriented affine maps at all scales. For any given diffeomorphism there exists a dyadic scale J such that for all scales finer than J the discrete Jacobians satisfy this condition. Thus, a diffeomorphism needs to be sufficiently finely sampled before the Morphlet transform may be applied.

Let ϕ be a diffeomorphism of the plane. As in the linear case define $\beta^j_{k,l}$:

$$\beta^j_{k,l} = \phi(\frac{k}{2^j}, \frac{l}{2^j}) \tag{9}$$

To clarify the sampling condition, and ease the notation for the definition of the fine scale coefficients, we define three intermediate sets of samples, $\{\beta^{j,0}_{k,l}\}$, $\{\beta^{j,1}_{k,l}\}$, and $\{\beta^{j,2}_{k,l}\}$.

$$\beta^{j,3}_{k,l} = \beta^j_{k,l} \tag{10}$$

$$\beta^{j+1,1}_{2k,2l} = \beta^{j+1,2}_{2k,2l} = \beta^{j,3}_{k,l} \tag{11}$$

$$\beta^{j+1,1}_{2k+1,2l+1} = \beta^{j+1,2}_{2k+1,2l+1} = \frac{1}{2}(\beta^{j,3}_{k+1,l} + \beta^{j,3}_{k,l+1}) \tag{12}$$

$$\beta^{j+1,1}_{2k+1,2l} = \frac{1}{2}(\beta^{j,3}_{k+1,l} + \beta^{j,3}_{k,l}) \ , \beta^{j+1,2}_{2k+1,2l} = \beta^{j+1,3}_{2k+1,2l} \tag{13}$$

We say the diffeomorphism *satisfies the sampling condition* for the scales $j \in [J_0, J_1]$ if the following condition is satisfied:

Discrete Bijectivity Constraint *For all $j \in [J_0, J_1]$, $(k,l) \in \mathbb{Z}^2$, $n = 0, 1, 2$,*

$$\delta \in \Delta = \{ \pm \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, \pm \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} \} : \tag{14}$$

$$\text{sign}\big(\det[\beta^{j,n}_{k,l} - \beta^{j,n}_{k+\delta_{1,2},l+\delta_{2,2}}, \beta^{j,n}_{k+\delta_{1,1},l+\delta_{2,1}} - \beta^{j,n}_{k+\delta_{1,2},l+\delta_{2,2}}]\big) = \text{sign}\big(\det(\delta)\big). \tag{15}$$

4.2 Defining the Morphlet Coefficients

To define the fine scale coefficients we begin by fixing a coarsest and a finest dyadic scale J_0 and J_1 for which ϕ satisfies the *Discrete Bijectivity Constraint*. As in the interpolating wavelet transform, we also fix an odd integer D which will serve as the order of an underlying linear interpolating wavelet transform as in section 3. In addition, we fix a sequence of exponentially decaying integers λ_j for $j = J_0, J_0 + 1, \ldots, J_1$. The order of the transform

D and the rate of decay of λ_j will determine the relationship between the decay of the Morphlet coefficients and the smoothness of the diffeomorphism.

For $J_0 \leq j \leq J_1$ and Δ as in (14), we first define the boundary penalty terms as:

$$\Lambda_i^j(k,l) = \sum_{\delta \in \Delta} \frac{\beta^{j,i}_{k+\delta_{1,1},l+\delta_{1,2}} - \beta^{j,i}_{k+\delta_{1,2},l+\delta_{2,2}}}{\det[\beta^j_{k,l} - \beta^{j,i}_{k+\delta_{1,2},l+\delta_{2,2}}, \beta^{j,i}_{k+\delta_{1,1},l+\delta_{1,2}} - \beta^{j,i}_{k+\delta_{1,2},l+\delta_{2,2}}]}. \tag{16}$$

Then we define the fine scale coefficients

$$\alpha^j_{2k+1,2l} = \alpha^{j,\text{linear}}_{2k+1,2l} - \lambda_j \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Lambda_1^{j+1}(2k+1,2l) \tag{17}$$

$$\alpha^j_{2k,2l+1} = \alpha^{j,\text{linear}}_{2k,2l+1} - \lambda_j \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Lambda_2^{j+1}(2k,2l+1) \tag{18}$$

$$\alpha^j_{2k+1,2l+1} = \alpha^{j,\text{linear}}_{2k+1,2l+1} - \lambda_j \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Lambda_3^{j+1}(2k+1,2l+1) \tag{19}$$

The Morphlet transform for ϕ is then:

$$\mathscr{M}(\phi) = \{\beta^{J_0}_{k,l}, \alpha^j_{k,l}\}. \tag{20}$$

Note that the detail coefficients of the Morphlet transform are perturbed versions of the linear interpolating wavelet coefficients. For a smooth diffeomorphism, the difference between the linear coefficients and Morphlet coefficients at fine scales is $O(\lambda_j)$. Thus, the exponential decay of λ_j indicates that the fine scale coefficient of both transforms are very similar. The biggest difference comes in the coarse and medium scales, where the perturbation can be large relative to λ_j. The perturbation is large when, due to a Gibbs' phenomenon, the local polynomial interpolation of the diffeomorphism has a vanishing Jacobian. Under these circumstances, the perturbation can dominate the coefficient.

4.3 Basic Properties of the Morphlet Transform

- Functions reconstructed with the inverse transform will always be discrete diffeomorphisms. On the set of diffeomorphisms that satisfy the *Discrete Bijectivity Constraint*, the Morphlet transform is invertible. Both transforms require $O(N)$ flops where N is the number of samples, though the inverse transform requires more work as it requires the use of Newton's method or another similar nonlinear solver.
- If ϕ is an affine map then $\mathscr{M}_{\text{detail}}(\phi) = 0$. In particular, the detail coefficients of the identity map vanish.
- If $\tau \in \mathbb{R}^2$ then $\mathscr{M}_{\text{coarse}}(\phi + \tau) = \mathscr{M}_{\text{coarse}}(\phi) + \tau$ and $\mathscr{M}_{\text{detail}}(\phi + \tau) = \mathscr{M}_{\text{detail}}(\phi)$. The detail coefficients are invariant under translation.
- If $A \in SO_n$, $\alpha^j_{k,l}(A\phi) = A\alpha^j_{k,l}(\phi)$. The detail coefficients are covariant under linear orthogonal transformation.
- If $\Omega \subset \text{dom}(\phi)$ is a open domain such that $\phi|_\Omega$ and $\phi^{-1}|_{\phi(\Omega)}$ are smooth then all Morphlet coefficients with support[2] in Ω decay geometrically as a function of the

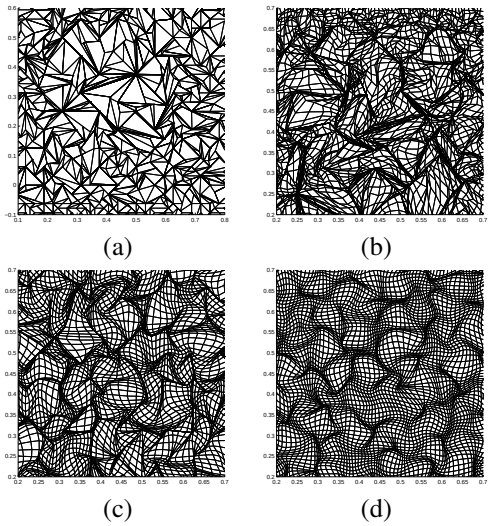
[2]The support of a Morphlet coefficient is the collection of all points in the domain of the bijection which appear in the penalty term (16) and the linear coefficient (7) for the corresponding formula (17) - (19).

discrete scale index. The rate of decay is determined by the smoothness of the refinement scheme and the smoothness of ϕ.

- If $\lambda_j = O(2^{-(D+1)j})$, the approximation rate of the Morphlet transform is as good as the approximation rate of the associated wavelet transform.

5 Stylized Applications

5.1 Random Diffeomorphisms



(a) (b) (c) (d)

Figure 1: (a) $\omega = 1$, (b) $\omega = 2$, (c) $\omega = 4$, (d) $\omega = 5$.

The inverse Morphlet transform can be used to generate random diffeomorphisms of any preselected smoothness: first, generate a random affine diffeomorphism[3]; second, subsample the affine map and use the samples as the coarse scale coefficients; third, randomly generate detail coefficients with a preselected decay; finally, apply the inverse transform. In particular, one can choose uncorrelated random Gaussian coefficients with scale-dependents standard deviations $\alpha_{k,l}^j \sim N(0, C2^{-\omega j})$ for some fixed $C, \omega > 0$. The larger ω, the smoother the random diffeomorphism. Figure 1 shows the action of four random diffeomorphisms on a rectangular grid. Each diffeomorphism was generated with coarse scale coefficients set to the identity map and with $C = 1$. The value of ω is varied.

5.2 Compression

In the case where one must compress an image consisting of a known template that has been deformed smoothly, note that if the template has sharp features, such as edges, it can be very difficult to compress a warped instance by standard means. Indeed traditional compression schemes will not work well on figure 2(b). However, as panels (c) and (d) in

[3]This is just a matter of linear algebra.

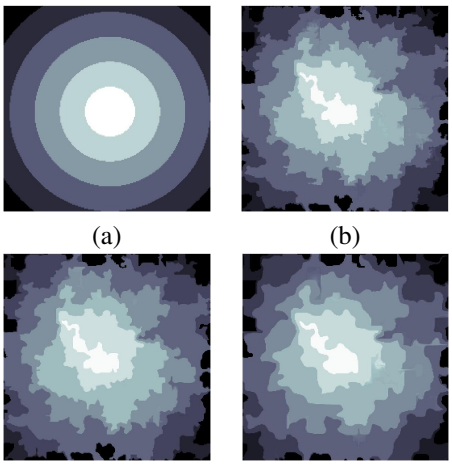


Figure 2: (a) Image I, (b) $\phi^*(I)$, (c) $\phi^*_{10\%}(I)$, Rel. L^2-error $= 9.0e-4$ (d) $\phi^*_{2\%}(I)$, L^2-error $= 1.9e-3$.

our example show, if we take the template as separately known to the compressor and the decompressor, and simply compress the morphism, we can reconstruct the warped image yielding a dramatically better visual fidelity than standard compression could offer.

The Morphlet coefficients of a smooth diffeomorphism decay rapidly. Because the approximation rate of the Morphlet transform is at least as good as the approximation rate of the wavelet transform, we may discard a large percentage of the coefficients and still reconstruct the diffeomorphism with high accuracy. This approximation rate can be used to "compress" an image that is a warped version of a known template. To store the image, we store the template and the registering diffeomorphism. If we have a collection of images all of which are diffeomorphic to the template, then the ability to compress the diffeomorphisms translates into low storage for all of the images in the collection.

In figure 2 we show a simple "bulls-eye" template, (a), and the action of a randomly generated diffeomorphism ϕ on the template, (b). Applying the Morphlet transform to ϕ, we threshold all but the largest 10% of the detail coefficients and apply the inverse transform to construct $\phi_{10\%}$. Similarly, we threshold all but the largest 2% of the detail coefficients to construct $\phi_{2\%}$. We apply both $\phi_{10\%}$ and $\phi_{2\%}$ to I and record the relative L^2-error between $\phi^*(I)$ and $\phi_{10\%}(I)$, $\phi_{2\%}(I)$ respectively. All images are resolved at 512 x 512.

Notice that thresholding the diffeomorphism and applying it to the template effectively smooths the level curves of $\phi(I)$. No linear method can achieve this.

5.3 Interpolation

Given two diffeomorphisms, ϕ_1 and ϕ_0, that are affine maps at the coarsest scale, the Morphlet transform can be used to interpolate between them in the space of diffeomorphisms. To do so, we interpolate between the two affine maps at the coarsest scale[4] and then we

[4]Pull back to the Lie algebra and linearly interpolate.

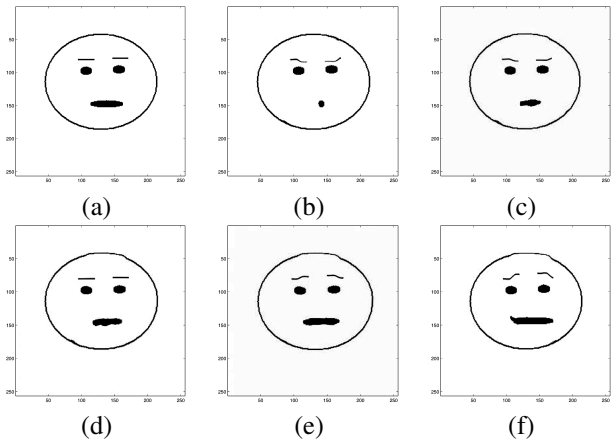


Figure 3: (a) The original template I, (b) $\phi_0^*(I)$, (c) $\phi_{\frac{1}{4}}^*(I)$, (d) $\phi_{\frac{1}{2}}^*(I)$, (e) $\phi_{\frac{3}{4}}^*(I)$, (f) $\phi_1^*(I)$.

simply linearly interpolate the respective detail coefficients of the Morphlet transform:

$$\alpha_{k,l}^j(\phi_t) = t\alpha_{k,l}^j(\phi_1) + (1-t)\alpha_{k,l}^j(\phi_0). \tag{21}$$

Figure 3 shows a simple example of the action of interpolated diffeomorphisms on a template. Figure 3 (a) is the original template, I. Figure 3 (b) and (f) show the action of ϕ_0 and ϕ_1 on the template, respectively, where both diffeomorphisms were synthetically generated by the authors. Both ϕ_0 and ϕ_1 are the identity map at the the coarsest scale. For $t = \frac{1}{4}, \frac{1}{2}$, and $\frac{3}{4}$ we interpolate between the two diffeomorphisms as in (21) and apply the maps to the template.

This interpolation provides a cartoon model for articulated motion. The underlying object–here a cartoon face–undergoes diffeomorphic changes. The Morphlet coefficients act like small control knobs with which we can change the expression of the face.

6 Conclusion

The space of Euclidean diffeomorphisms is highly nonlinear. Yet, there is a large submanifold of diffeomorphisms that satisfy the *Discrete Bijectivity Constraint* and for this submanifold the Morphlet transform acts as an embedding into L^∞ (or $\mathbb{R}^n$ in the finitely sampled case). All of the coefficients are calculated using only local information and, similar to wavelets, diffeomorphisms have sparse Morphlet transforms. The Morphlet transform provides a representation where approximation and manipulation are simple arithmetic operations. Future work will further explore how these properties can be used in image and signal analysis.

7 Acknowledgements

Thanks to: Claire Tomlin and other participants in the ONR-MURI "CoMotion" project that partially supported this work; NSF DMS 0072661, NSF DMS 0140698(FRG) and CNS-0085984 for partial support; Peter Schroeder (CalTech), Victoria Stodden, and Inam

Rahman for discussions about software for Multiscale Analysis of Riemannian Symmetric-Space Valued data; IPAM for its support of the first author during the "Multiscale Geometric Analysis" long program; MSRI for its support of both authors during its "Mathematical, Statistical, and Computational Aspects of Vision" long program. The first author has been supported by an NSF Postdoctoral Fellowship.

References

[1] R. Bernstein. Digital image processing of earth observation sensor data. *IBM Journal of Research and Development*, (20):40–57.

[2] T. F. Cootes, C. J. Twining, and C. J. Taylor. Diffeomorphic statistical shape models. In *Proceedings of BMVC 2004*, volume 1, pages 447–456, 2004.

[3] David Donoho. Interpolating wavelet transforms. Technical report, Stanford University, 1992.

[4] David L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995.

[5] David L. Donoho, Martin Vetterli, R. A. DeVore, and Ingrid Daubechies. Data compression and harmonic analysis. *IEEE Trans. Inform. Theory*, 44(6):2435–2476, 1998. Information theory: 1948–1998.

[6] S. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9(8):1357–1370, August 2000.

[7] Y. Keller and A. Averbuch. Fast motion estimation using bidirectional gradient methods. *Image Processing, IEEE Transactions on*, 13(8):1042 – 1054, August 2004.

[8] S. Kovačič and R. Bajcsy. Multiscale/multiresolution representations. In A. W. Toga, editor, *Brain Warping*, pages 45–65. Academic Press, 1999.

[9] M. I. Miller and U. Grenander. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, LVI(4):617–694, December 1998.

[10] Jan Modersitzki. *Numerical methods for image registration*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2004. Oxford Science Publications.

[11] E. Sharon and D. Mumford. 2d-shape analysis using conformal mapping. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, volume 2, pages 350–357, 2004.

[12] A. Willse, A. Belcher, G. Preti, J. Wahl, M. Thresher, P. Yang, K. Yamazaki, and G. Beauchamp. Identification of major histocompatibility complex-regulated body odorants by statistical analysis of a comparative gas chromatography/mass spectrometry experiment. *Anal. Chem.*, 77(8):2348 –2361, 2005.

Non-linear Registration Between 3D Images Including Rigid Objects: Application to CT and PET Lung Images With Tumors

A. Moreno[1,2] G. Delso[3] O. Camara[4] I. Bloch[1]

Antonio.Moreno@enst.fr

[1] GET-ENST, Dept. TSI, CNRS UMR 5141 LTCI, Paris, France
[2] Segami Corporation, Paris, France [3] Philips Medical Systems, Suresnes, France
[4] Center for Medical Image Computing, University College London, UK

Abstract

This paper deals with the problem of non linear image registration is the case the images include objects undergoing different types of deformation. As an illustrative application, we consider the registration of CT and PET images of thoracic and abdominal regions. Registration of these two modalities has to cope with deformations of the lungs during breathing. Potential tumors in the lungs usually do not follow the same deformations, since they can be considered as almost rigid, and this should be taken into account in the registration procedure. We show in this paper how to introduce rigidity constraints into a non-linear registration method. The proposed approach is based on registration of landmarks defined on the surface of previously segmented objects and on continuity constraints. The results demonstrate a significant improvement of the combination of anatomical and functional images for diagnosis and for oncology applications.

1 Introduction

Registration between several images of the same scene is a widely addressed topic and is important in many different domains. One of the difficult problems concerns the case where the images include objects undergoing different types of deformation that have to be compensated during the registration. In particular, the behavior of the registration close to the interfaces between such objects has to be carefully controlled in order to avoid discontinuities or other unrealistic phenomena. The aim of this paper is to address this problem.

As an illustrative application, we consider Computed Tomography (CT) and Positron Emission Tomography (PET) in thoracic and abdominal regions, which furnish complementary information about the anatomy and the metabolism of human body. Their combination has a significant impact on improving medical decisions for diagnosis and therapy [16] even with combined PET/CT devices where registration remains necessary to compensate patient respiration and heart beating [14]. Registration of these two modalities is

a challenging application due to the poor quality of the PET image and the large deformations involved in these regions.

Most of the existing methods have as a limitation that regions placed inside or near the main structures will be deformed more or less according to the registration computed for the latter, depending on how local is the deformation. A critical example of this situation occurs when a tumor is located inside the lungs and there is a large volume difference between CT and PET images (due to the breathing). In this case, if the tumor is registered according to the transformation computed for the lungs, it may take absurd shapes, such as shown in Figure 1.

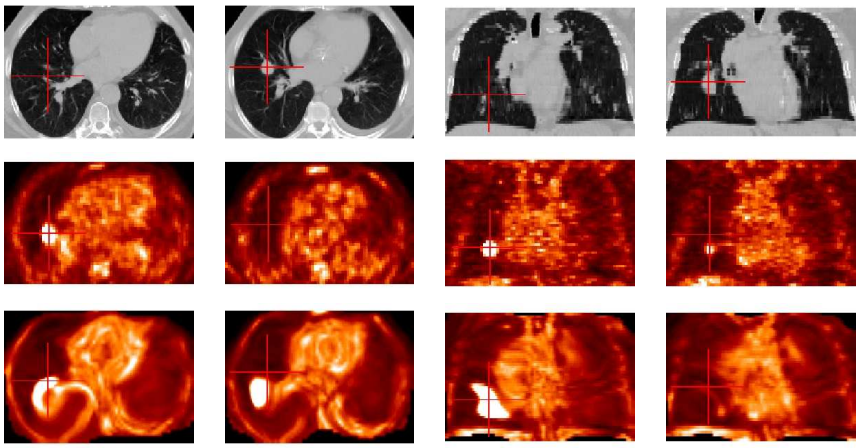


Figure 1: Axial and coronal slices in CT (first row) and in PET (second row). Result of the non-linear registration without tumor-based constraints (third row). The absence of these constraints leads to undesired and irrelevant deformations of the pathology. On the images of the first and third columns, the cursor is positioned on the tumor localization in PET data, while in the second and fourth columns, it is positioned on the tumor localization in CT data. This example shows an erroneous positioning of the tumor and illustrates the importance of the use of tumor-specific constraints.

In this case, two very different deformations exist: the non-linear deformations of the lungs due to the breathing and the linear displacement of the tumor during the breathing cycle. Thus, the aim of this paper is to avoid the undesired tumor misregistrations by adding some rigidity constraints on the tumors. The goal is to preserve tumor geometry and, in particular, intensity since it is critical for clinical studies, for instance based on SUV (Standardized Uptake Value) [6], and for diagnosis and radiotherapy planning.

In Section 2, we summarize existing work related to this subject and we provide an overview of the proposed approach. The introduction of tumor-based constraints into the registration algorithm is detailed in Section 3. Section 4 presents some results obtained on real data. Finally, conclusions and future works are discussed in Section 5.

2 Related Work and Overview of the Proposed Approach

Some approaches have already been developed for registration of multimodality images in pathological cases (pulmonary nodules, cancer), such as in [2]. However these approaches compute a rigid (or affine) registration for all the structures and they do not take into account the local nature of the deformations.

Tanner et al. [15] have developed a method of non-rigid registration based on B-spline Free-Form Deformations (FFD) as in [3]. Their algorithm is applied on MR breast images and it guarantees volume and shape preservation in the rigid regions defined by the lesions. However, the region of the rigid transformation is larger than the lesions. Another approach that uses B-spline FFD is the one by Rohlfing and Maurer [12]. They have used a grid refinement and added some incompressibility constraints (using the properties of the Jacobian) which only guarantee the preservation of the volume of the structures but not their shape. Loeckx et al. [10] have added a local rigidity constraint in order to guarantee shape preservation and they have obtained very promising results. Nevertheless, this algorithm does not enforce the considered structures to be totally rigid, therefore they actually might be slightly deformed.

The recent work of Hachama et al. [7] uses a Bayesian framework in order to characterize the pathologies as outliers of a probabilistic distribution. Their method is applied to mammogram registration and proved to be robust. An implicit assumption is that grey levels in both images are similar, thus making the method appropriate for mono-modality images. This assumption should be relaxed to extend the method to multimodality images.

A different approach, that we consider closer to physical reality of human body, is based on the combination of rigid and non-rigid deformations, as suggested by Little et al. [9] and Huesman et al. [8]. These methods are based on the use of point interpolation techniques, together with a weighting of the deformation according to a distance function. Castellanos et al. [4] developed a slightly different methodology, in which local non-rigid warpings are used to guarantee the continuity of the transformation.

The advantage of these approaches is that they take into account rigid structures and the deformations applied to the image are continuous and smooth. The method we propose is inspired by these ones and adapted to develop a registration algorithm for the thoracic region in the presence of pathologies. In order to illustrate our algorithm, we have applied it on medical data. These data consist of 3D CT and PET images of pathological cases, exhibiting tumors in the lungs. We assume that the tumor is rigid and thus a linear transformation is sufficient to cope with its movements between CT and PET images. This hypothesis is relevant and in accordance with the clinicians' point of view, since tumors are often a compact mass of pathological tissue. In order to guarantee a good registration of both normal and pathological structures, the first step consists of a segmentation of all structures which are visible in both modalities. Then we define two groups of landmarks in both images, which correspond to homologous points, and will guide the deformation of the PET image towards the CT image. The positions of the landmarks are therefore adapted to anatomical shapes. This is an important feature and one of the originalities of our method. The deformation at each point is computed using an interpolation procedure based on the landmarks, on the specific type of deformation of each landmark depending on the structure it belongs to, and weighted by a distance function, which guarantees that the transformation will be continuous.

Thus, the proposed approach has two main advantages:

1. As the transformation near the tumor is reduced by using the distance weight, even if we have some small errors in the tumor segmentation (often quite challenging, mainly in CT), we will obtain a consistent and robust transformation.

2. In the considered application, one important fact is that the objects to register are not the same in the two images. For instance, the volume of the "anatomical" tumor in CT is not necessarily the same as the volume of the "functional" tumor in PET because the two modalities highlight different characteristics of the objects. The registration of these two views of the tumor must preserve these local differences, which can be very useful because we could discover a part of the anatomy that is touched by the pathology and could not be seen in the CT image. This also advocates in favor of a rigid local registration.

3 Combining Rigid and Non-linear Deformations

Based on a segmentation of the objects visible in both images, pairs of homologous points are defined. They constitute landmarks guiding the registration. We assume that globally a non-linear transformation has to be found, while for some objects $O_1,\ldots,O_{n_0}$ (tumors in our application) specific constraints have to be incorporated. For instance, these objects may undergo only a rigid transformation between both images. The global transformation is then interpolated over the whole image. We introduce the rigid structures constraints so that the non-rigid transformation is gradually weighted down in the proximity of objects $O_1,\ldots,O_{n_0}$.

Point-Based Displacement Interpolation

The first step in a point-based interpolation algorithm concerns the selection of the landmarks guiding the transformation. Homologous structures in both images are then registered based on landmarks defined on their surface. The resulting deformation will be exact at these landmarks and smooth elsewhere, which is achieved by interpolation.

Let us denote by $\mathbf{t}_i$ the n landmarks in the source image that we want to transform to new sites $\mathbf{u}_i$ (the homologous landmarks) in the target image.

The deformation at each point $\mathbf{t}$ in the image is defined as:

$$\mathbf{f}(\mathbf{t}) = \mathscr{L}(\mathbf{t}) + \sum_{j=1}^{n} B_j^T \sigma(\mathbf{t},\mathbf{t}_j) \tag{1}$$

under the constraints

$$\forall i, \quad \mathbf{u}_i = \mathbf{f}(\mathbf{t}_i). \tag{2}$$

The first term, $\mathscr{L}(\mathbf{t})$, represents the linear transformation of every point $\mathbf{t}$ in the source image. When n_0 rigid objects ($O_1,O_2,\ldots,O_{n_0}$) are present, the linear term is a weighted sum of each object's linear transformation. The weights $w_i(\mathbf{t})$ are dependent on a measure of distance $d(\mathbf{t},O_i)$ from the point $\mathbf{t}$ to the object O_i as described in [9]:

$$w_i(\mathbf{t}) = \begin{cases} 1 \text{ if } \mathbf{t} \in O_i \\ 0 \text{ if } \mathbf{t} \in O_j,\ j=1,\ldots,n_0,\ j \neq i \\ \frac{q_i(\mathbf{t})}{\sum_{j=1}^{n_0} q_j(\mathbf{t})} \text{ otherwise} \end{cases} \quad \text{where} \quad q_i(\mathbf{t}) = \frac{1}{d(\mathbf{t},O_i)^{\mu}} \tag{3}$$

and $\mu = 1.5$ (for the work illustrated in this paper).

Therefore, for any point $\mathbf{t}$ we define our linear transformation as:

$$\mathscr{L}(\mathbf{t}) = \sum_{i=1}^{n_0} w_i(\mathbf{t}) L_i \tag{4}$$

where $L_i,\ i = 1,\ldots,n_0$ are the linear transformations of the rigid objects. The closer $\mathbf{t}$ is to the object O_i, the more similar its linear transformation will be to L_i.

The second term represents the non-linear transformation which is, for a point $\mathbf{t}$, the sum of n terms, one for each landmark. Each term is the product of the coefficients of a matrix B (that will be computed in order to satisfy the constraints on the landmarks) with a function $\sigma(\mathbf{t},\mathbf{t}_j)$, depending on the (normalized) distance between $\mathbf{t}$ and $\mathbf{t}_j$:

$$\sigma(\mathbf{t},\mathbf{t}_j) = |\mathbf{t}-\mathbf{t}_j|. \tag{5}$$

This form has favorable properties for image registration [17]. However, different functions could be used, as the one described in [9].

With the constraints given by Equation 2, we can calculate the coefficients B of the non-linear term by expressing Equation 1 for $\mathbf{t} = \mathbf{t}_i$. The transformation can then be defined in a matricial way:

$$\Sigma B + L = U \tag{6}$$

where U is the matrix of the landmarks $\mathbf{u}_i$ in the target image (the constraints), $\Sigma_{ij} = \sigma(\mathbf{t}_i,\mathbf{t}_j)$ (given by Equation 5), B is the matrix of the coefficients of the non-linear term and L represents the application of the linear transformations to the landmarks in the source image, $\mathbf{t}_i$.

From Equation 6, the matrix B is obtained as: $B = \Sigma^{-1}(U-L)$. Once the coefficients of B are found, we can calculate the general interpolation solution for every point in $\mathbb{R}^3$ as shown in Equation 1.

Introducing Rigid Structures

In this section, we show how to introduce the constraints imposed by the rigid structures in the images.

To add the influence of the rigid structures $O_1,\ldots,O_{n_0}$, we have redefined the function $\sigma(\mathbf{t},\mathbf{t}_j)$ as $\sigma'(\mathbf{t},\mathbf{t}_j)$ in the following way:

$$\sigma'(\mathbf{t},\mathbf{t}_j) = d(\mathbf{t},O_0)d(\mathbf{t}_j,O_0)\sigma(\mathbf{t},\mathbf{t}_j) \tag{7}$$

where $d(\mathbf{t},O_0)$ is a measure of the distance from point $\mathbf{t}$ to the union of rigid objects $O_0 = O_1 \cup O_2 \cup \ldots \cup O_{n_0}$. It is equal to zero for $\mathbf{t} \in O_0$ (inside any of the rigid structures) and takes small values when $\mathbf{t}$ is near one of the structures. This measure of the distance is continuous over $\mathbb{R}^3$ and it weights the function $\sigma(\mathbf{t},\mathbf{t}_j)$ (see Equation 5). Thus the importance of the non-linear deformation is controlled by the distance to the rigid objects in the following manner:

- $d(\mathbf{t},O_0)$ makes $\sigma'(\mathbf{t},\mathbf{t}_j)$ tend towards zero when the point for which we are calculating the transformation is close to one of the rigid objects;
- $d(\mathbf{t}_j,O_0)$ makes $\sigma'(\mathbf{t},\mathbf{t}_j)$ tend towards zero when the landmark $\mathbf{t}_j$ is near one of the rigid objects. This means that the landmarks close to the rigid structures hardly contribute to the non-linear transformation computation.

Note that this formalism could be more general by replacing $d(\mathbf{t}, O_0)$ by any function of the distance to O_0 that characterizes accurately the behavior of the surrounding regions. Further research is necessary to define such a function in the case of lung tumors. We have used a linear (normalized) distance function as a first approach.

Finally, Equation 6 is rewritten by replacing Σ by Σ', leading to a new matrix B'. We can then calculate the general interpolation solution for every point in $\mathbb{R}^3$ as in Equation 1.

Definition of landmarks and matching

Landmarks can be defined according to the needs of each specific application. They can be uniformly distributed over the surface of homologous objects or based on points having specific properties (maximum of curvature, points undergoing the largest deformations, etc). In our application, we first define a set of landmarks on the surface of the lungs on the CT image, because it has a much better resolution than the PET image. They are approximately uniformly distributed on the surface. Then, we calculate the corresponding points on the surface of the segmented lungs in PET. This is automatically computed by using the Iterative Closest Point (ICP) algorithm [1] and avoids defining by hand the landmarks on both images.

4 Results

We present in this section some results that we have obtained on synthetic, segmented and real images. The structures and the tumors are segmented using the methods in [5] and then, based on pairs of corresponding landmarks in the CT and the PET images, the transformation is computed over the whole image. As mentioned in Section 2, it is reasonable to assume a rigid transformation between the tumors in CT and in PET. As a first approach, we have used a translation. Each translation L_i, $i = 1, \ldots, n_0$ is directly obtained from the segmentation results.

Synthetic images

This first experiment on synthetic images aims at checking that the rigid structures are transformed rigidly, that the landmarks are correctly translated too and, finally, that the transformation elsewhere is consistent and smooth.

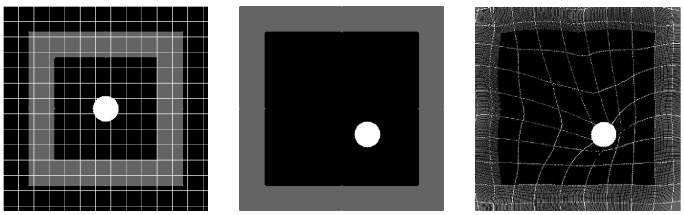


Figure 2: Result on synthetic images: the effect of expanding a frame (in grey in the figure) and translating the "tumor" (in white in the figure). The source image (with a grid) is shown on the left, the target image is in the middle and the result of the transformation on the right. The landmarks are located on the internal and external edges of the frame in grey (on the corners and in the middle of the sides). The total number of landmarks is 16.

As we are taking the PET image as the one to be deformed (source image), we simulate an expansive transformation because the lungs in PET are usually smaller than in

CT images. This is due to the fact that the CT image is often acquired in maximal inspiration of the patient. A simple translation of the "tumor" is simulated too. In order to observe the transformation all over the image, we have plotted a grid on it. It can be seen in Figure 2 that the results with the synthetic images are satisfactory as the shape of the rigid structure (the "tumor") is conserved and the landmarks are translated correctly. The frame, on which the landmarks are placed, is deformed in a continuous and smooth way. If we do not apply the constraints on the rigid structure we obtain an undesired transformation. This is illustrated in [11]. However, it must be noticed that the edges of the frame are not totally straight after the transformation. In general, the more landmarks we have, the better the result will be, and the positions of the landmarks are also very important. Here we have chosen to distribute them uniformly over the internal and external edges of the frame.

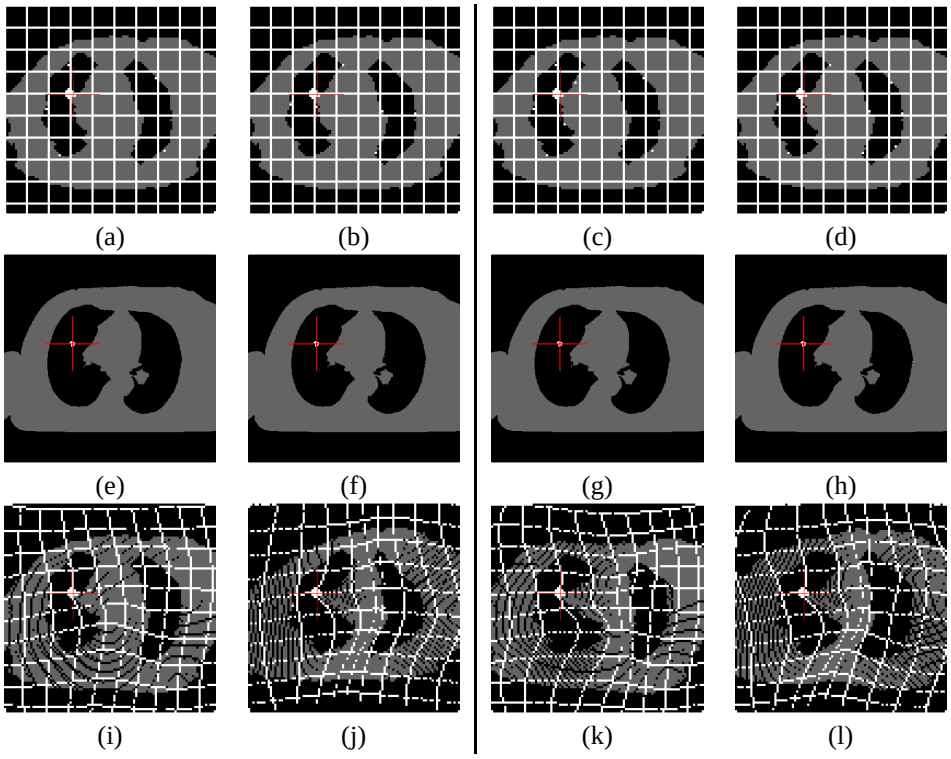


Figure 3: Results on simplified images. Top row: segmented PET images with a grid for visualization purpose (landmarks are also marked in white). Middle row: segmented CT images. Bottom row: results of the registration of the simplified PET and CT images using 4 landmarks (fixed on the corners of the image) and additional landmarks on the walls of the lungs. Left columns: 8 landmarks are chosen on the walls of the lungs using different distributions. Right columns: 12 landmarks are chosen on the walls of the lungs using different distributions. In all the images the cursor is centered on the tumor in the CT image.

Segmented Images

In order to appreciate more clearly the effect of the transformation, we have applied the proposed approach on segmented images. Figure 3 shows some results on the simplified

(segmented) images. A grid is superimposed on the segmented PET image for better visualization. We have fixed the corners of the images to avoid undesired deformations (see illustrations in [11]). It can be observed that for any number of landmarks, the tumor is registered correctly with a rigid transformation. Nevertheless, the quality of the result depends on the quantity of landmarks and their positions. If the number of landmarks is too low or their distribution on the surfaces is not appropriate, the algorithm does not have enough constraints to find the desired transformation. Here the results are obtained by applying the direct transformation in order to better appreciate the influence of the deformation in every region of the image. However it is clear that the final result should be based on the computation of the inverse transformation at each point of the result image in order to avoid unassigned points.

Real Images

Figure 4 shows the results on real images. The tumor is registered correctly with a rigid transformation in all the cases. However, the accuracy of the registration depends on the number and the distribution of the landmarks. If the number of landmarks is not sufficient there are errors. It can be seen that with an appropriate number of landmarks the registration is very satisfactory. The best results (Figure 4(d)) are obtained with 16 landmarks placed as in Figure 3(l). In particular, they include high curvature points. The lower part of the lungs is better registered and the walls of the lungs are perfectly superimposed. The results are considerably improved using 16 landmarks, compared to those obtained with 12 or less landmarks. This shows that the minimal number of landmarks does not need to be very large if the landmarks are correctly distributed.

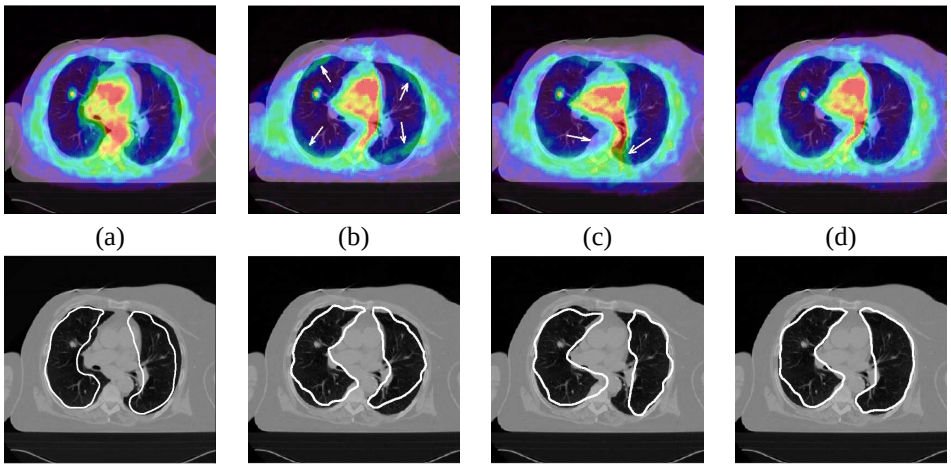


Figure 4: Results on real images. Superimposition of the CT image with: the original PET before registration (a), the deformed PET image using 12 (b) and 16 (c, d) landmarks. (e-h): same results as in (a-d) showing only the contours of the lungs in PET on the CT image. The locations and distribution of the landmarks in (c) are different from the ones in (d) what implies different results. Arrows show misregistrations. This illustrates the importance of the choice of the appropriate landmarks.

5 Conclusion and Future Work

We have developed a non-linear registration method incorporating constraints on deformations of specified objects. It has been shown to be adapted to images which contain rigid structures. The method consists in computing a deformation guided by a group of landmarks and with rigidity constraints. This method has been illustrated on the example of CT/PET registration, in pathological cases where most tissues undergo non-linear transformations due to breathing while tumors remain rigid. In this application, results are very satisfactory and our algorithm avoids undesired tumor misregistrations and preserves tumor geometry and intensity.

One of the originalities of our approach, in particular compared to the method in [9], is that the positions of the landmarks are adapted to the shapes of the structures in the images. In addition to this, with our algorithm, the landmarks are only defined manually in one of the images (the CT) and automatically in the second one (the PET) by means of the ICP algorithm. In the illustrated application, as the transformation near the tumor is reduced by a weight depending on a distance measure, even if the tumor segmentation is not perfect, the registration remains consistent and robust. Moreover, the tumor in CT and PET has not necessarily the same size and shape, therefore the registration of these two modalities is very useful because all the information of the PET image is preserved. This is very important in order to know the true extension of the pathology for diagnosis and for the treatment of the tumor with radiotherapy, for example.

The choice of the landmarks in the CT image is done manually for the moment. However, future work aims at developing an automatic method for defining the landmarks homogeneously distributed all over the surface and on the regions of maximum curvature. A quantitative measure of the alignment between the images will be used in order to find the best distribution of the landmarks that minimizes this similarity measure.

It is also necessary to carry out a detailed study of the rigidity properties of the tissues surrounding a pathology. Replacing the distance by another function would then be straightforward using our formulation.

Although validation is a common difficulty in registration [13], we plan an evaluation phase in collaboration with clinicians, as well as comparison with other methods.

Acknowledgements

The authors would like to thank Liège, Lille, Louisville and Val de Grâce Hospitals for the images and helpful discussions and the members of Segami Corporation for their contribution to this project. This work was partially supported by the French Ministry for Research.

References

[1] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[2] T. Blaffert and R. Wiemker. Comparison of Different Follow-Up Lung Registration Methods with and without Segmentation. In *Proceedings of SPIE Medical Imaging 2004*, volume 5370, pages 1701–1708, San Diego, California, USA, February 2004.

[3] O. Camara-Rey. *Non-Linear Registration of Thoracic and Abdominal CT and PET Images: Methodology Study and Application in Clinical Routine*. PhD thesis, ENST 2003 E 043, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, December 2003.

[4] N. P. Castellanos, P. L. D. Angel, and V. Medina. Nonrigid Medical Image Registration Technique as a Composition of Local Warpings. *Pattern Recognition*, 37:2141–2154, 2004.

[5] G. Delso. *Registro Elástico de Imágenes Médicas Multimodales. Aplicación en Oncología*. PhD thesis, Centre de Recerca en Enginyeria Biomèdica, Universitat Politècnica de Catalunya, Barcelona, Spain, October 2003.

[6] J. Feuardent, M. Soret, O. de Dreuille, H. Foehrenbach, and I. Buvat. Reliability of SUV Estimates in FDG PET as a Function of Acquisition and Processing Protocols. In *Proceedings of the IEEE Nuclear Science Symposium Conference Record*, volume 4, pages 2877–2881, October 2003.

[7] M. Hachama, F. Richard, and A. Desolneux. A Mammogram Registration Technique Dealing With Outliers. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI'06)*, Arlington, Virginia, USA, April 2006.

[8] R. H. Huesman, G. J. Klein, J. A. Kimdon, C. Kuo, and S. Majumdar. Deformable Registration of Multimodal Data Including Rigid Structures. *IEEE Transactions on Nuclear Science*, 50(3):4S–14S, June 2003.

[9] J. A. Little, D. L. G. Hill, and D. J. Hawkes. Deformations Incorporating Rigid Structures. *Computer Vision and Image Understanding*, 66(2):223–232, May 1997.

[10] D. Loeckx, F. Maes, D. Vandermeulen, and P. Suetens. Nonrigid Image Registration Using Free-Form Deformations with Local Rigidity Constraint. In *Proceedings of MICCAI 2004*, volume LNCS 3216, pages 639–646, Rennes - St-Malo, France, September 2004.

[11] A. Moreno, G. Delso, O. Camara, and I. Bloch. CT and PET Registration Using Deformations Incorporating Tumor-Based Constraints. In *Proceedings of the 10th Iberoamerican Congress on Pattern Recognition (XCIARP)*, volume LNCS 3773, pages 1–12, La Habana, Cuba, nov 2005.

[12] T. Rohlfing and C. R. Maurer. Intensity-Based Non-Rigid Registration Using Adaptive Multilevel Free-Form Deformation with an Incompressibility Constraint. In *Proceedings of MICCAI 2001*, volume LNCS 2208, pages 111–119, Utrecht, The Netherlands, October 2001.

[13] J. A. Schnabel, C. Tanner, A. D. Castellano-Smith, A. Degenhard, M. O. Leach, D. R. Hose, D. L. G. Hill, and D. J. Hawkes. Validation of Nonrigid Image Registration Using Finite-Element Methods: Application to Breast MR Images. *IEEE Transactions on Medical Imaging*, 22(2):238–247, February 2003.

[14] R. Shekhar, V. Walimbe, S. Raja, V. Zagrodsky, M. Kanvinde, G. Wu, and B. Bybel. Automated 3-Dimensional Elastic Registration of Whole-Body PET and CT from Separate or Combined Scanners. *The Journal of Nuclear Medicine*, 46(9):1488–1496, September 2005.

[15] C. Tanner, J. A. Schnabel, D. Chung, M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Volume and Shape Preservation of Enhancing Lesions When Applying Non-rigid Registration to a Time Series of Contrast Enhancing MR Breast Images. In *Proceedings of MICCAI 2000*, volume LNCS 1935, pages 327–337, Pittsburgh, USA, 2000.

[16] H. N. Wagner. PET and PET/CT: Progress, Rewards, and Challenges. *The Journal of Nuclear Medicine*, 44(7):10N–14N, July 2003.

[17] R. Wiemker, K. Rohr, L. Binder, R. Sprengel, and H. S. Stiehl. Application of Elastic Registration to Imagery from Airbone Scanners. In *XVIII Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS'96)*, volume XXXI-B4, pages 949–954, Vienna, Austria, 1996.

Evaluation of a Rigidity Penalty Term for Nonrigid Registration

Marius Staring, Stefan Klein and Josien P.W. Pluim

Image Sciences Institute, University Medical Center Utrecht,
P.O. Box 85500, 3508 GA, Room Q0S.459, Utrecht, The Netherlands,
{marius,stefan,josien}@isi.uu.nl

Abstract

Nonrigid registration of medical images usually does not model properties of different tissue types. This results for example in nonrigid deformations of structures that are rigid. In this work we address this problem by employing a local rigidity penalty term. We illustrate this approach on a 2D synthetic image, and evaluate it on clinical 2D DSA image sequences, and on 3D CT follow-up data of the thorax of patients suffering from lung tumours. The results show that the rigidity penalty term does indeed penalise nonrigid deformations of rigid structures, whereas the standard nonrigid registration algorithm compresses those.

1 Introduction

Nonrigid registration of patient data is an important technique in the field of medical imaging. One of the remaining problems of nonrigid registration algorithms is that usually everything in the images is treated as nonrigid tissue, even objects that are clearly rigid, or objects for which it is desired to keep them rigid. Examples of the first are bones and surgical instruments. Examples of the second include structures that contain contrast material, visible in one image, but not in the other. Standard intensity based nonrigid registration algorithms will typically give undesired compression of these structures [10, 15]. It is necessary to prevent nonrigid behaviour of these local structures, and keep them rigid instead.

In the literature several methods have been described to constrain deformations. The employment of a regularisation or penalty term is a well-known strategy. Examples of such terms are the bending energy of a thin plate [12], the linear elasticity constraint [1, 2] and the incompressibility constraint [10]. Particular methods to enforce rigidity on structures have also been proposed. Tanner et al. [15] propose to couple the control points of a B-spline deformation. Another approach is taken by Little et al. [6], who use modified basis functions describing the deformation, constraining the nonlinear part of the deformation at rigid locations by multiplication with a weight function.

Recently, some approaches similar to our own were published [7, 11], in which rigidity is enforced by penalising deviation of the Jacobian from being orthonormal. In this paper we propose a penalty term that is capable of penalising locally nonrigid transformations, which we call a rigidity penalty term. It is based on three criteria a transformation

must meet in order to be locally rigid: linearity of the transformation, and orthonormality and properness of the Jacobian of the transformation. First results of the method were published previously [14].

In the following section we describe the rigidity penalty term. Standard nonrigid registration is compared against registration using the rigidity penalty term in Section 3. We end with a discussion in Section 4.

2 Method

Registration of a moving image $M(\boldsymbol{x}) : \Omega_M \subset \mathbb{R}^d \mapsto \mathbb{R}$ to a fixed image $F(\boldsymbol{x}) : \Omega_F \subset \mathbb{R}^d \mapsto \mathbb{R}$, both of dimension d, consists of finding a deformation $\boldsymbol{u}(\boldsymbol{x})$ that makes $M(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))$ spatially aligned to $F(\boldsymbol{x})$. The quality of alignment is defined by a similarity or distance measure $\mathscr{D}$, such as the sum of squared differences (SSD), the correlation ratio, or the mutual information (MI) measure.

Because this problem is ill-posed, a regularisation term or smoother $\mathscr{S}$ is introduced and the registration problem is formulated as an optimisation problem in which a cost function $\mathscr{J}$ is minimised w.r.t. $\boldsymbol{u}$, with:

$$\mathscr{J}[F,M;\boldsymbol{u}] = \mathscr{D}[F,M;\boldsymbol{u}] + \alpha\mathscr{S}[\boldsymbol{u}], \tag{1}$$

where α weighs similarity against smoothness. Note that at the minimum the derivatives of the similarity measure and the regularisation term are not necessarily zero. Merely, a balance is found between the two, which is influenced by the parameter α. Therefore, the penalty term can not be considered a hard constraint, but it is sometimes referred to as a soft constraint.

In [14] we propose a penalty term $\mathscr{S}^{\text{rigid}}[\mathbf{u}]$ that penalises nonrigid deformations of rigid objects, which we call the rigidity penalty term. This penalty term can be weighted locally, so that some parts of the image are restricted to rigid movement, while other parts are penalised partially or are allowed to deform freely.

2.1 Registration Algorithm

We employ a registration framework largely based on the papers of Rueckert et al. [12] and Mattes et al. [8]. The similarity measure is the mutual information measure using an implementation by Thévenaz and Unser [16]. The deformation field is parameterised by cubic B-splines. A multiresolution approach is taken to avoid local minima, using a Gaussian pyramid with a subsampling factor of two in each dimension. We also employ a multiresolution approach of the deformation grid: when the image resolution in the pyramid is doubled, the B-spline control point spacing is halved. For the optimisation of the cost function $\mathscr{J}$, we employ a stochastic gradient descent optimiser, using only a small, randomly chosen portion of the total number of pixels for calculating the derivative of $\mathscr{J}$ with respect to the B-spline parameters [5].

2.2 Construction of the Rigidity Penalty Term

The rigidity penalty term $\mathscr{S}^{\text{rigid}}[\mathbf{u}]$ is constructed by penalising deviation from three conditions. For a deformation field $\boldsymbol{u}$ to be rigid, it must hold that $\boldsymbol{u}(\boldsymbol{x})+\boldsymbol{x} = R\boldsymbol{x}+\boldsymbol{t}$, with R

and $\boldsymbol{t}$ a rotation matrix and a translation vector, respectively. Note that the rotation matrix R is the Jacobian of the transformation $\boldsymbol{u}(\boldsymbol{x})+\boldsymbol{x}$. The three conditions are (given in 2D for readability):

linearity of $\boldsymbol{u}(\boldsymbol{x})$, stating that the second order derivatives are zero:

$$\mathrm{LC}_{kij}(\boldsymbol{x}) = \frac{\partial^2 u_k(\boldsymbol{x})}{\partial x_i \partial x_j} = 0, \tag{2}$$

for all $k,i,j = 1,2$, not counting duplicates.

orthonormality of R, which can be expressed in terms of derivatives of $\boldsymbol{u}(\boldsymbol{x})$:

$$\mathrm{OC}_{ij}(\boldsymbol{x}) = \sum_{k=1}^{2} \left(\frac{\partial u_k(\boldsymbol{x})}{\partial x_i} + \delta_{ki} \right) \left(\frac{\partial u_k(\boldsymbol{x})}{\partial x_j} + \delta_{kj} \right) - \delta_{ij} = 0, \tag{3}$$

for all $i,j = 1,2$, again not counting duplicates.

properness of R: $\mathrm{PC}(\boldsymbol{x}) = \det(R) - 1 = 0$, which can again be expressed in terms of derivatives of $\boldsymbol{u}(\boldsymbol{x})$. Note that this condition basically amounts to an incompressibility constraint, see also [10].

We define the rigidity penalty term $\mathscr{S}^{\mathrm{rigid}}[\boldsymbol{u}]$ to be the sum of these conditions squared. In order to distinguish between rigid and nonrigid tissue, the penalty term is weighted by a so-called rigidity coefficient $c(\boldsymbol{x}) \in [0,1]$ of the tissue type at position $\boldsymbol{x}$. The complete expression reads:

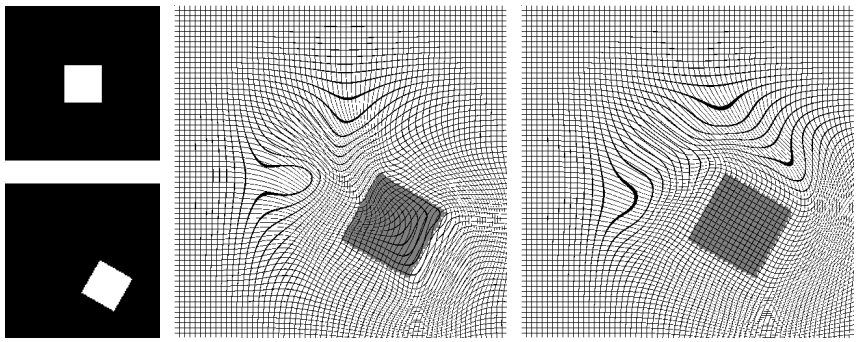
$$\mathscr{S}^{\mathrm{rigid}}[\boldsymbol{u}] \triangleq \sum_{\boldsymbol{x}\in\Omega_F} c(\boldsymbol{x}) \left\{ \sum_{k,i,j} \left[\mathrm{LC}_{kij}(\boldsymbol{x})\right]^2 + \sum_{i,j} \left[\mathrm{OC}_{ij}(\boldsymbol{x})\right]^2 + \left[\mathrm{PC}(\boldsymbol{x})\right]^2 \right\}. \tag{4}$$

The rigidity coefficient $c(\boldsymbol{x})$ is equal to zero for pixels $\boldsymbol{x}$ in completely nonrigid tissue, thereby not penalising deformations at those locations. For completely rigid tissue $c(\boldsymbol{x})$ is set to one. For other tissue types a value of $c(\boldsymbol{x})$ is chosen between zero and one. The rigidity coefficient image can be constructed by performing a manual or automatic segmentation of structures of interest, after which rigidity coefficients can be assigned. Depending on the application, different methods to create the rigidity coefficient image can be considered. For the case of CT images the Hounsfield unit might be used, rescaled to the range $[0,1]$, since more rigid tissue usually has a higher attenuation value.

In [14] we argue the validity of this rigidity penalty term by showing that $\mathscr{S}^{\mathrm{rigid}}[\boldsymbol{u}] = 0$ if and only if the deformation field $\boldsymbol{u}(\boldsymbol{x})$ is locally rigid. The linearity term might be dropped, since it can be shown that orthonormality implies linearity. However, it might aid the penalty term by guiding the optimisation path. The proposed constraint is not dependent on the B-spline parameterisation of the deformation field. However, from a computational point of view, we can benefit from this parameterisation by evaluating the rigidity penalty term only over the control points. For details we refer to [14].

3 Experiments and Results

Standard nonrigid registration using only the similarity term, as described in Section 2.1, is compared with nonrigid registration using the rigidity penalty term. The two methods



(a) (b) (c)

Figure 1: Comparison of registration with and without $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$ for a 2D synthetic example. The white square is to be kept rigid. (a) fixed image (lower) and moving image (upper), (b) resulting deformation field of standard registration, and (c) resulting deformation field including the rigidity constraint.

are illustrated on synthetic images. They are compared on clinical data, viz. 3D CT follow-up data of the thorax containing lung tumours, and on 2D DSA image data of different parts of the body. The computation time for registration with the rigidity penalty term increases about 50% in 2D and about 80% in 3D with a B-spline grid spacing of eight voxels.

All experiments were performed with software (`www.isi.uu.nl/Elastix/`) developed by the authors. This registration package is largely based on the Insight Segmentation and Registration Toolkit [4].

3.1 2D Synthetic Example

Rotation of a rigid object is illustrated with the square in Figure 1, where the background represents nonrigid tissue and the square a rigid object. The rigidity coefficient $c(\boldsymbol{x})$ is set to 1.0 on the square and 0.0 elsewhere.

Both algorithms give near perfect registration results for the matching of the squares in Figure 1. However, the underlying deformation field is highly nonlinear if no rigidity penalty term is used. By including the penalty term the deformation field is almost perfectly rigid at the rigid part. This is also reflected in the rigidity constraint $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$, which has a value of 2.21×10^2 for standard nonrigid registration and a value of 1.28×10^{-3} for registration using the rigidity constraint. The rigidity constraint is not perfectly zero, because some B-spline control points outside the square, but influencing the points within the square, are not set to be rigid.

3.2 3D CT Follow-up

In order to compare 3D CT follow-up data by visual inspection of the difference image, the data sets must be registered nonrigidly. We have five CT follow-up data sets of the thorax available, of patients suffering from lung tumours, collected at the Radiology department of the University Medical Centre Utrecht. The images are acquired under

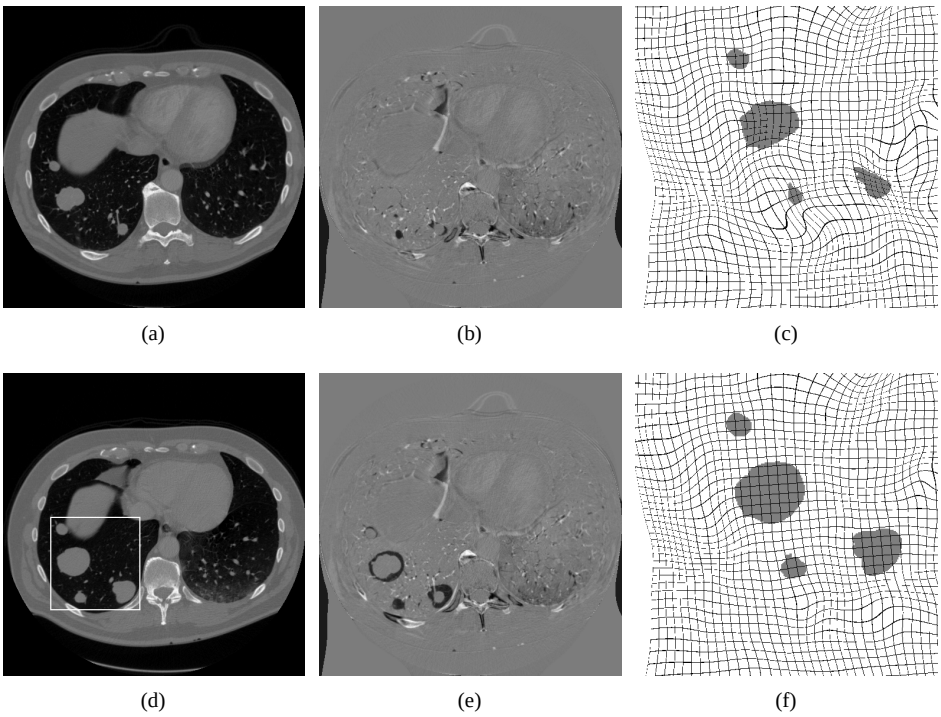


Figure 2: Comparison of both nonrigid registration methods for a slice taken from 3D CT thorax images. The tumours, located within the box (see (d)), have to be kept rigid for diagnostic reasons. (a) and (d): CT slice at time t_0 (the fixed image) and time t_1 (the moving image), respectively, (b) and (c): difference of the result of similarity only nonrigid registration with the fixed image, and a part of the resulting deformation field, respectively, (e) and (f): the same for nonrigid registration using $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$.

breath-hold with a Philips 16 slice spiral CT scanner (Mx8000 IDT 16). The data are of size 512×512 by 400 - 550 slices, have voxel sizes of around $0.7 \times 0.7 \times 0.7$ mm, and are resized by a factor of 2 in each dimension before registration. The five data sets contain 36 tumours in total, with an average volume of 2.5 ml for the first scan and 5.1 ml at the second. The CT image taken at time t_0 is set to be the fixed image, the follow-up CT image (time t_1) the moving image. For visual inspection of the tumour growth from the difference image, the tumours must not deform nonrigidly, but only in a rigid fashion. This way tumour growth can be seen from the difference image, in relation with the anatomy shown in the CT image. For an example of the difference image in these cases, see Figure 2(b) and 2(e). Tumour growth is effectively concealed in the similarity only based registration.

To get a coarse alignment between fixed and moving image a rigid registration is performed first. Then the nonrigid registration is performed, using a B-spline grid spacing of 8 voxels, 4 resolutions, 300 iterations per resolution, and 5000 voxel samples to calculate (the derivative of) the mutual information. For the nonrigid registration with the rigidity penalty term, the tumour regions are defined by a crude manual delineation, setting

Table 1: Average lung overlap.

before registration	rigid	similarity only	with $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$
0.64 ± 0.22	0.91 ± 0.06	0.98 ± 0.01	0.97 ± 0.02

$c(\boldsymbol{x}) = 1.0$ in the tumour regions and 0.0 elsewhere.

Accuracy of the registration is measured by calculating the lung overlap of the registered image with the fixed one. For this purpose automatic lung segmentations are made with an algorithm based on the method by Hu et al. [3,13]. The overlap measure is defined as

$$\text{overlap} \triangleq \frac{2 \cdot |L_1 \cap L_2|}{|L_1| + |L_2|}, \tag{5}$$

where L_i is the set of all voxels from the lung, and where $|L_i|$ is the size of set L_i. From the results reported in Table 1, we see that both registrations lead to good lung overlap. Employing a rigidity penalty term, constrains the deformation, leading to a slightly less accurate lung overlap. Because of the (compact) support of the B-splines, a boundary around the tumour is influenced by the control points within the tumour. The extent of this boundary can be controlled with the B-spline grid spacing.

Manual segmentations of the tumours are used to evaluate their rigidity. Tumour volume measurements are performed to see if the registration is at least volume preserving. In order to compare tumours volumes with different sizes we can not use the arithmetic mean, because large tumours influence the arithmetic mean disproportionally. Therefore, volume growth ratios are calculated, where every tumour volume is divided by its volume at t_1. For growth ratios it is better to use the *geometric mean* and the *geometric standard deviation*, which are defined as:

$$\mu_g = \sqrt[n]{\prod_{i=1}^{n} r_i}, \qquad \sigma_g = \exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\ln r_i - \ln \mu_g)^2}\right), \tag{6}$$

where r_i denotes the growth ratio of tumour i. We report the geometric mean growth ratios and standard deviations in Table 2. It can be appreciated that volume is much better preserved when applying the rigidity penalty term, compared to similarity only based registration. Part of the residual volume difference can be explained by interpolation artifacts due to resampling, as can be seen from the results for rigid registration. From the deformation field, see Figure 2(f) (compare with 2(c)) is can also be appreciated that nonrigid registration using the rigidity penalty term preserves rigidity locally.

3.3 Digital Subtraction Angiography

Evaluation is also performed on 2D clinical digital X-ray angiography image data, acquired with an Integris V3000 C-arm imaging system (Philips). Digital Subtraction Angiography (DSA) imaging often suffers from motion artifacts, due to motion of or within the patient, see Figure 3(a) - 3(c). Nonrigid registration is needed to compensate for this. We have 26 image sequences available of twelve different patients, each containing about 10 images. Images are mostly of size 512×512 and are taken of different locations in the

Table 2: Geometric mean tumour volume ratios. Geometric means are calculated for four growth groups and for all growth ratios. The second group for example is the group of tumours that corresponds to a volume ratio t_1/t_0 between 1 and 3/2.

group t_1/t_0	t_0	rigid	similarity only	with $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$
$(0,1]$	1.18 ×/ 1.08	0.99 ×/ 1.02	1.00 ×/ 1.03	0.96 ×/ 1.06
$\left(1,\frac{3}{2}\right]$	0.84 ×/ 1.05	0.96 ×/ 1.04	0.97 ×/ 1.05	1.03 ×/ 1.03
$\left(\frac{3}{2},3\right]$	0.48 ×/ 1.10	1.00 ×/ 1.02	0.79 ×/ 1.07	1.00 ×/ 1.02
$(3,\infty)$	0.23 ×/ 1.13	1.02 ×/ 1.03	0.69 ×/ 1.12	0.99 ×/ 1.02
all	0.52 ×/ 1.29	0.99 ×/ 1.03	0.83 ×/ 1.10	1.00 ×/ 1.03

body, such as the abdomen, the brain, neck and lungs. Because it takes time for the contrast bolus to travel through the vasculature, different parts of the vasculature are visible at different times. If the whole vasculature is to be extracted, all images have to be registered to some baseline image. For this, we choose the first image in each sequence, which is taken before arrival of the contrast bolus, to be the fixed image. For this experiment we register only one of the other images to the fixed image: the image showing most of the vasculature is chosen as the moving image. The nonrigid registration of DSA images can lead to undesired compression of the vasculature, as reported in [9] for CT-DSA. Although the vessels are not intrinsically rigid, they are to be kept rigid, since there exists no information to do otherwise: vessels are visible in one image and not in the other.

We applied a rigid registration first for coarse alignment. For the nonrigid registration we use a B-spline grid spacing of 16 pixels, 2 resolutions, 600 and 300 iterations per resolution, and 5000 samples for calculating (the derivative of) the MI. A crude manual segmentation of the vessels is used for defining $c(\boldsymbol{x})$.

To measure the success of the nonrigid registration, the Mean Square Difference (MSD) of the background $\mathscr{B}$ is calculated. The MSD is defined as:

$$\text{MSD} = \frac{1}{|\mathscr{B}|} \sum_{\boldsymbol{x}\in\mathscr{B}} \Big(F(\boldsymbol{x}) - M(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) \Big)^2 . \tag{7}$$

Results are shown in the top row of Table 3, showing improvement of this measure for nonrigid registration. Compare also Figure 3(d) with Figure 3(e) and 3(e). As a measure of rigidity, vessel diameter measurements are carried out to quantify vessel compression. In every one of the 26 images six diameter measurements are carried out. The geometric mean of the vessel diameter ratios are reported in the bottom row of Table 3. Without employing the rigidity penalty term the vessels are severely compressed, which is avoided with the use of $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$. The shape of the deformation fields confirms this, see Figure 3(g) and 3(g).

4 Conclusions and Discussion

We have proposed a method to perform nonrigid registration, while keeping user-defined local structures rigid. This is achieved by adding a rigidity penalty term to the registration

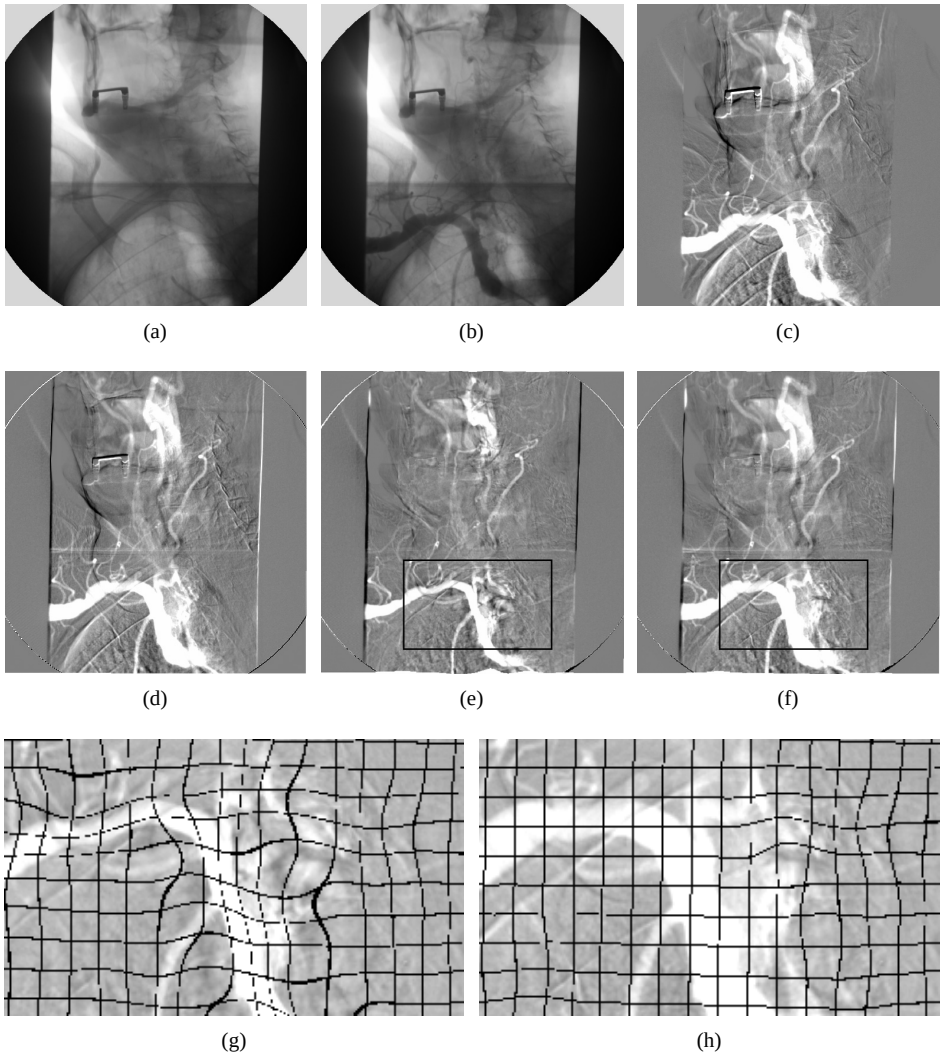


Figure 3: Comparison of different registration algorithms for 2D DSA images. The vessels are to be kept rigid, while motion artifacts are to be reduced by the registration. (a) DSA baseline image: the fixed image, (b) DSA image after injection of the contrast bolus: the moving image, (c) - (f) are difference images with the fixed image, (c) with the moving image, (d) with the result of rigid registration, (e) with the result of similarity only nonrigid registration, (f) with the result of nonrigid registration using the rigidity penalty term. The bottom row depicts parts of the resulting deformation field of similarity only (g), and using the rigidity penalty term (h). The black box in Figure (e) and (f) denotes the part of the deformation field that is depicted.

framework (1). The method is illustrated on a synthetic 2D example and evaluated on 2D and 3D clinical data.

Table 3: Results for DSA. Arithmetic means for the MSD and geometric means for the vessel diameter ratios compared with t_1.

	no registration	rigid	similarity only	with $\mathscr{S}^{\text{rigid}}[\boldsymbol{u}]$
MSD	231 ± 205	215 ± 181	163 ± 105	171 ± 128
diameter	$1.00 \times/ 1.00$	$1.00 \times/ 1.00$	$0.85 \times/ 1.07$	$0.99 \times/ 1.01$

From the results we can see that the rigidity penalty term indeed penalises nonrigid deformations, but complete rigidity is sometimes not achieved. Some reasons are the following. 1) The rigidity penalty term is not a hard constraint, it is merely a tradeoff between similarity and penalty. 2) For the case of modelling the deformation field with cubic B-splines, the control points outside a rigid structure influence the inside. Therefore, if complete rigidity is wanted, the two adjacent control points outside the structure must also be kept rigid, for example by performing a dilation on the rigidity coefficient image $c(\boldsymbol{x})$. Of course the inverse is also true: the rigid part influencing the nonrigid part, thereby restricting the deformation at the boundaries of the rigid structure. The extent of this boundary is controlled by the B-spline control point spacing.

The linearity condition (2) might not be necessary, since it can be shown that orthonormality of the Jacobian of $\boldsymbol{u}$ implies linearity of $\boldsymbol{u}$. However, it might guide the optimiser in reaching rigidity; this is interesting future work.

Acknowledgments

This research was funded by the Netherlands Organisation for Scientific Research (NWO). This work also benefited from the use of the Insight Segmentation and Registration Toolkit (ITK). We gratefully acknowledge Oskar Škrinjar for a fruitful discussion about the linearity condition of the rigidity penalty term.

References

[1] G. E. Christensen and H. J. Johnson. Consistent image registration. *IEEE Transactions on Medical Imaging*, 20(7):568 – 582, 2001.

[2] B. Fischer and J. Modersitzki. A unified approach to fast image registration and a new curvature based registration technique. *Linear Algebra and its Applications*, 380:107 – 124, 2004.

[3] S. Hu, E. A. Hoffman, and J. M. Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric X-Ray CT images. *IEEE Transactions on Medical Imaging*, 20(6):490 – 498, 2001.

[4] Luis Ibáñez, Will Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, second edition, 2005.

[5] S. Klein, M. Staring, and J. P. W. Pluim. Comparison of gradient approximation techniques for optimisation of mutual information in nonrigid registration. In *SPIE*

Medical Imaging: Image Processing, volume 5747, pages 192 – 203. SPIE Press, 2005.

[6] J. A. Little, D. L. G. Hill, and D. J. Hawkes. Deformations incorporating rigid structures. *Computer Vision and Image Understanding*, 66(2):223 – 232, 1997.

[7] D. Loeckx, F. Maes, D. Vandermeulen, and P. Suetens. Nonrigid image registration using free-form deformations with a local rigidity constraint. In *MICCAI*, volume 3216 of *Lecture Notes in Computer Science*, pages 639 – 646. Springer Verlag, 2004.

[8] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank. PET-CT image registration in the chest using free-form deformations. *IEEE Transactions on Medical Imaging*, 22(1):120 – 128, 2003.

[9] T. Rohlfing and C. R. Maurer Jr. Intensity-based nonrigid registration using adaptive multilevel free-form deformation with an incompressibility constraint. In *MICCAI*, volume 2208 of *Lecture Notes in Computer Science*, pages 111 – 119. Springer Verlag, 2001.

[10] T. Rohlfing, C. R. Maurer Jr., D. A. Bluemke, and M. A. Jacobs. Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint. *IEEE Transactions on Medical Imaging*, 22(6):730 – 741, 2003.

[11] D. Ruan, J. A. Fessler, M. Roberson, J. Balter, and M. Kesler. Nonrigid registration using regularization that accomodates local tissue rigidity. In *SPIE Medical Imaging: Image Processing*, volume 6144, pages 346 – 354. SPIE Press, 2006.

[12] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712 – 721, 1999.

[13] I. C. Sluimer, M. Prokop, and B. van Ginneken. Towards automated segmentation of the pathological lung in CT. *IEEE Transactions on Medical Imaging*, 24(8):1025 – 1038, 2005.

[14] M. Staring, S. Klein, and Josien P. W. Pluim. Nonrigid registration using a rigidity constraint. In *SPIE Medical Imaging: Image Processing*, volume 6144 of *Proceedings of SPIE*, pages 355 – 364. SPIE Press, 2006.

[15] C. Tanner, J. A. Schnabel, D. Chung, M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Volume and shape preservation of enhancing lesions when applying nonrigid registration to a time series of contrast enhancing MR breast images. In *MICCAI*, volume 1935 of *Lecture Notes in Computer Science*, pages 327 – 337. Springer Verlag, 2000.

[16] P. Thévenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image Processing*, 9(12):2083 – 2099, 2000.

The Constrained Distance Transform: Interactive Atlas Registration With Large Deformations Through Constrained Distances

Bill Hill and Richard A. Baldock
MRC Human Genetics Unit
Edinburgh, EH4 2XU, UK
Bill.Hill@hgu.mrc.ac.uk
Richard.Baldock@hgu.mrc.ac.uk

Abstract

We describe a modification to distance based transforms used for non–linear registration, such as radial basis function transforms, which is both simple and novel. In the Constrained Distance Transform (CDT), distances are constrained to geodesics within the domain of the object being transformed. We show that the modified method is capable of producing large meaningful deformations for both an extreme synthetic case and for real biological image data. We further show that the method is suitable for interactive use on current workstations.

1 Introduction

Spatially mapped databases are becoming increasingly common in biomedical research. Some of these databases, such as EMAGE [1], contain volumetric models onto which spatially organised data are mapped through non–linear spatial transformations.

Two connected problems that are frequently encountered when defining these transformations are: Large deformations, such as those required for the pose of limbs or a curvature in the main body of an object, frequently fail to produce meaningful results; secondly, very different deformations are frequently required for two regions which are close in Euclidean space but distant in terms of the object. This last situation is illustrated in figure 1, in which despite the two points H and T being close in Euclidean space they are distant in terms of the object being transformed. It is clear that Euclidean geometry does not provide a good description of organisms with complex shapes and that a natural coordinate system might be more appropriate. A natural coordinate system can be defined as a "coordinate system which requires the least residual deformation to explain variability across individuals" [8]. Defining transformations which will uncurl the object or deform the head without deforming the tail are not possible using current methods, but it is just such transformations that are required to establish mappings between the assay and atlas model images shown later in figure 4.

A feature of the images shown in figure 4 is the variability of the assay images, which frequently do not correspond in either image values or their gradients. It is clear that, given the state of existing automatic registration algorithms, some interaction or expert guidance

is required to register the assay images to the atlas if the registration is to generate high quality mappings.

Existing methods which allow large deformations include articulated and fluid models. Martín-Fernández et al. [4] have developed articulated models for registering hand radiographs. In these models a skeleton composed of articulated rods is registered using landmarks at the ends of the rods and displacements away from the rods are interpolated using weighted combination of affine transforms. These models allow large deformations but do not appear to be a natural model for objects that do not have rigid components and in such cases a large number of articulated components might be required to achieve a reasonable transformation. Fluid models based on solving viscoelastic systems have been developed for registration with large deformations. While these models are often successful in producing large deformations, they are computationally expensive, non–interactive and do not always give biologically meaningful results: "Hands and brains do not in general deform like honey"[5].

Radial basis function (RBF) transforms are well suited to interactive use and are frequently used successfully for defining transforms with small deformations. But unfortunately these transforms do not behave well for large deformations or for large deformation gradients and even for smooth transformations they may not give diffeomorphic mappings [5].

In this paper we propose a simple modification to RBF transforms which allows them to be used interactively to produce high quality mappings for large deformations on common desktop workstations.

Figure 1: From left to right: Volume rendering showing points in the head (H) and tail (T), diagram showing the path for the constrained distance between H and T, distance transform showing the distance of all points in the domain from H. In the distance transform image the distance values have been normalised to the range 0 (black) to 255 (white) for clarity.

2 Radial Basis Function Transforms

Because of the difficulty in computing warp transformations automatically, RBF transforms based on manually or semi–manually defined landmark points are frequently used in biomedical applications. These manually defined landmarks can be combined with automatically generated landmarks through various weighting schemes. It is also possible

to combine landmark based registration with intensity based registration [3]. Landmark points give the set of displacements between the defined coordinates of the point pairs, but to achieve a warp transformation it is necessary to have a displacement function that is defined for all points within the domain of the source object. RBF transforms, in which the displacement at any point is given by the sum of weighted functions of the radial distances from the landmark points, are frequently used in this role.

Given a source object $O_s(x,y)$ and a target object $O_t(u,v)$, the displacements Δu and Δv defined by

$$\Delta u = u - x \tag{1}$$

$$\Delta v = v - y \tag{2}$$

can be computed using a RBF transform with the general form

$$\Delta u = P_u(x,y) + \sum_{i=1}^{i=N} \lambda_i b(r_i) \tag{3}$$

$$\Delta v = P_v(x,y) + \sum_{i=1}^{i=N} \mu_i b(r_i) \tag{4}$$

where P_u and P_v are two first order polynomials (the affine component), N is the number of landmark points, λ_i and μ_i are the basis function coefficients and b is the chosen basis function. In these equations r_i is the distance of some point $p(x,y)$ from landmark point l_i.

Two frequently used basis functions are the thin plate spline (TPS) and the multi-quadric (MQ), which in 2D have the form

$$b_{TPS}(r) = r^2 ln r^2 \tag{5}$$

$$b_{MQ}(r) = \sqrt{r^2 + \delta^2}. \tag{6}$$

The value of the MQ regularisation parameter δ is chosen so as to balance the smooth variation in displacements against exact landmark point displacements. This parameter is application dependent, but in practise it can be computed using a simple algorithm [2].

From a set of landmark points a system of linear equations may be written using the coefficients of the polynomial (P_u or P_v) and the RBF (λ_i or μ_i). These equations, known as the design equations, may then be solved using a linear system solver such as singular value decomposition. In practise it is often beneficial to rescale the parameters to reduce the condition number of the design matrix [2].

3 Distance Transforms

A distance transform maps a set of points within some domain of interest to their distance from a reference domain. Most distance transform algorithms assume a single rectangular (or at least convex) domain of interest, but in this work no such restriction is imposed and both the domains may be non–convex. In this more general case, the distances are evaluated along geodesics that are restricted to the domain of interest.

The distance transform algorithm used is based on a region growing algorithm described by Piper and Granum [7], but which has been implemented using efficient morphological primitives based on interval coding [6]. Given two domains, Ω_r the reference

domain and Ω_f the domain specifying the region of interest, a domain with a thin shell Ω_i is iteratively expanded from it's initial domain corresponding to the reference domain Ω_r. At each iteration Ω_i is dilated and clipped by it's intersection with Ω_f until Ω_i becomes the null domain $\emptyset$. At each iteration the current distance is recorded in a value table which covers the domain Ω_f. This is shown in algorithm 1. To approximate Euclidean distances, the domains are rescaled using a scale parameter and dilation is performed using a circular structuring element with a radius equal to the scale parameter. This rescaling is relatively efficient because the morphological primitives are based on interval processing and the number of intervals increases with a domain's linear dimension not it's area.

Algorithm 1 Distance Transform.

Require: Ω_f, Ω_r
1: $c \leftarrow$ initial connectivity
2: $s \leftarrow 0$
3: $V_d \leftarrow$ new integer value table with domain Ω_f
4: $\Omega_i \leftarrow \Omega_r$
5: **while** $\Omega_i \neq \emptyset$ **do**
6: $\quad s \leftarrow s+1$
7: $\quad \Omega_p \leftarrow \Omega_i$
8: $\quad \Omega_i \leftarrow \mathit{diffdom}(\Omega_p, \mathit{dilation}(\Omega_p, c))$
9: $\quad \Omega_d \leftarrow \Omega_i \cap \Omega_f$
10: $\quad V_d|\Omega_d \leftarrow s$
11: **end while**

4 Constrained Distance Transforms

RBF transformations are relatively easy to implement and are well suited for interactive use. But in practise there are many situations in which they fail to produce useful transformations, such as where large deformation gradients are required.

In this paper we propose that a solution to the problem of interactively registering images with large deformations can be found through reconsidering the distances used by the RBFs (and other distance based transforms) and constraining the paths along which the distances are evaluated to be geodesics that lie within the domain of the source object. With regard to figure 1, it is clear that the head and tail are close in Euclidean space and that any landmarks which create a large displacement for the head will result in a similar displacement for the tail. However if the path along which this distance is measured is constrained to a geodesic within the object's domain, then the head is far from the tail despite it's being close in Euclidean space. By using distances constrained to the source object's domain, large displacements applied to the head may be effectively independent of those applied to the tail. All that is required is to substitute constrained distances in place of all Euclidean distances in the RBFs. We call this transformation the Constrained Distance Transform (CDT).

When evaluating the displacement of the source domain at some point p_i, using RBFs defined using landmarks $\{l_j\}$, then all distances between p_i and l_j must be computed for all i and j. Because constrained distances are considerably more expensive to compute

than Euclidean distances efficiency gains can be made by caching the the distance transforms, with one distance transform cached per landmark pair. Using this approach the cached distance transforms may be created and destroyed incrementally in step with the interactively defined landmark points.

In this work the transformation of both object domains and values are accomplished using non–regular triangulated conforming meshes. The use of conforming meshes simplifies image re–sampling in situations, such as limb articulation, which would otherwise give rise to large mesh element deformations requiring complex re–meshing schemes. Objects are transformed by first defining a mesh covering the source domain. This mesh then has a displacement, computed by evaluation of the radial basis function transform, associated with each of it's nodes. The mesh is first used to forward transform the source object's domain and then, if the source object has values associated with it, a new value table is created for the transformed domain and the new values are set using a sweep–line algorithm which scans through the transformed domain setting it's values from the source object via the current mesh element's inverse transform.

5 Results

An initial 2D implementation of the CDT has been evaluated using both synthetic and real biological images. In all cases a multiquadric RBF was used and the approximate Euclidean distance transform scale factor was set to 10.0.

A pair of synthetic images was generated corresponding to a large deformation problem which is known to be problematic for unconstrained RBFs [5]. These images are shown in figure 2. The 'C' image has a domain that is an incomplete annulus, with radii of 100 and 200 pixels and a gap sector of 30°. The 'I' image is an axis aligned rectangle with dimensions 100×800 pixels. Eighteen landmark pairs were computed for the C and I images during their generation and these were used to transform both the C to the I and the I to the C images using a CDT. The resulting images clearly show that CDTs are capable of registering objects requiring large deformations. that are problematic for unconstrained RBFs.

An assay image was selected from the EMAGE database in which the head and tail are close in Euclidean space. The image was segmented from its background using interactive grey value thresholding combined with connected component labelling. Landmark points were then evenly distributed around the segmented image, with all the landmarks having zero displacements except for those in the tail which had a displacement set away from the head. Figure 3 shows this image together with the landmark points and the displacements set. This figure also shows the results of applying a CDT and an unconstrained RBF transform to the image. When a CDT was applied, the head was not subjected to any significant deformation despite the large deformation experienced by the tail. However the unconstrained RBF transform resulted in large deformations to both the head and the tail. In both cases the same landmarks and displacements were used.

Figure 4 shows a set of eight typical 2D assay images from a set of submissions to the EMAGE database. This figure also shows a 2D projection of the corresponding 3D atlas model. These images display the typical variation that is seen in both the pose of the organisms and in their image values. Unconstrained RBF transforms are unable to map the assay images onto the atlas image satisfactorily. Attempts to do so result in folds

of the source image and a confused image in which multiple foreground regions may be blended with background. All assay images were first segmented from their background using interactive grey value thresholding combined with connected component labelling and any extraneous parts (such as the allantois) were removed. The segmented images were then mapped to the 2D atlas image interactively using CDTs, with the landmarks being placed at plausible correspondences. Figure 5 shows the successfully mapped assay images together with the target atlas image. For these images the number of landmark pairs used varied between 31 and 45.

In it's current implementation, the CDT computation time is dominated by the time taken to compute the distance transforms, with this accounting for approximately 99% of the total computation time. To compute the approximate distance transform, in which all distances from a single landmark are computed within the domain of the C image (outer diameter 400 pixels), takes 2.1s in the worst case on a 2.2GHz Opteron CPU using the approximate Euclidean distance transform with a scale factor of 10.0. This is the time delay between a user entering a landmark pair and seeing the displaced image computer.

6 Conclusions

RBF transforms defined using landmarks points are easily implemented and are frequently used for interactive non–rigid registration, however it is well known that these functions often fail to produce satisfactory warps for large deformations. A novel modification of the method has been presented which makes use of constrained distances. The CDT has been shown to produce meaningful results for a synthetic dataset that is known to be problematic for the unmodified method. It has also been shown to be capable of decoupling the displacements of regions that are close in Euclidean space, provided that an adequate segmentation can be achieved. Further, the CDT has been shown to produce satisfactory mappings, requiring large deformations, between biological assay images and a standard atlas model.

The computational cost of the approximate distance transforms currently restricts the size of image that can be transformed interactively. With one distance transform being computed for each landmark pair, the source is limited to 2D objects of a few hundred pixels in each dimension, such as the C image. Although it is possible to use a lower scaling factor in the approximate Euclidean distance transform, this results in a lower quality mapping which may be unacceptable. One method of making the method interactive, even for 3D data, is to map the atlas to the assay image, using landmarks fixed in the atlas image and pre–computed distance transforms. The resulting transform can then easily be inverted with mesh based implementations.

Future developments will include the addition of compact RBFs and the extension of the work to 3D transformations.

References

[1] R. Baldock, J. Bard, A. Burger, N. Burton, J. Christiansen, G. Feng, B. Hill, D. Houghton, M. Kaufman, J. RAo, J. Sharpe, A. Ross, P. Stevenson, S. Venkataraman, A. Waterhouse, Y. Yang, and D.R. Davidson. EMAP and EMAGE: A framework for understanding spatially organised data. *Neuroinformatics*, 1:309–325, 2003.

[2] R.A. Baldock and B. Hill. *Image Warping and Spatial Data mapping*, chapter 9. Oxford Uiversity Press, Oxford, UK, 2000. Editors: Baldock, R. and Graham, J.

[3] B. Fischer and J. Modersitzki. Combination of automatic non-rigid and landmark based registration: the best of both worlds. In M. Sonka and J. M. Fitzpatrick, editors, *Medical Imaging 2003: Image Processing*, pages 1037–1048. SPIE Press, 2003. Proceedings of the SPIE 5032.

[4] M. Á. Martín-Fernández, E. Muñoz Moreno, M. Martín-Fernández, and C. Alberola-López. Articulated registration: Elastic registration based on a wire-model. In J. M. Fitzpatrick and J. M. Reinhardt, editors, *Medical Imaging 2005: Image Processing*, pages 182–191, San Diego, CA, USA, February 12–17 2005. SPIE Press. Proceedings of the SPIE 5747.

[5] Jan Modersitzki. *Numerical Methods for Image registration*. Oxford University Press, Oxford, UK, 2004.

[6] J. Piper and D. Rutovitz. Data structures for image processing in a C language and Unix environment. *Pattern Recognition Letters*, 3:119–129, 1985.

[7] Jim Piper and Erik Granum. Computing distance transformations in convex and non-convex domains. *Pattern Recognition*, 20(6):599–615, 1987.

[8] Daniel Rueckert, Alejandro F. Frangi, and Julia A. Schnabel. Automatic construction of 3D statistical deformation models of the brain using non-rigid registration. *IEEE Transactions on Medical Imaging*, 22(8):1014–1025, 2003.

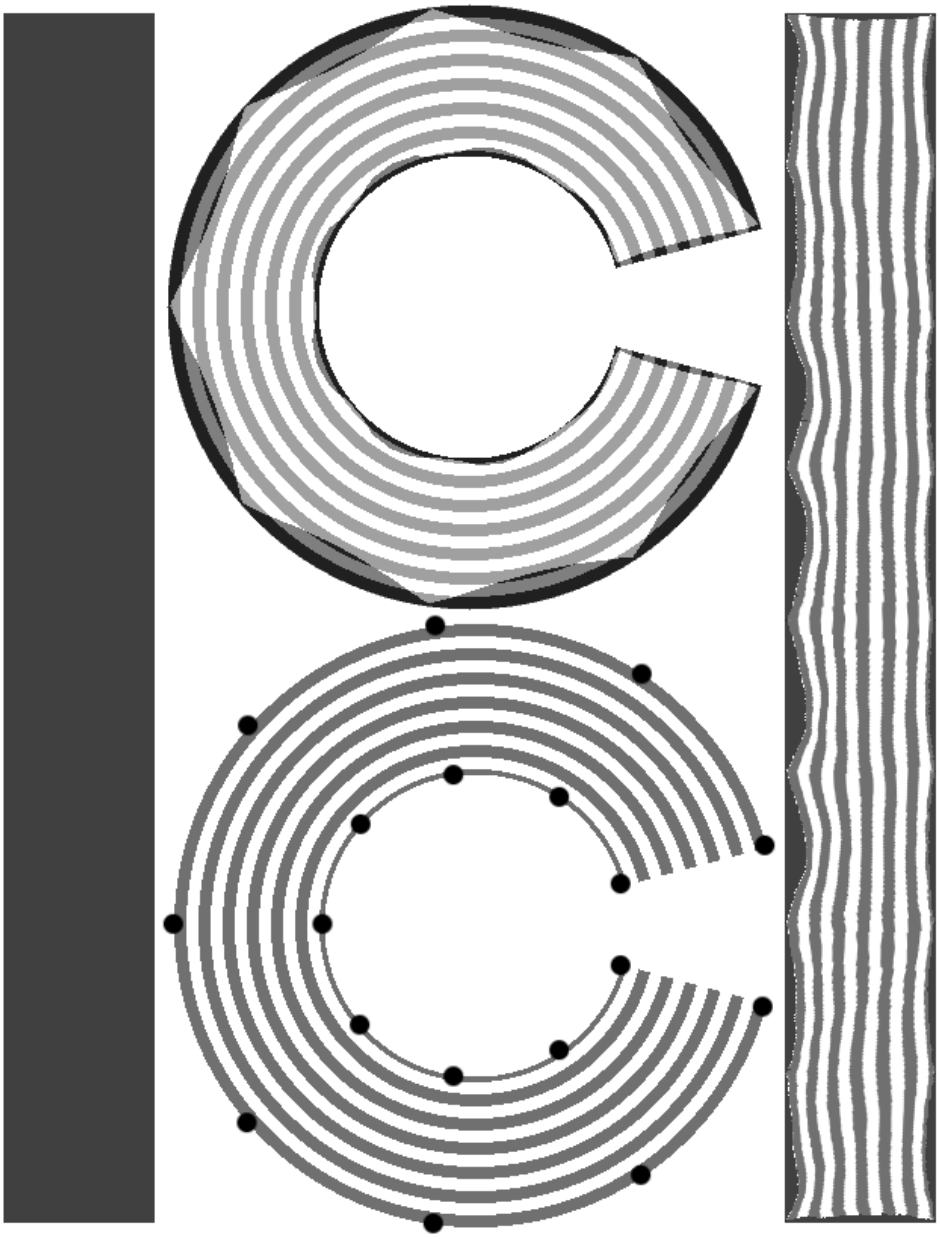


Figure 2: C and I dataset. Counter clockwise from the left: I object, C object showing the positions of the landmark points, overlay of the C object warped to the I, overlay of the I object warped to the C.

Figure 3: From left to right: Source image (from EMAGE) showing fixed points and displacements, source image transformed using a CDT and source image transformed using an unconstrained RBF transformation.

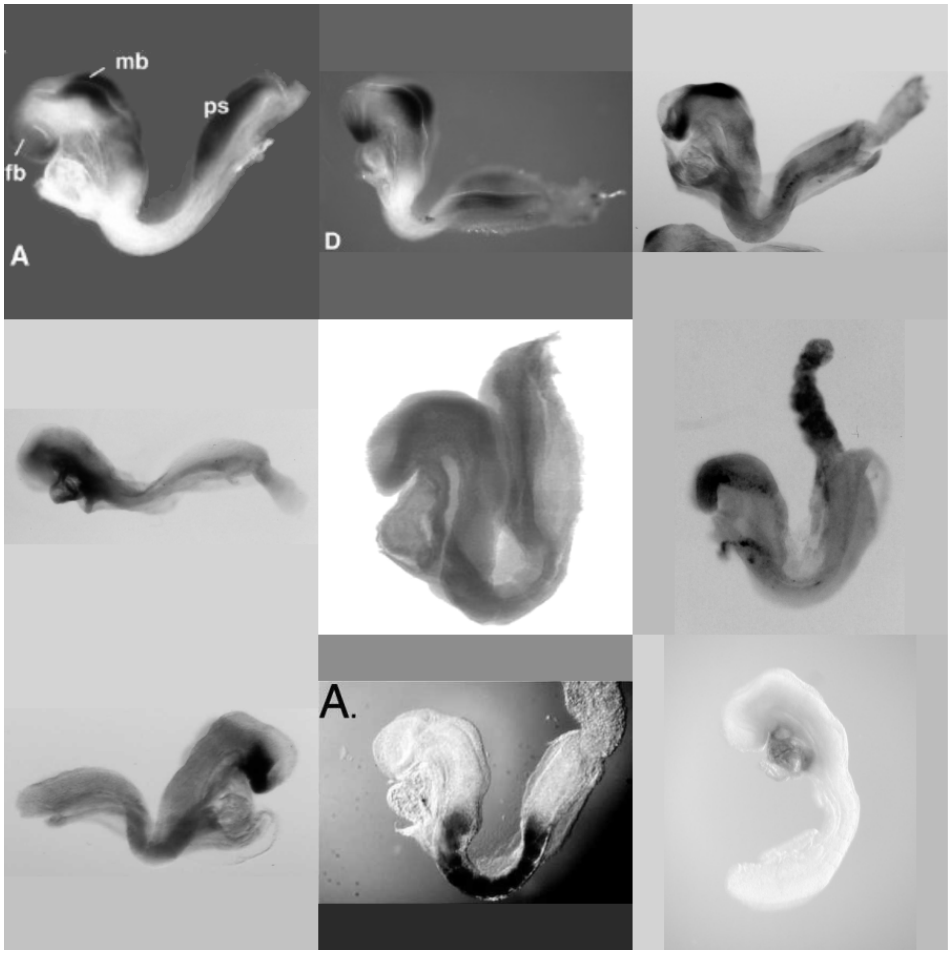


Figure 4: EMAGE TS12 dataset: Outer images are assay images to be mapped to an atlas image (centre).

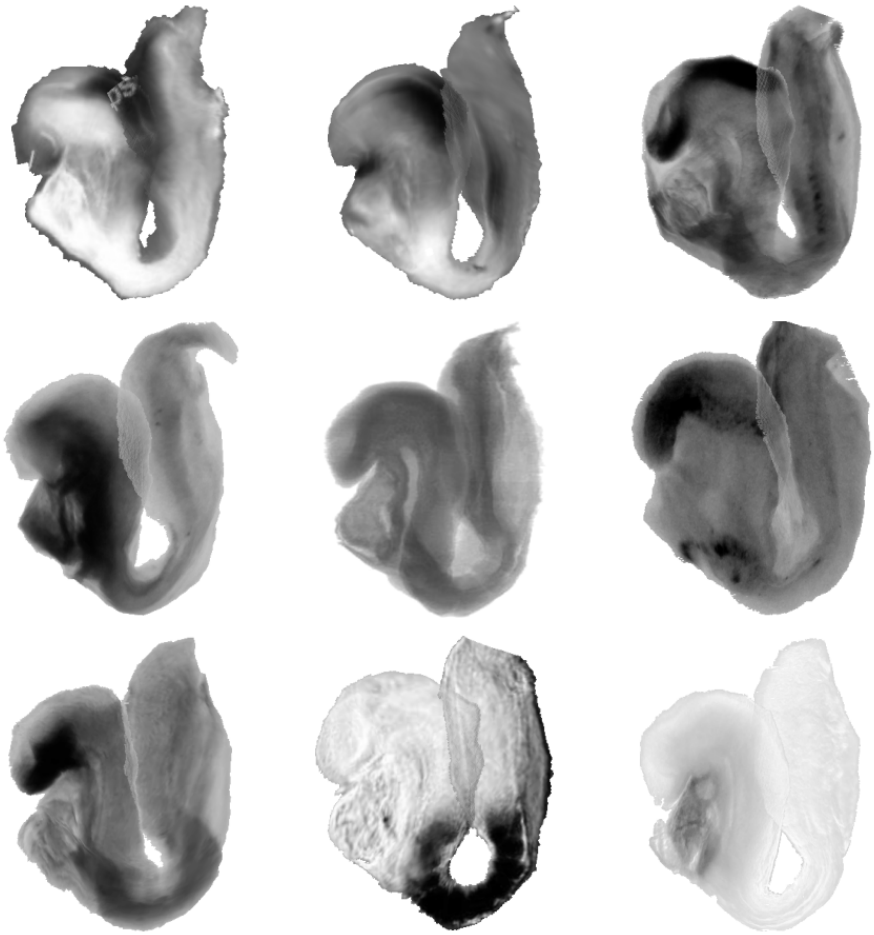


Figure 5: EMAGE TS12 dataset: Outer images are the registered assay images mapped to an atlas image (centre).

Intensity-Unbiased Force Computation for Variational Motion Estimation

Darko Zikic[1], Ali Khamene[2], and Nassir Navab[1]

[1] Chair for Computer Aided Medical Procedures
Technische Universität München
{zikic,navab}@cs.tum.edu

[2] Imaging and Visualization Dept., Siemens Corporate Research
Princeton, NJ, USA
ali.khamene@siemens.com

Abstract

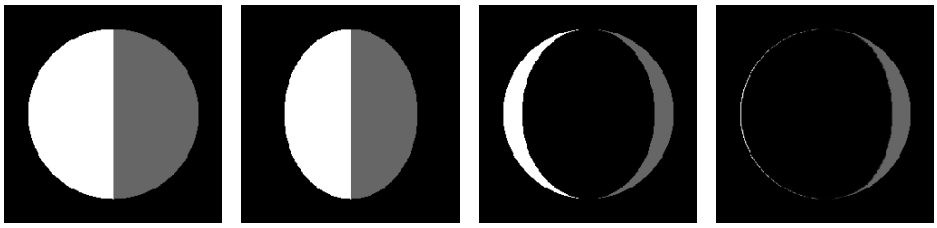
We show that the sum of squared differences, commonly used as a dissimilarity measure in variational methods is biased towards high gradients and large intensity differences, and that it can affect drastically the quality of motion estimation techniques such as deformable registration. We propose a method which solves that problem by recalling that the Euler-Lagrange equation of the dissimilarity measure yields a force term, and computing the direction and the magnitude of these forces independently. This results in a simple, efficient, and robust method, which is intensity-unbiased. We compare our method with the SSD-based standard approach on both synthetic and real medical 2D data, and show that our approach performs better.

1 Introduction and Motivation

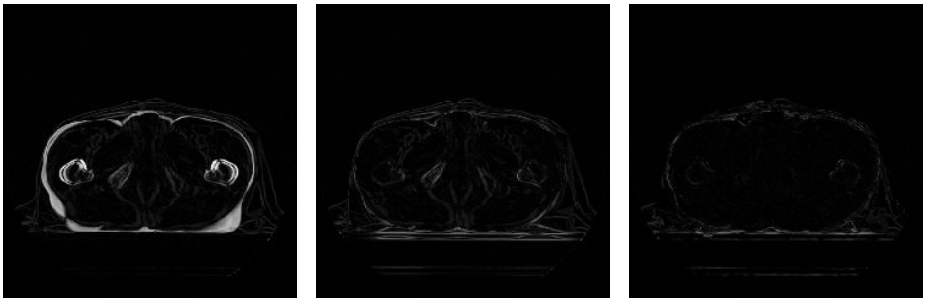
The sum of squared differences (SSD) dissimilarity measure is often used in computer vision applications because of its computational efficiency. Among other applications, it is used for variational methods for motion estimation such as optical flow or deformable registration. Since the optical flow and the deformable registration problem are basically equivalent, we will from now on focus on deformable registration.

The variational deformable registration task is posed as a minimization of a certain energy functional. The functional $\mathscr{I} = \mathscr{D} + \mathscr{S}$ consists of two components, the dissimilarity measure $\mathscr{D}$ to be minimized and the regularization component $\mathscr{S}$ which is used to enforce the well-posedness of the problem.

The minimization problem is mostly solved by deriving and solving the corresponding Euler-Lagrange partial differential equations. The Euler-Lagrange equation can be expressed as $A(\varphi)(\mathbf{x}) = f(\varphi)(\mathbf{x})$ [6]. Here φ is the deformation function while the differential operators A and f are resulting from the regularizer and from the dissimilarity measure respectively [6].



(a) Reference (b) Template (c) Initial Difference (d) Difference



(e) Initial Difference (f) Intensity-Biased (g) Intensity-Unbiased

Figure 1: **Upper Row:** Illustration of the biased behaviour of the unmodified SSD approach on a synthetic "coffee bean". (a) and (b) are the input images for the registration. The initial difference is presented in (c) and (d) shows the result of the algorithm. Due to the intensity values, the convergence is faster in the left-hand side of the "bean". This way, the regularizer is saturated by displacements which result mostly from high gradient regions of the image. This behaviour we refer to as *intensity-biased*. **Lower Row:** Comparison of the registration results on medical data. The images present differences between reference image R and the template T. (a) Initial difference image. Notice the large intensity differences at the border of the patient; (b) Difference after the unmodified SSD-based approach. While the large displacement at the border of the patient is registered well, the displacement inside the patient is not corrected with the SSD-based approach due to low intensity differences and gradients; (c) Difference after registration with our intensity-unbiased modification. The displacements inside the patient are also corrected.

An interpretation of the Euler-Lagrange equation is helpful as a motivation for the method we propose in this paper. The operator A can be seen as describing a reaction of a body, in our case the image, to a set of forces f. It relates the forces f to the deformation φ. The acting force f is computed by the derivation of the Euler-Lagrange equations from the dissimilarity term $\mathscr{D}$.

As shown in Figure 1, the force resulting from the SSD dissimilarity term is highly biased depending on the intensity values of the two input images. At a certain point, the force depends on the gradient of one of the images and the difference of the two images at this point. Together this leads to large forces at high gradients and large image differences.

Hence, the unmodified use of the SSD implies the assumption that objects evolve differently depending on their intensity and background. This would mean that for example bright objects in front of a dark background move or deform more than darker objects. We

refer to this behaviour as *intensity-biased*. A basic illustration of the problem is presented in Figure 1 on synthetic images and an example from a medical domain.

For some applications like morphing where the only goal is to make the images look similar, this is not a real drawback. Small intensity differences are not very noticeable, hence they do not have to be corrected as much as the larger ones. However if physical quantities like real motion are estimated from the results, the intensity-bias problem becomes crucial. In these applications, not the appearance is important but the correctness of the underlying displacement field. In many settings there is no reason why an intensity-biased assumption should be made. Besides not being justified for many applications, this assumption has several drawbacks. As shown in Figure 1 it can lead to slow convergence in certain areas with low gradients and/or low intensity differences between the reference and the template image. A consequence is that the regularizer is saturated by displacements resulting from the image regions with faster convergence. Thus the minimum solution of Equation (1) tends to be biased towards the regions of the image with high intensity differences.

In practice these drawbacks are present in many medical imaging modalities such as the computed tomography (CT) shown in Figure 1, where high gradients and intensity differences are present mostly on the boundary of the patient. The actually interesting regions however, are inside the patient and most often have a rather similar intensity.

To the best of our knowledge, this drawback of the SSD for respective applications has not yet been addressed in the literature.

We state that for the registration task the result should not be dependent on the particular intensity values of the input images at one point, that is the gradient and the intensity difference. This way, although the method stays intensity-based, it becomes intensity-unbiased.

Instead of computing the forces directly from the Euler-Lagrange equations corresponding to the SSD, we suggest to perform the computation of the force directions and the computation of the force magnitude independently. Using this approach, we modify the standard SSD method in order to establish a simple method which is intensity-unbiased. The developped algorithm can be seen in the framework of "'demons"' registration by Thirion [8].

With this modification, the direction of the forces is the same as with the standard unmodified SSD-based approach. The magnitude of the force vectors however is computed independently. The magnitude is not based on the intensity difference and the gradient at one point but it is computed according to the structure of the neighbourhood of the point. This way, the actual displacement of the image point, which can be estimated from the local neighbourhood, influences the magnitude of the force at that point rather than only the information at that point.

Some work that might be regarded as similar is the research on robust estimation of optical flow. To this end, several different error measures for the difference of the images have been proposed, see for example [2]. Among these measures is also the L1 norm, which results in the sum of absolute differences (SAD) dissimilarity measure when applied to the difference of the images. While the SAD is less intensity-biased than the SSD because the magnitude of the force is independent on the difference of the images, there is still a bias present based on the magnitude of the gradient. Furthermore, the motivation for the SAD in the context of robust estimation is not to remove the bias but to weigh the outliers less heavily. One other difference is that the work on robustness

is performed on the level of the energy functional while the approach presented here modifies the forces on the level of the Euler-Lagrange equations.

2 Method

This section presents a simple and efficient method which can be used for the separate computation of the direction and the magnitude of forces for variational algorithms. First however, we briefly introduce the used methods and notation.

2.1 Definitions and Problem Setting

The standard methods for deformable registration used in the following are described in detail in the literature, for example in [6, 1, 7, 4].

We define an image I to be a mapping $I : \Omega \rightarrow B$ from a respective functional space H from the d-dimensional image domain $\Omega = [0,1]^d \subset \mathbb{R}^d$ to a bounded interval of real numbers $B = [0,1] \subset \mathbb{R}$. For our applications the dimension is restricted to $d = 2,3$.

In the following, we consider the registration task of deforming the template image T such that it becomes similar to the reference image R. The deformation is described by the deformation function φ, which is a combination of the identity mapping Id and a displacement field u, such that $\varphi = Id + u$. Here, φ, Id and u are all functions from a space F, with $F = \{f | f : \Omega \rightarrow \Omega\}$.

For the computation of the deformation function we define an energy functional $\mathscr{I} : H \times H \times F \rightarrow \mathbb{R}$ to be minimized as

$$\mathscr{I}[R,T,\varphi] = \mathscr{D}[R,T,\varphi] + \alpha \mathscr{S}[\varphi] \ . \tag{1}$$

Here $\mathscr{I}$ consists of a dissimilarity term $\mathscr{D}$ and a regularizer, also known as smoothing operator, $\mathscr{S}$ whose influence is governed by a scalar parameter $\alpha \in \mathbb{R}$, $\alpha \geq 0$.

As a dissimilarity measure we use the SSD measure

$$\mathscr{D}_{SSD}[R,T,\varphi] = \frac{1}{2} \int_\Omega \left(R(\mathbf{x}) - T(\varphi(\mathbf{x}))\right)^2 \, d\mathbf{x} \ . \tag{2}$$

For the regularizer component $\mathscr{S}$ many different terms can be used. The actual regularizer component is not essential for the following since the problem is not restricted to the choice of one special regularization term. The most simple term is the isotropic homogeneous diffusion term and it is used in this paper. It is defined as

$$\mathscr{S}[\varphi] = \int_\Omega \sum_{i=1}^{d} |\nabla_{\mathbf{x}} \varphi_i(\mathbf{x})|^2 \, d\mathbf{x} = \int_\Omega \sum_{i=1}^{d} \langle \nabla_{\mathbf{x}} \varphi_i(\mathbf{x}), \nabla_{\mathbf{x}} \varphi_i(\mathbf{x}) \rangle \, d\mathbf{x} \ , \tag{3}$$

where ∇_x is the spatial gradient operator $\partial / \partial \mathbf{x}$ and $\langle \cdot, \cdot \rangle$ denotes the scalar product.

With this choice of dissimilarity and regularizer component, this model represents the well-known Horn and Schunck approach [5].

In order to minimize the functional $\mathscr{I}$ we first have to derive the Euler-Lagrange equation. The deformation function which solves this equation is set to be the solution of the registration problem. Because of the linearity of the functional, the Euler-Lagrange equations can be derived independently for the dissimilarity and the regularization term.

The Euler-Lagrange equation derived from the dissimilarity component $\mathscr{D}$ is

$$f_{SSD}(\varphi)(\mathbf{x}) = -\left[R(\mathbf{x}) - T(\varphi(\mathbf{x}))\right] \nabla_{\mathbf{x}} T(\varphi(\mathbf{x})) \ , \tag{4}$$

and will also be referred to as *force*. This paper deals with the modification of this term.[1] The Euler-Lagrange equation resulting from the regularizer $\mathscr{S}$ is

$$A(\varphi)(\mathbf{x}) = \Delta\varphi(\mathbf{x}) \ , \tag{5}$$

with Δ being the Laplacian operator.[2]

The resulting Euler-Lagrange equation for the functional $\mathscr{I}$ can be expressed using the differential operators A and f as

$$-\alpha A(\varphi)(\mathbf{x}) = f(\varphi)(\mathbf{x}) \ . \tag{6}$$

Here f stands for a force term corresponding to a chosen dissimilarity measure, such as f_{SSD} or f_{SAD} defined in the following. Using a discretization technique such as the standard finite difference scheme we obtain the discretized form of the upper equation. This non-linear partial differential equation is usually solved by a gradient descent method which is also the approach we take here [1]. Furthermore we can employ a Gaussian resolution pyramid in order to allow for larger displacements [1].

Since we compare the method presented in this paper also to the SAD-based approach, we present the definition of the SAD

$$\mathscr{D}_{SAD}[R,T,\varphi] = \int_{\Omega} \left|R(\mathbf{x}) - T(\varphi(\mathbf{x}))\right| \, d\mathbf{x} \ , \tag{7}$$

as well as the corresponding Euler-Lagrange equation

$$f_{SAD}(\varphi)(\mathbf{x}) = -\frac{R(\mathbf{x}) - T(\varphi(\mathbf{x}))}{\left|R(\mathbf{x}) - T(\varphi(\mathbf{x}))\right|} \nabla_{\mathbf{x}} T(\varphi(\mathbf{x})) \ . \tag{8}$$

This SAD-based force can be used as an alternative to the SSD-based term.

2.2 Modification of the Force Term

If we take a closer look at the Equation (4) we can see that two factors influence the magnitude of the forces. The main part of the SSD-based force term used in (4) is $\left[R(\mathbf{x}) - T(\varphi(\mathbf{x}))\right] \nabla_{\mathbf{x}} T(\varphi(\mathbf{x}))$. We see that the force at point $\mathbf{x}$ is proportional to the difference between the reference $R(\mathbf{x})$ and the deformed template $T(\varphi(\mathbf{x}))$ at this point. Furthermore, it is proportional to the gradient magnitude of the deformed template image $\|\nabla_{\mathbf{x}} T(\varphi(\mathbf{x}))\|$ at the same point which also depends on the intensities at the point.

To sum up, the above properties of the SSD-based measure cause larger forces for large image intensity differences and gradients than for others. Since the forces cause the deformation, this implies the assumption that points with certain intensities are more likely to move than others. The resulting behaviour is illustrated in Figure 1.

The same problem occurs for the SAD-based approach. In Equation (8), we can see that although the difference of the intensities is normalized, such that only the sign of the

[1] The force term is dependent not only on the point $\mathbf{x}$ but also on the images R and T and the deformation function φ. However, for the sake of simplicity we will drop these arguments in the following.

[2] For scalar-valued functions $g : \mathbb{R}^d \to \mathbb{R}$, the Laplace operator is $\Delta g = \sum_{i=1}^{d} \partial_{x_i, x_i} g$. For the vector-valued case, $G : \mathbb{R}^d \to \mathbb{R}^m$, the Laplace operator is defined component-wise as $\Delta G = (\Delta G_1, \ldots, \Delta G_m)^T$.

difference influences the force, the bias is still present through the unscaled magnitude of the gradient.

Therefore we propose a modification of the standard SSD-based force term in order to be able to perform intensity-unbiased registration. To this end we separate the computation of the direction and the magnitude of the force vectors.

In our approach we keep the direction of the vectors computed with the SSD-based method, which is the direction of $\nabla_{\mathbf{x}} T(\varphi)$. We make this choice since forces at edges of the body, orthogonal to the edges are meaningful and since the force directions do not cause the intensity-biased behaviour. The direction of the forces is then the normalized force

$$f_n(\mathbf{x}) = \frac{f_{SSD}(\mathbf{x})}{\|f_{SSD}(\mathbf{x})\|} = -\frac{R(\mathbf{x}) - T(\varphi(\mathbf{x}))}{|R(\mathbf{x}) - T(\varphi(\mathbf{x}))|} \cdot \frac{\nabla_{\mathbf{x}} T(\varphi(\mathbf{x}))}{\|\nabla_{\mathbf{x}} T(\varphi(\mathbf{x}))\|} \quad . \tag{9}$$

Regarding the magnitude of the force vectors, several alternatives are possible. We can model these alternatives by introducing a function $m : \Omega \to \mathbb{R}$ which assigns a magnitude to a force at every point of the domain.[3] The same general concept for computing the forces with separate terms for direction and magnitiude is used in [8]. This way we get the following modified formula for the computation of the forces

$$f(\mathbf{x}) = m(\mathbf{x}) f_n(\mathbf{x}) \quad . \tag{10}$$

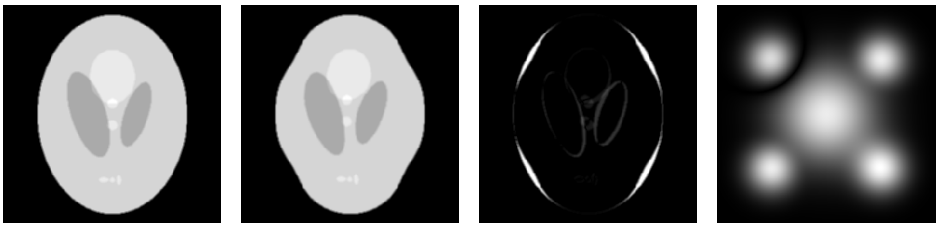
The most simple alternative for m is to not further modify the force term from Equation (9), that is $m(\mathbf{x}) = 1$, $\forall \mathbf{x} \in \Omega$. For images without noise this approach works satisfyingly, compare Table 1. In presence of noise however, the forces caused by noise might cause a wrong behaviour, since they have the same magnitude as all other force vectors. One possible approach to this problem might be to assume that the forces caused by random noise will have different directions and thus cancel each other out. This is actually also supported by the regularizer component $\mathscr{S}$ used in the method. For some tests, performed with Gaussian and uniform noise of different magnitudes on synthetic images this approach also produced good results. The question is however, how this approach behaves for general images from real applications. In order to develop a more robust approach without having to rely on the quality of noise, a modified method for magnitude computation is needed.

The basic idea behind the modification we introduce for the magnitude computation is to make the magnitude dependent not on the values at a single point but on the structure of the neighbourhood of the point.

The intuition behind the method is that if large displacements occur, there is a difference between the two input images not only at one point but also in the surrounding area. In order to make this decision intensity-unbiased we are not interested in how large the difference of the intensities is but only if it is present.

We implement the above intuition by performing the following steps. First, we compute the difference between the reference and the template image $D(\mathbf{x}) = R(\mathbf{x}) - T(\varphi(\mathbf{x}))$. In order to make the method more robust in presence of noise, we filter the difference $D(\mathbf{x})$ by a median filter and take the absolute value of the result. In our experiments, this was enough to remove the effects caused by noise. This yields a signal, which is very close to

[3] Again, m is dependent on more parameters than only the domain point (R,T,φ) which we drop for simpler notation.



(a) Reference (b) Template (c) Initial Difference (d) Displacement Field

Figure 2: Input for the study on synthetic data. We display the input reference and template images, the initial difference and the displacement field used to generate the template from the reference. The results of the experiments are presented in Table 1 and Figure 3.1.

zero in the regions which have no real difference and larger in other regions, depending on the intensities. Now we perform a thresholding step in order to remove the still present bias and set all values below the threshold to 0 and the values above or equal the threshold to 1. This way, the magnitude is no longer dependent on the magnitude of the intensity differences of the two images but only on their existence.

3 Results and Evaluation

We test the proposed method on synthetic and medical 2D data. While the medical data confirm the practicability of the proposed method for real applications, the synthetic test allow us a quantitative evaluation.

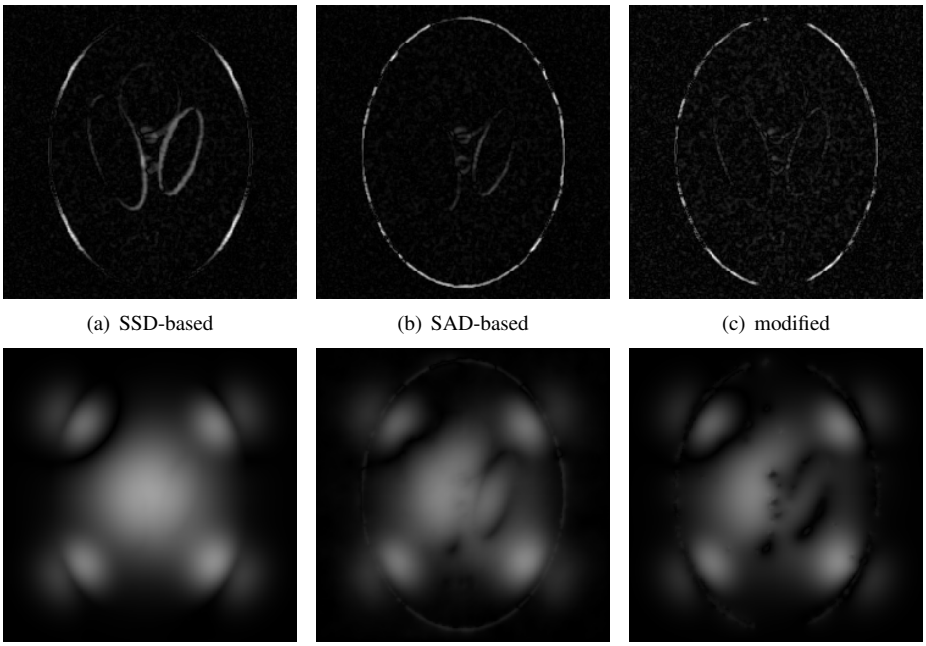
We use a gradient descent approach in order to overcome the non-linearity of the deformable registration problem [1] and a homogeneous isotropic regularization term for all experiments. For the solution of the arising algebraic linear systems we employ a multigrid solver [3]. The algorithm is implemented in Matlab with no performance tuning. The runtime for one iteration step (computing forces, solving the linear system and applying the deformation) of the gradient descent method is approximately 0.75 seconds for a 256^2 image. The runtime overhead needed for the proposed method is approximately 10-15%.

For the comparison of the different methods we try to set the parameters as similar as possible in order to achieve a fair evaluation. For all methods we use the same regularization parameter $\alpha = 1.0$, the same number of iterations, which are sufficient for convergence of all methods, and we scale the forces, such that the resulting mean force magnitude is approximately the same for all methods.

3.1 Evaluation on Medical Data

The medical data used for this experiment are 2D slices from an abdominal CT scan. Because in this case we deal with real data we have no ground truth displacement. So we can only perform a qualitative comparison of the different methods.

We compare the SSD-based method and our approach by visual inspection of the difference images between the reference R and the deformed template image T after the registration process. We can see a clear improvement in small gradient and low intensity



(a) SSD-based (b) SAD-based (c) modified

(d) SSD-based (e) SAD-based (f) modified

Figure 3: Comparison of the results on synthetic data with 10% uniform noise. The input data is shown in Figure 2. The upper row presents the difference images after the respective algorithms. Please notice that the maximal errors are lower for the modified method (range:0-0.25) than for the SSD and SAD-based approaches (range: 0-0.45). The lower row illustrates the improvement in the displacement field by displaying the norm of the error fields, compare also Table 1. Here the black values mean that the error of the computed displacement is low. Clearly, the modified intensity-unbiased approach performs better than the SSD and the SAD-based method in the inner area with low intensity differences and low gradients. Of course, the error stays large for all methods in regions with no structure.

difference areas when using our method. The results are displayed in Figure 1. The registration in this experiment is performed using a Gaussian resolution pyramid.

3.2 Evaluation on Synthetic Data

Comparison of difference images has a drawback that only the apparent similarity is compared - small intensity differences are not visible. For our purposes however, it is more important to examine the computed displacement fields. In order to be able to perform this quantitative evaluation of the proposed method, we use synthetic data sets. We employ a ground truth displacement field in order to compute the template from the reference image and we also use this synthetic setting to test the methods in presence of noise.

We compare the performance of the standard SSD-based method, the SAD-based method, the simple modification where all force magnitudes are normalized to unity and finally our approach.

Results on Image		SSD	SAD	Unit Force	Our Method
no noise	mean	1.3671	1.1311	0.9525	0.9793
	std. dev.	1.5760	1.2636	1.1066	1.1371
10% uniform	mean	1.3802	1.1605	1.5652	1.0047
	std. dev.	1.5839	1.2666	1.2098	1.1634
20% uniform	mean	1.3890	1.2010	1.5693	1.1471
	std. dev.	1.5918	1.3232	1.2286	1.2216

Results on ROI		SSD	SAD	Unit Force	Our Method
no noise	mean	3.2495	2.5384	2.0943	2.2077
	std. dev.	1.4085	1.1965	1.1692	1.1288
10% uniform	mean	3.2639	2.5792	2.4610	2.2448
	std. dev.	1.1500	1.2015	1.3454	1.1833
20% uniform	mean	3.2798	2.7180	2.6505	2.4385
	std. dev.	1.4202	1.2494	1.3602	1.2561

Table 1: Quantitative results of the phantom study. The upper table shows the results for the complete image, the lower for the region of interest inside the phantom where the low intensity differences and gradients are dominating. We compute the norm of difference between the ground truth u_{GT} and the estimated displacement field u and give the mean value and the standard deviation of $\|u_{GT} - u\|$. The mean of the u_{GT} is 3.5459 with a standard deviation of 1.5477. The methods tested are the SSD and SAD-based approach, the simple modification with forces normalized to unity and our intensity-unbiased modification. Tests were performed with no noise and with 10% and 20% uniform noise. The modified method presents a clear improvement over both the SSD and the SAD-based approach. All values are in pixel units.

The used images are generated with the Matlab inbuilt *phantom* function and the displacement field is a combination of Gaussians in each dimension, see Figure 2. For experiments with noise with use a uniform noise in range of 10% and 20% of the maximal image intensities, that is $[0, 0.1]$ and $[0, 0.2]$. For the experiments we do not use a Gaussian pyramid since we want to isolate the behaviour of the methods and the pyramid implicitly influences the results in presence of noise by smoothing on the low-resolution levels.

We compare the methods by examining the norm of difference field $e = \|u_{GT} - u\|$ between the ground truth u_{GT} and the displacement field u estimated by the respective method. The error norm is inspected by computing the mean value and the standard deviation of e.

The parameters for the proposed approach were determined experimentally. However, they did not have to be changed during the tests. The size of the neighbourhood for the median filter is 5×5 and the threshold is $\varepsilon = 0.025$.

The results of the experiments are summarized in Table 1 and Figure 3.1. We can see a clear improvement of the error of the displacement field with our method in regions with low intensity differences and small gradients. This leads to an overall better performance of our approach. The simple force-normalizing approach performs well for the case with no noise, however it is very sensitive to noise in the homogeneous areas.

4 Summary and Further Work

We address the problem of the intensity-bias of the SSD measure for variational motion estimation methods. The proposed solution separates the computation of the direction and the magnitude of the forces, which are usually yielded from the Euler-Lagrange equations corresponding to the SSD. While we keep the direction of the forces, we modify the magnitude in such way that it is not biased to large intensity differences or high gradients. The robustness is improved by setting the magnitude to zero in regions where the forces are resulting only from the presence of noise and not from real deformations. The proposed method is tested on 2D synthetic images and real medical data and shows a better performance than the standard SSD and SAD-based techniques.

Our further work on this topic will include an integration of the methods presented here into an existing framework for deformable registration of 3D medical images. Furthermore, we plan to investigate the behaviour of other dissimilarity measures like the Cross-Correlation, Correlation Ratio and Mutual Information with respect to the problem of the intensity-bias.

References

[1] Luis Álvarez, Joachim Weickert, and Javier Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, 2000.

[2] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Fourth International Conference on Computer Vision*, pages 231–236, 1993.

[3] Andrés Bruhn, Joachim Weickert, Christian Feddern, Timo Kohlberger, and Christoph Schnörr. Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005.

[4] Gerardo Hermosillo. *Variational Methods for Multimodal Image Matching*. PhD thesis, Université de Nice - Sophia Antipolis, 2002.

[5] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980.

[6] Jan Modersitzki. *Numerical methods for image registration*. Oxford University Press, 2004.

[7] Christoph Schnörr and Joachim Weickert. Variational image motion computation: Theoretical framework, problems and perspectives. In Gerald Sommer, Norbert Krüger, and Christian Perwass, editors, *DAGM-Symposium*, Informatik Aktuell, pages 476–488. Springer, 2000.

[8] J.P. Thirion. Image matching as a diffusion process: an analogy with maxwells demons. *Medical Image Analysis*, 2(3):243–260, 1998.

GPU implementation of deformable pattern recognition using prototype-parallel displacement computation

Yoshiki Mizukami, Katsumi Tadamura
Yamaguchi University
2-16-1 Tokiwadai, Ube, 754-8611 Japan
mizu@yamaguchi-u.ac.jp

Abstract

In this paper, for the reduction of the computation time of a deformable approach to pattern recognition, prototype-parallel displacement computation on GPUs (PPDC-GPU) is proposed. The displacement computation used in this study has the virtue of simplicity and consists of locally parallel processing, therefore it is suitable for the implementation on graphical processing units (GPUs). In the proposed method, large plates of image and displacement function are generated from input images, prototypes, and displacement functions on the main memory, and then these plates are transferred collectively to the video memory. After computing the displacement between the input image plate and the prototype image plate on the GPU, the displacement function plates are transferred back to the main memory. The simulation results show that PPDC-GPU reduces the computation time to less than 10% of the ordinary implementation on CPUs. This study especially focused on handwritten character recognition, since it is one of the most fundamental and important problems in the field of computer vision and pattern recognition. However the proposed framework can be widely applied to other problems, for instance, face recognition or object recognition.

1 Introduction

In these years, the performance of graphical processing units (GPUs) has been remarkably improved as hardware for computer graphics [1]. The main features are fast access to video memory, high frequency clock, and multiple programmable pipelines. Many researchers have applied GPUs to various numerical problems [2], and made it possible to shorten the computation time without sacrificing the degree of accuracy. In the field of computer vision, Yang et al. implemented a stereo matching method on GPUs [3]. Fung et al. utilized plural GPUs for speeding up the eigenspace approach [4]. Strzodka et al. implemented on GPUs an image registration method proposed by Clarenz et al. in 2002 [5, 6]. In addition, signal processing problems were also studied such as fast Fourier transformation and artificial neural networks [2, 7]. Recently linear algebra operators were effectively implemented on GPUs [8, 9] and applied to physical simulations such as fluid motion.

Deformable approaches have been studied since 1970s. Widrow pointed out that rigid templates cannot deal with the shape variation of the input image, and proposed a deformable template [10]. Burr proposed a dynamic cooperative method for computing the deformation between images [11]. Kass et al. proposed active shape models for object tracking [12]. Bajcsy et al. proposed a multiresolutional elastic matching method for analyzing brain images [13]. Yuille proposed a deformable template for face recognition [14]. Lades et al. proposed an object recognition method using dynamic link architecture [15], where Gabor feature was employed for the image representation and a simulated annealing method was utilized for minimizing the evaluation function. Jain et al. proposed a deformable template in the framework of Bayesian optimization algorithm for object detection [16].

Handwritten character recognition is one of the fundamental and important problems in the field of image processing and has greatly influenced other problems such as face recognition and object recognition. Many researchers have investigated the processes of feature extraction and classification in this area. Deformable or elastic approaches in which the displacement between input and prototype patterns is computed have also been studied (e.g. [17, 18, 19, 20, 21, 22]). After reviewing the previous works, it was thought that the complexity of the procedure and the high computation cost were the main drawbacks of most of these deformable approaches.

In 1994, a deformable approach to handwritten character recognition was proposed [23, 24], which was originally inspired by a regularization method for a stereo matching problem [25]. The deformable approach has the virtue of simplicity, and is also applicable to handwritten Chinese character by using the processes of feature extraction and statistical classification together [26]. Although the simple iterative equation could efficiently compute the displacement, further reduction of computation time was desired. To mitigate this problem, gradual prototype elimination was proposed in 2000 [27], where the computation time was reduced to $1/3$ without degradation of recognition performance. However, the required computational load still fails to match the ability of present common computers.

This study proposed an implementation of the displacement computation on GPU processors and achieved the drastic reduction of computation time, making the deformable recognition approach more practical in the field of computer vision and pattern recognition. The implementation was designed by considering the characteristics of GPUs and the deformable recognition approach. The NIST handwritten database was used in the simulations to assess the validity of the proposed method. The focus was on handwritten character recognition, however the proposed implementation can be widely applied to other problems, for instance, face recognition or object recognition.

2 Proposed method

Candidate prototypes are selected for computing the displacement based on the result of rough classification. The number of the candidate prototypes is referred to as L. In the final classification, the distance between an input pattern and a prototype is determined based on the Euclidean distance considering the computed displacement. The category of the prototype with the smallest distance is employed as the classification result. The proposed method is mainly composed of two concepts, namely, displacement computation

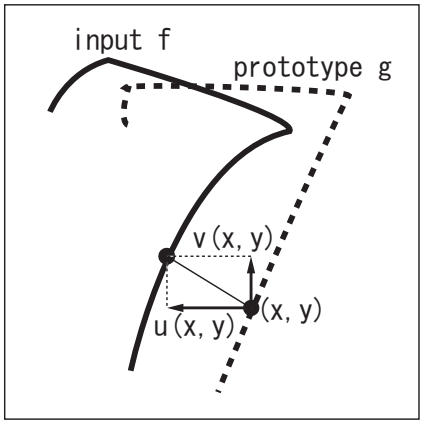


Figure 1: displacement function (u, v)

with the coarse-to-fine strategy and a novel implementation on GPUs. These concepts are described below.

2.1 Displacement computation

The displacement computation in the proposed method [23, 24] was originally based on a regularization method for a stereo matching problem [25]. As shown in Fig. 1, input and prototype images are referred to as $f(x,y)$ and $g(x,y)$, respectively. The horizontal and vertical displacements at the coordinate (x,y) on g are represented by the set of displacement functions, $(u(x,y),v(x,y))$. The optimum displacement between the two images can be obtained by minimizing the following functional E,

$$E(u,v) = P(u,v) + \lambda S(u,v), \tag{1}$$

$$P(u,v) = \iint (f(x+u,y+v) - g(x,y))^2 dxdy, \tag{2}$$

$$S(u,v) = \iint (u_x^2 + u_y^2 + v_x^2 + v_y^2) dxdy, \tag{3}$$

where P is the Euclidean distance considering the computed displacement, S is a stabilizing functional which imposes a smoothness constraint on (u,v), and λ is a so-called regularization parameter controlling the effect of S. From the framework of calculus of variations, the following iterative equations are derived from Eq. (1),

$$u^{[t+1]}(x,y) = \bar{u}^{[t]} - \frac{1}{4\lambda} f_x(x+u^{[t]},y+v^{[t]}) \\ (f(x+u^{[t]},y+v^{[t]}) - g(x,y)), \tag{4}$$

$$v^{[t+1]}(x,y) = \bar{v}^{[t]} - \frac{1}{4\lambda} f_y(x+u^{[t]},y+\bar{v}^{[t]}) \\ (f(x+u^{[t]},y+v^{[t]}) - g(x,y)), \tag{5}$$

where $(\bar{u}^{[t]}, \bar{v}^{[t]})$ is the average of four neighborhoods. That is,

$$\begin{aligned} \bar{u}^{[t]}(x,y) &= (u^{[t]}(x+1,y) + u^{[t]}(x,y-1) \\ &\quad + u^{[t]}(x-1,y) + u^{[t]}(x,y+1))/4, \qquad (6) \\ \bar{v}^{[t]}(x,y) &= (v^{[t]}(x+1,y) + v^{[t]}(x,y-1) \\ &\quad + v^{[t]}(x-1,y) + v^{[t]}(x,y+1))/4. \qquad (7) \end{aligned}$$

It should be noted that the iterative equations of (4) and (5) require the subpixel value of f, f_x and f_y in the second terms of the right side, which are calculated by using the bilinear interpolation with the four surrounding pixel values.

In this study, for the improvement of computation stability, instead of Eqs. (4) and (5), the modified iterative equation was adopted [23, 24],

$$\begin{aligned} u^{[t+1]}(x,y) &= \bar{u}^{[t]} - \frac{1}{4\lambda} f_x(x+\bar{u}^{[t]}, y+\bar{v}^{[t]}) \\ &\quad (f(x+\bar{u}^{[t]}, y+\bar{v}^{[t]}) - g(x,y)), \qquad (8) \\ v^{[t+1]}(x,y) &= \bar{v}^{[t]} - \frac{1}{4\lambda} f_y(x+\bar{u}^{[t]}, y+\bar{v}^{[t]}) \\ &\quad (f(x+\bar{u}^{[t]}, y+\bar{v}^{[t]}) - g(x,y)), \qquad (9) \end{aligned}$$

where the averaged displacement function $(\bar{u}^{[t]}, \bar{v}^{[t]})$ is also used in the second terms of the right side instead of $(u^{[t]}, v^{[t]})$.

To avoid the solution being trapped in the local minima and to compute the displacement with a small number of iterations, the coarse-to-fine strategy should be very promising [28, 29]. The number of stages for the coarse-to-fine strategy is referred to as N. $N-1$ lower resolutional sets of images with $1/(2^{N-n})^2$ of the original size, f_n and g_n, are produced from the original set in preparation for computing the displacement $(1 \leq n \leq N)$. The displacement computation is started at the lowest stage, and the displacement computed at the n-th stage is used as the initial value at the (n+1)-th stage. Finally, the displacement functions the same size of the original images are obtained at the N-th stage.

2.2 Prototype-parallel displacement computation on GPUs; PPDC-GPU

As mentioned in 1, we believe that one of the main drawbacks of most deformation approaches is its expensive computation cost. It is very important to drastically reduce the computation time. Therefore, we discuss how to implement our deformable recognition approach on GPUs by considering both characteristics of GPUs and the deformable recognition approach. The main features of GPUs are fast access to video memory, a high frequency clock, and multiple programmable pipelines, and then it is easy to understand that these features are suitable for locally parallel computation. Therefore the GPU implementation of our iterative displacement computation seems to be very promising.

A GPU characteristic that should be more deeply considered is that the multiple pipeline processing is not suitable for dealing with many small images but for a few large images. In addition, the frequent transfer of small images between the main memory and the video memory might be a bottleneck in the whole computation.

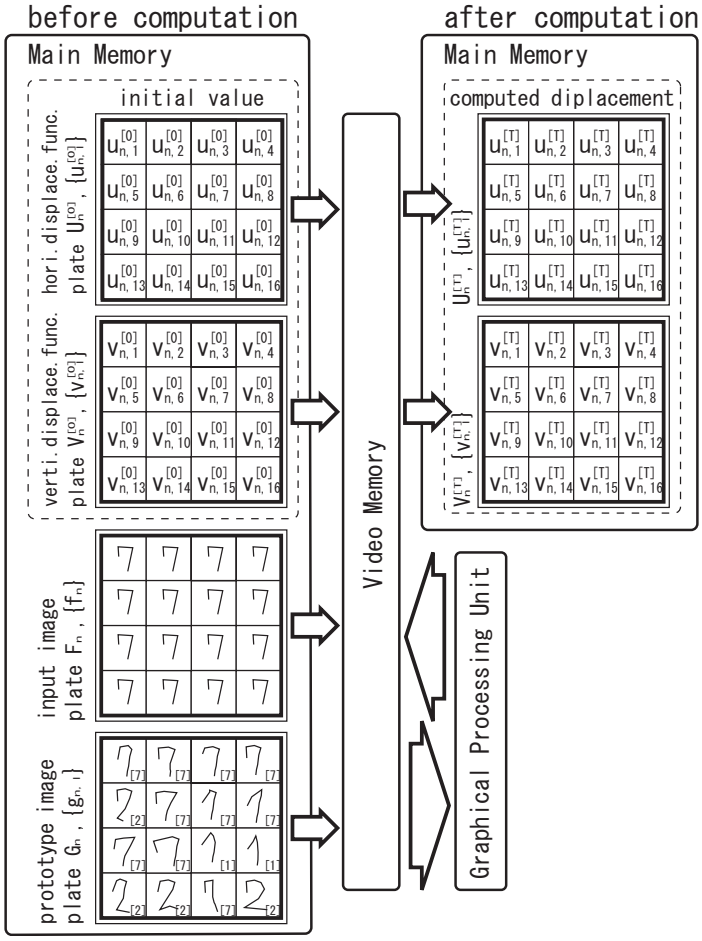


Figure 2: Prototype-parallel displacement computation on GPUs; PPDC-GPU

To overcome the above-mentioned problem, this study proposes *prototype-parallel displacement computation on GPUs (PPDC-GPU)*, whose concept is illustrated in Fig. 2. A large prototype image plate, $G_n(x,y)$, is generated by putting L prototype image $g_{n,l}$'s in lattice geometry on the main memory $(1 \leq l \leq L)$. In a similar fashion, an input image plate, $F_n(x,y)$, is generated from f_n. We can also generate displacement function plates, $(U_n^{[0]}(x,y), V_n^{[0]}(x,y))$, from L initial displacement functions $(\{u_{n,l}^{[0]}\}, \{v_{n,l}^{[0]}\})$. These generated plates are transferred to the video memory only once at the n-th stage. The displacement computation is applied to the set of image plates instead of individual sets of images. After computing the displacement on the GPU, the plates of the computed displacement function, $(\{u_{n,l}^{[T]}\}, \{v_{n,l}^{[T]}\})$, are transferred back from the video memory to the main memory. As a result, PPDC-GPU is expected to reduce the number of the interruptions in the multiple pipeline processing and the data transfers between the main memory and video memory.

Please note that, in Eqs. (8) and (9), the computation of $(\{u_{n,l}^{[t+1]}\}, \{v_{n,l}^{[t+1]}\})$ is based

on the pre-computed displacement $(\{u_{n,l}^{[t]}\},\{v_{n,l}^{[t]}\})$. To enable such iterative computation on GPUs, this study employs the Framebuffer Object extension [2]. Since the 32-bit floating-point arithmetic on GPUs is provided through a set of OpenGL extensions, GL_ARB_texture_float, the degree of accuracy on GPUs compares favorably with CPUs.

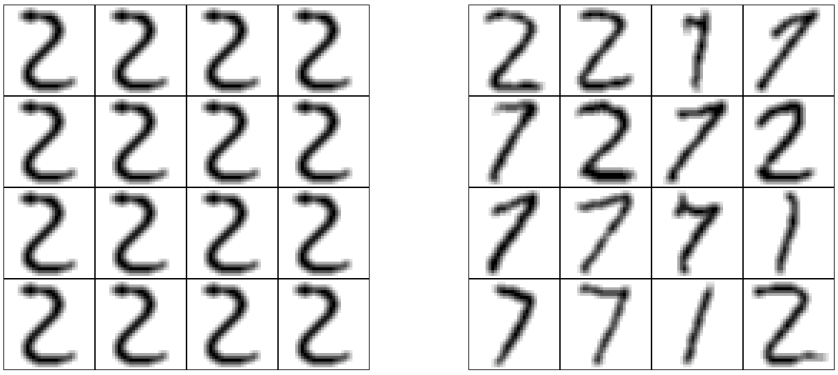
The details about the GPU implementation of the iterative computation are described below. The averaging procedure of displacement function in Eqs. (6) and (7) should not be applied beyond the borders between different prototype regions, therefore border information about the presence or location of neighborhoods is stored as an image format on the video memory and referred in the averaging procedure. In addition, to avoid the memory access to the coordinate of $(x+u, y+v)$ from wandering off into different prototype regions, a clamp procedure is employed.

3 Simulations

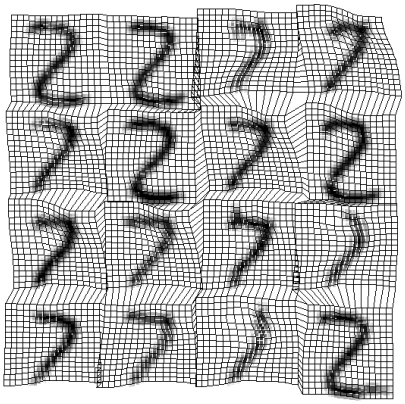
The NIST special database (HSF7) contains over 60,000 digit samples. Eight thousand input patterns (800 per category) and 8,000 prototype patterns (800 per category) are independently chosen at random. The number of stages for the coarse-to-fine strategy, N, was set to 3. Prior to processing, all character images were reduced to 32×32 pixels, and the characters on the images were normalized to 28×28 pixels and placed at the center. The values of $\lambda_n (n = 1,2,3)$ were set to $\{0.10, 0.10, 0.50\}$ and the number of iteration, T, at each stage was set to 50. Euclid distance between the coarsest input and prototype images was adopted as a measure for the rough classification. These parameters were empirically decided based on the result of small-scale preliminary simulations. The main specifications of the computer used in the simulations were Intel Pentium4 CPU (3.2GHz), DDR2 main memory (1GB), NVIDIA GeForce 7800 GT with PCI-Express 16x (video memory 128MB). Microsoft Visual C++ .NET 2003 and NVIDIA Cg Toolkit 1.4 were utilized for the programming environment [1].

Figure 3 (a) and (b) show a part of image plates at the latest stage, F_N and G_N, respectively. We can see that the same input image f_N is put on the input image plate F_N in lattice geometry, and that candidate prototypes $g_{N,l}$ having a shape similar to the input image f_N are put on the prototype image plate G_N. Figure 3 (c) illustrates the computed displacement with deformed grid and prototype patterns. It should be noted that the shapes of the deformed prototypes became more similar to the input f_N and that there were discontinuities of the displacement on the borders between different prototype regions.

Figure 4(a) shows the computation time of the proposed PPDC-GPU for recognizing a single input handwritten character as a function of the number of prototypes L. For comparison, the figure also shows the computation time of the ordinary implementation using only the CPU and that of *prototype-sequential displacement computation on GPUs (PSDC-GPU)*, where L prototype images and L displacement functions were transferred separately between the main memory and the video memory without generating the image plates. The proportionality relations between L and the computation time can be seen in all three implementations. Compared with CPU implementation, the PSDC-GPU reduced the computation time to about 20%. It should be noticed that PPDC-GPU provided a drastic reduction to less than 10%, especially to less than 7% using greater or equal to 200 as L. Figure 4(b) shows the recognition rate of our implementations as a function of L. Through the simulations, it was confirmed that these three implementations gave the

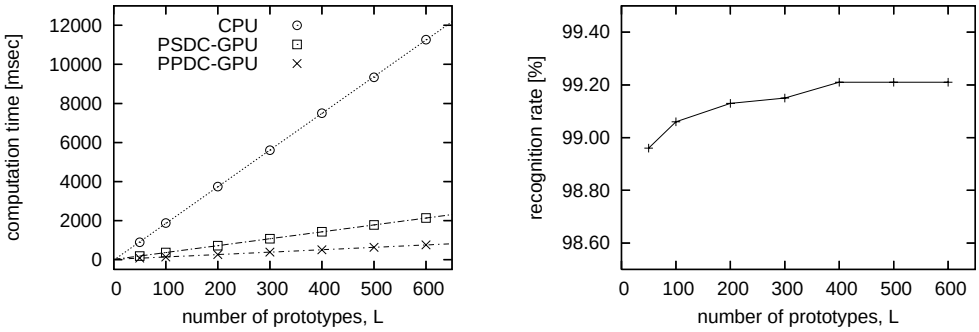


(a) input image plate F_N (b) prototype image plate G_N



(c) displacement function with deformed g_N's

Figure 3: image and function plates in prototype-parallel displacement computation on GPUs



(a) computation time (b) recognition rate

Figure 4: total computation time and recognition rate using different number of prototypes

(*msec*)	CPU	PSDC-GPU	PPDC-GPU
displacement computation	7400	1220	245
data transfer	-	109	83
plate image generation	-	-	90
others	98	98	90
total processing time	7498	1427	508

Table 1: necessary time for respective processes in three implementations at $L = 400$.

same rates. The best recognition rate, 99.21%, was obtained with L=400, 500 and 600, which was superior to the result of simple matching without computing the displacement, 96.5%, obtained in the rough classification.

Table 1 shows the necessary time for respective processes in three implementations. Compared with PSDC-GPU, it was clarified that PPDC-GPU could reduce the necessary time for the displacement computation drastically. This result seems to confirm that the generation of image plates is useful for the efficient pipeline processing. However, the adoption of image plates could not considerably reduce the time for data transfer. It should be also noticed that the necessary time for generating the image plates resulted in 90 msec and the sum of them, 173 msec, corresponded to 34% of the total processing time.

4 Conclusion

In this study, we investigated speeding up a deformable approach to pattern recognition. By considering the characteristics of GPUs and the deformable recognition approach, *prototype-parallel displacement computation on GPUs (PPDC-GPU)* was proposed. Simulations were conducted based on the NIST handwritten database, and it was clarified that the proposed implementation reduced the computation time to less than 10% compared with the ordinary implementation on CPUs.

In future tasks, we will investigate the generation of image plates on the video memory instead of the main memory, since it is expected to eliminate the heavy data transfer between the main memory and the video memory, and to generate the image plates more quickly with the fragment processor on GPUs. The video memories of the recent consumer graphic cards are getting larger and it seems to be possible to store the set of prototypes in the video memory. Another future task is the extension for three dimensional object recognition.

We intentionally kept the recognition method as simple as possible, since the main proposes of this study were to show how to implement the deformable recognition approach on GPUs efficiently and to clarify how much the GPU implementation could reduce the computation time without degrading the accuracy, However, in order to obtain higher recognition performance, it is essential to employ other procedures such as statistical classification and feature extraction [26]. Obviously, the dimensional reduction by feature extraction is also helpful for saving the computation time.

Since handwritten character recognition is one of the most fundamental and important topics in the field of computer vision and pattern recognition, this study discussed the validity of the proposed method from its viewpoint. However the proposed framework can

be widely applied to other problems, for instance, face recognition or object recognition.

Acknowledgement

This study was partially supported by JSPS Grants-in-Aid for Science Research (16700208).

References

[1] R. Fernando and M. J. Kilgard. *The Cg Tutorial : the definitive guide to programmable real-time graphics*. Addison-Wesley, 2003.

[2] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. In *Eurographics 2005, State of the Art Reports*, pages 21–51, August 2005.

[3] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Proceedings of International Conference of Computer Vision and Patter Recognition*, pages 211–217, 2003.

[4] J. Fung and S. Mann. Using multiple graphics cards as a general purpose parallel computer : Applications to computer vision. In *Proceeding of International Conference of Pattern Recognition*, volume 01, pages 805–808, 2004.

[5] R. Strzodka, M. Droske, and M. Rumpf. Image registration by a regularized gradient flow - a streaming implementation in DX9 graphics hardware. *Computing*, 73(4):373–389, 2004.

[6] U. Clarenz, M. Droske, and M. Rumpf. Towards fast non-rigid registration. *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, 313:67–84, 2002.

[7] K Moreland and E. Angel. The FFT on a GPU. In *Proc. of SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 112–119, 2003.

[8] J. Kruger and R. Westermann. Linear algebra operators for GPU implementation of numerical algorithms. In *Proc. SIGGRAPH 2003*, pages 908–916, 2003.

[9] J. Bolz, I. Farmer, and E. G. P. Schroder. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. In *Proc. SIGGRAPH 2003*, pages 917–924, 2003.

[10] B. Widrow. The rubber mask technique – I. pattern measurement and analysis. *Pattern Recognition*, 5(3):175–197, March 1973.

[11] D.J. Burr. A dynamic model for image registration. In *Proc. IEEE Comput. Soc. Conf. Pattern Recognition and Image Processing*, pages 17–24, August 1979.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[13] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46(1):1–21, 1989.

[14] A.L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, 1991.

[15] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C.v.d. Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE-C*, 42(3):300–311, March 1993.

[16] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable template. *IEEE Trans. PAMI*, 18(3):267–278, March 1996.

[17] D.J. Burr. Elastic matching of line drawings. *IEEE Trans. PAMI*, 3(6):708–713, November 1981.

[18] T. Wakahara. Shape matching using LAT and its application to handwritten numeral recognition. *IEEE Trans. PAMI*, 16(6):618–629, June 1994.

[19] A. Daffertshofer and H. Haken. A new approach to recognition of deformed patterns. *Pattern Recognition*, 27(12):1697–1705, December 1994.

[20] H. Sakoe, M. Masroor, and Y. Katayama. One dimensional-two dimensional dynamic programming matching algorithm for character recognition. *IEICE Trans. Inf. & Syst.*, E77-D(9):1047–1054, 1994.

[21] M. Revow, C. K. I. Williams, and G. E. Hinton. Using generative models for handwritten digit recognition. *IEEE Trans. PAMI*, 18(6):592–606, 1996.

[22] A.K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. PAMI*, 19(12):1386–1391, December 1997.

[23] Y. Mizukami, K. Koga, and T. Torioka. A handwritten character recognition system using hierarchical extraction of displacement. *IEICE*, J77-D-II(12):2390–2393, December 1994 (in Japanese).

[24] Y. Mizukami and K. Koga. A handwritten character recognition system using hierarchical displacement extraction algorithm. In *Proc. 13th Int. Conf. Pattern Recognition*, volume 3, pages 160–164, 1996.

[25] R. March. Computation of stereo disparity using regularization. *Pattern Recognition Letters*, 8(3):181–188, March 1988.

[26] Y. Mizukami. A handwritten Chinese character recognition system using hierarchical displacement extraction based on directional features. *Pattern Recognition Letters*, 19:595–604, 1998.

[27] Y. Mizukami, T. Sato, and K. Tanaka. Handwritten digit recognition by hierarchical displacement extraction with gradual prototype eliminations. In *Proc. 15th Int. Conf. Pattern Recognition*, volume 3, pages 339–342, 2000.

[28] F. Glazer. Multilevel relaxation in low-level computer vision. *Multiresolution Image Processing and Analysis*, pages 312–330, July 1984.

[29] N. Yokoya. Dense matching of two views with large displacement. In *Proceedings of the 1st IEEE ICIP*, volume I, pages 213–217, November 1994.

INVITED TALK: ABSTRACT

Algorithms for Nonrigid Factorization

Andrew W. Fitzgibbon

Microsoft Research Ltd
7 JJ Thomson Avenue
Cambridge CB3 0FB
England

awf@microsoft.com
http://research.microsoft.com/∼awf

Matrix factorization underpins several recent approaches to deformable object tracking and modelling. In long sequences with occlusion, this factorization must be performed despite missing data, a problem which has seen significant attention in recent computer vision research.

Two classes of approach dominate the literature: EM-like alternation of closed-form solutions for the two factors of the matrix; and non-linear optimization of both factors simultaneously. As has been argued previously, nonlinear optimization based on a second-order approximation of the error function should be the faster approach. However alternation schemes remain enduringly popular.

I will show through theoretical and experimental arguments why this dicohotomy remains: it's not just that lazy programmers don't bother to implement second order methods, but that alternation schemes often do well on "easy" problems, and that many problems are "easy". Secondly, one must take care that the second-order methods do not throw away the nice properties of the alternation schemes. We introduce a class of hybrid algorithms which combine elements of both strategies, and show that these algorithms reach the correct minimum more frequently and at reasonable speed.

Finally, I show that for some well-known structure-from-motion data, the global optimum of the factorization problem does not, in fact, yield a realistic reconstruction. To achieve a realistic result, Bayesian priors and robust error kernels must be introduced. This is pretty obvious; what is less obvious is that this further cements the advantage of the direct optimization approaches.