

As-Rigid-As-Possible Image Registration for Hand-drawn Cartoon Animations

Daniel Sýkora*
Trinity College Dublin

John Dingliana
Trinity College Dublin

Steven Collins
Trinity College Dublin

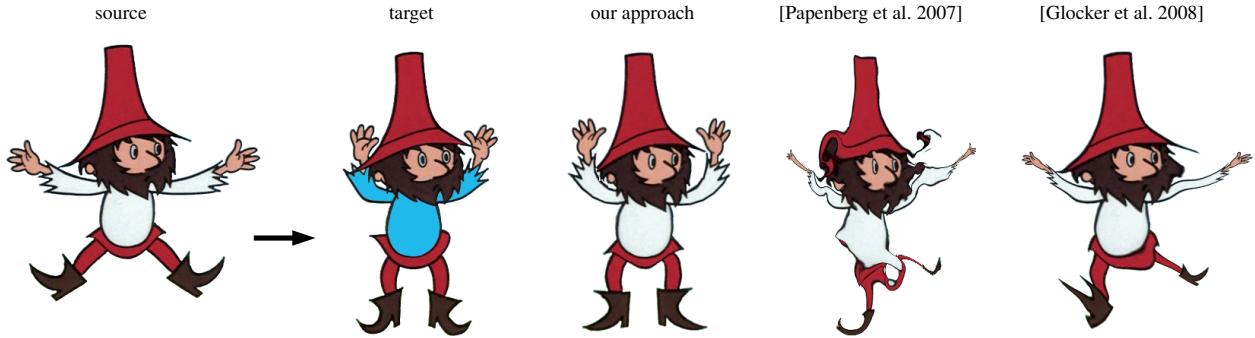


Figure 1: Compared with the state-of-the-art in deformable image registration, our novel approach reaches plausible results even for challenging configurations undergoing large amounts of free-form deformation and notable changes in appearance.

Abstract

We present a new approach to deformable image registration suitable for articulated images such as hand-drawn cartoon characters and human postures. For such type of data state-of-the-art techniques typically yield undesirable results. We propose a novel geometrically motivated iterative scheme where point movements are decoupled from shape consistency. By combining locally optimal block matching with as-rigid-as-possible shape regularization, our algorithm allows us to register images undergoing large free-form deformations and appearance variations. We demonstrate its practical usability in various challenging tasks performed in the cartoon animation production pipeline including unsupervised inbetweening, example-based shape deformation, auto-painting, editing, and motion retargeting.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Registration; I.3.4 [Computer Graphics]: Graphics Utilities—Graphics editors; J.5 [Computer Applications]: Arts and Humanities—Fine arts

Keywords: deformable image registration, as-rigid-as-possible deformation, interactive shape manipulation

1 Introduction

In a traditional cartoon animation each animation frame is drawn by hand so that the correspondences between them are unknown. Such a drawback considerably limits the usage of traditional approaches in recent computer-based animation systems, where the knowledge of correspondences between individual key-frames plays an important role.

Obtaining correspondences automatically is a challenging task since each hand-drawn image is unique and typically undergoes a large amount of free-form deformation and notable change in appearance. In this context popular computer vision techniques based on local similarity [Lowe 2004] or global contexts [Belongie et al. 2002] fail since they rely on unique local features or stable global configurations. Although such features are typical for real world photographs they are rare in hand-made drawings. Moreover, the aforementioned techniques provide only isolated point correspondences and do not consider global consistency. Thus, they can easily lead to spatially inconsistent mapping.

A more powerful approach to this problem is deformable image registration [Maintz and Viergever 1998; Modersitzki 2004; Gholipour et al. 2007] which allows the retrieval of dense correspondences between images and simultaneously maintains spatial consistency of the resulting mapping. It is typically formulated as a non-linear optimization problem where a predefined energy function is minimized through some established numerical optimization technique [Klein et al. 2007]. However, there are two key difficulties which make the solution challenging: (1) non-convexity of the energy function and (2) sensitivity to outliers (i.e. appearance variations that do not fit the selected deformation model). To overcome these, an initial guess close to the global minima is required. It can be obtained through various heuristics such as the popular multi-scale approach [Lucas and Kanade 1981] or by using a hierarchy of deformation models [Bergen et al. 1992]. Unfortunately, for large displacements or appearance variations, even these heuristics yield erroneous results. This fundamental problem has been recently addressed by techniques that attempt to minimize the energy not through the iterative numerical optimization, but directly via discrete labelling [Glocker et al. 2008; Shekhovtsov et al. 2008]. These are built upon recent advances in algorithms for inference from random fields [Szeliski et al. 2008] which allow fast approximate solutions to non-linear problems with effective avoidance of local minima. Nevertheless, they still do not guarantee global optimum (since the problem is NP-hard) and become computationally intractable for large displacements due to significantly increasing number of labels.

In this paper we develop a new approach to deformable image registration which addresses the issue of local minima and is able to reach a desired pose even from large initial displacements or no-

*e-mail: sykorad@cs.tcd.ie

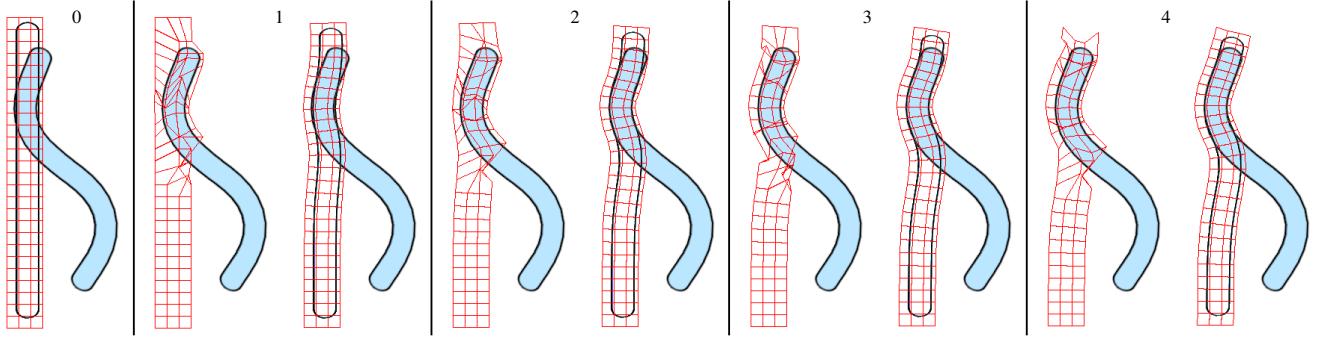


Figure 2: Deformable image registration in progress – we want to register a straight stripe (filled with transparent color and embedded in a square lattice) with its S-shaped counterpart filled with light blue color (column 0). In each iteration (columns 1–4) two steps are repeated: points are first pushed towards locations with minimal visual difference without considering shape consistency (left) and then the shape is regularized using a variant of as-rigid-as-possible shape matching algorithm (right). Note, how the shape gradually approaches the desired configuration.

table changes in appearance. It is inspired by the success of recent work in real-time simulation of deformable objects [Müller et al. 2005; Rivers and James 2007], where points are pushed directly towards desired positions and then as-rigid-as-possible shape regularization is used to ensure consistency of the original shape. In our case, points are not influenced by gravity or inertial forces, but instead attracted to locations with minimal visual difference. A key benefit of this new approach lies in the fact that the aforementioned shifts can be arbitrary and only the shape regularization ensures consistency. This is the fundamental difference to numerical optimization where incremental shifts are predicted by minimizing locally linearized version of the energy function, which typically leads to an inappropriate local minima. Our new technique is closer to the approaches based on discrete optimization in the sense it can recover from inappropriate local minima and lead to more plausible results. However, the key difference is that we do not minimize an energy function but instead use a geometrically motivated shape regularization scheme which preserves local rigidity and does not require computationally demanding discrete optimization inapplicable to large initial displacements.

Since our work is mainly motivated by the needs of the cartoon animation production pipeline, we demonstrate the practical usability of our new algorithm in this context. We show how it can reduce the amount of manual work in tasks such as inbetweening, painting, retargeting, shape deformation, and reusing traditional animation. We believe that these examples demonstrate the practical potential of our new technique and will motivate developers of recent professional cartoon animation systems to incorporate our technique as a versatile building block applicable to various practical scenarios.

The rest of the paper is organized as follows. First we briefly overview related work and analyze key drawbacks which motivated us to develop a new approach. Then we describe the proposed algorithm in more detail and discuss its strengths and limitations. Finally we demonstrate its practical usability in the context of the cartoon animation production pipeline and conclude with several ideas for future work.

2 Related work

Obtaining correspondences between hand-drawn images is a challenging task that has captured the attention of many researchers within the last two decades. Xie [1995] used simple affine transformations to match two line drawings and perform automatic inbetweening. Madeira et al. [1996] pioneered a region-centered ap-

proach where drawings are first sub-divided into regions and then matched using shape similarity and topological relations. This approach has been later improved by several authors who assume additional semantic information about the image [Kort 2002], cater specifically for black-and-white cartoons [Sýkora et al. 2005], rely on a hierarchy of regions [Qiu et al. 2005] or employ skeleton matching [Qiu et al. 2008]. Their common limitation is that they are applicable only to specific easy-to-analyze drawing styles and do not provide dense correspondences.

Bregler et al. [2002] presented a more general approach in their famous framework for cartoon motion capture. They overcome the problem of deformable registration by sampling the space of possible deformations using as-rigid-as-possible interpolation [Alexa et al. 2000] and then infer optimal linear combinations of these samples to fit the target pose. Although this approach works in the context of motion retargeting, it is not applicable to our problem since it does not directly provide dense correspondences between two animation frames. De Juan and Bodenheimer [2006] utilize dense correspondences in their framework for re-using traditional animation. However, they completely rely on manual initialization and refine dense mapping using an existing algorithm [Wirtz et al. 2004] based on numerical optimization. Recently, Xu et al. [2008] proposed a system for animating motion from several snapshots captured in a single image. However, since they rely on shape contexts [Belongie et al. 2002], unstable under large free-form deformations, extensive manual intervention is needed to identify stable features.

3 Our approach

Our novel approach to deformable image registration stems from the successful workflow recently used for dynamic simulation of deformable objects by Müller et al. [2005] and later extended by Rivers and James [2007]. A core idea of this technique is to decouple point movements from shape consistency so that the physical simulation can be applied directly on points, treating them as particles without considering their mutual connectivity. To keep the shape consistent, a geometrically motivated shape matching phase is then used to regularize point locations.

A key observation is that such a workflow can bring significant benefit also to deformable image registration since it allows retrieval of locally optimal shifts and still keeps the shape consistent. This is in contrast to energy-based approaches where shifts are limited by the deformation model which does not allow temporary increase of the overall energy and so typically leads to undesirable local minima.

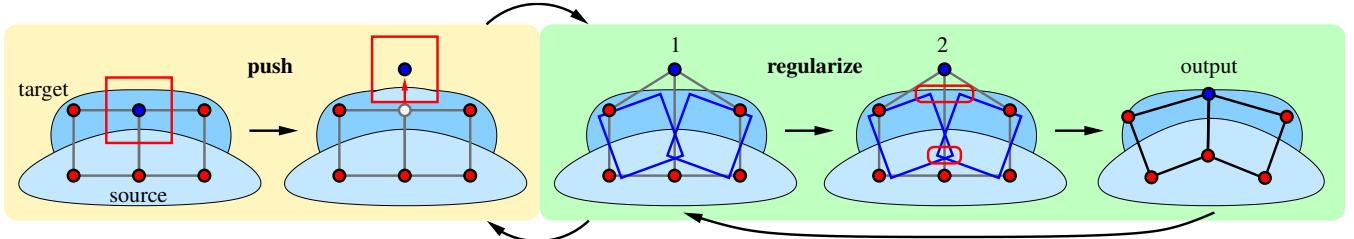


Figure 3: A schematic overview of the proposed algorithm – the aim is to register a light blob (source) with its darker counterpart (target). The light blob is embedded in a square lattice (partly visible). The algorithm iterates two main phases: **push** (yellow part) & **regularize** (green part). In the pushing phase block matching is used to move lattice points towards locations where a sum of absolute differences over a local neighborhood (red square) is minimal. Then in the shape regularization phase two steps are iterated: (1) optimal rigid transformation is computed for each lattice square and then (2) lattice points are moved to the centroid of their shared instances in all connected squares.

The only necessary modification as compared to the original concept is to replace physically motivated forces by an attraction to locations with visually similar neighborhood. Based on this setup, the resulting algorithm is as follows. Similarly to [Rivers and James 2007] we embed the input image into a regular square lattice respecting its articulated shape and then iterate two following steps until a stable configuration is reached:

1. **Push** all points to locations with minimal visual difference (Section 3.1).
2. **Regularize** the point locations to keep the shape consistent (Section 3.2).

These steps are illustrated in Figure 3 and a practical example is presented in Figure 2. Note how in each iteration, points are first pushed arbitrarily towards desired locations and how the overall shape becomes messy. Nevertheless, after regularization the shape is consistent and better fits the target pose.

In the following sections we describe these two steps in more detail, discuss implementation issues, stopping criteria, and possible limitations.

3.1 Push

The aim of the pushing phase is to find a new location for each point on the embedding lattice that minimizes visual difference in its local neighborhood. Since we are not limited by shape consistency we can utilize simple block matching which guarantees globally optimal shift within a predefined search area (see Figure 3, yellow part). Formally, the aim is to find a shift vector \mathbf{t}^* from search area M that minimizes the sum of absolute differences over a neighborhood N , i.e.:

$$\mathbf{t}^* = \arg \min_{\mathbf{t} \in M} \sum_{\mathbf{p} \in N} |\mathbf{S}(\mathbf{p} + \mathbf{t}) - \mathbf{T}(\mathbf{p})| \quad (1)$$

where \mathbf{S} denotes the source and \mathbf{T} the target image. Note that in spite of shift optimization, the overall algorithm is not limited to pure translation, since \mathbf{S} is slightly deformed in each iteration local neighborhoods of points gradually adapt to more complicated deformations.

The important parameters of the block matching phase are the size of the neighborhood $|N|$ and the size of search area $|M|$. In general $|N|$ should be large enough to contain substantial information but also small enough to preserve locality, whereas $|M|$ should allow attraction to further locations but also avoid ambiguity. The other limiting factor is the computational overhead which can increase dramatically due to the worst case complexity of the block

matching algorithm $\mathcal{O}(|N||M|)$. If we consider that in each iteration typically hundreds of block matching operations are executed, the complexity can be very high even for small neighborhoods and search areas. However, due to the fact that optimal block positions typically remain constant during most iterations and all other shifts have much higher sums of absolute differences, the early termination heuristic [Li and Salari 1995] can be used to gain considerable speed up. Based on this observation, we set the width of N to 16 pixels and the width of M to 48 pixels (providing that images are in PAL resolution). This setting yields good results both in robustness and computational overhead in all examples shown in this paper.

3.2 Regularize

The second step of our algorithm is a geometrically motivated routine that iteratively regularizes the point locations so that local rigidity of the shape is preserved. This is another important difference from state-of-the-art techniques, which typically use elastic models that do not preserve rigidity and produce undesirable deformations when the initial displacements are large or when there is a notable variation in appearance (see Figure 1).

In Müller's original algorithm, the aim was to find an optimal rigid transformation (rotation \mathbf{R}^* and translation \mathbf{t}^*) that moves points of the original shape $\mathbf{p}_i \in P$ so that the sum of squared distances to the desired pose \mathbf{q}_i is minimized:

$$(\mathbf{R}^*, \mathbf{t}^*) = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i |\mathbf{R} \cdot \mathbf{p}_i + \mathbf{t} - \mathbf{q}_i|^2 \quad (2)$$

In 3D the computation of \mathbf{R}^* is non-linear, thus polar decomposition is required to solve this problem. However, as shown by Schaeffer et al. [2006], a simple closed form solution exists in 2D. It can be obtained when we compute centroids \mathbf{p}_c and \mathbf{q}_c of the source and target pose and then substitute $\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p}_c$ and $\hat{\mathbf{q}}_i = \mathbf{q}_i - \mathbf{q}_c$:

$$\mathbf{R}^* = \frac{1}{\mu} \sum_i \begin{pmatrix} \hat{\mathbf{p}}_i \\ \hat{\mathbf{p}}_i^\perp \end{pmatrix} \left(\begin{array}{cc} \hat{\mathbf{q}}_i^T & \hat{\mathbf{q}}_i^{\perp T} \end{array} \right), \quad (3)$$

where

$$\mu = \sqrt{\left(\sum_i \hat{\mathbf{q}}_i \hat{\mathbf{p}}_i^T \right)^2 + \left(\sum_i \hat{\mathbf{q}}_i \hat{\mathbf{p}}_i^{\perp T} \right)^2}, \quad (4)$$

T denotes transposition, and the operator \perp denotes the perpendicular vector, i.e.: $(x, y)^\perp = (y, -x)$. Once the rotation matrix \mathbf{R}^* is known, the translation vector \mathbf{t}^* can be computed directly:

$$\mathbf{t}^* = \mathbf{p}_c - \mathbf{R}^* \cdot \mathbf{q}_c \quad (5)$$

Our embedding lattice consists of several connected squares. In this case local rigid transformations are computed individually for each square and then the global smoothing step is used to ensure consistency. This simple extension enables more flexible deformations and still preserves local rigidity of the original shape (see Figure 4).

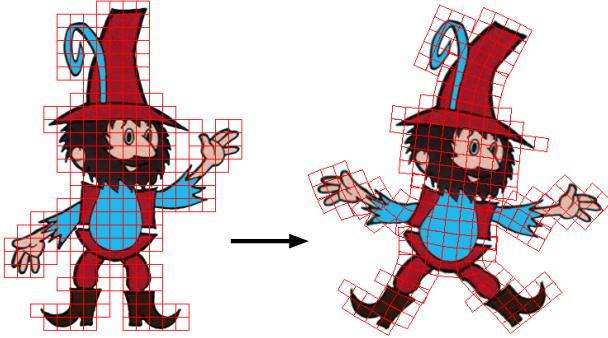


Figure 4: An example of as-rigid-as-possible image deformation – the original image embedded in a square lattice (left) and its deformed counterpart (right).

A similar mechanism is also used in the context of interactive shape deformation [Igarashi et al. 2005; Sumner et al. 2007; Botsch et al. 2007; Sorkine and Alexa 2007], however, the key difference is that in these techniques user-specified point locations are treated as hard constraints and the aim is to find an optimal deformation to satisfy them. In our case we do not have hard constraints. What we want is to smooth out point locations so that the shape becomes consistent. To perform this smoothing we exploit a very simple iterative approach inspired by recent work in interactive shape deformation [Wang et al. 2008]. It produces similar results to [Rivers and James 2007] but allows smooth control over shape rigidity (see Figure 3, green part):

1. For each square on the embedding lattice, use equations (3), (4) and (5) to obtain $(\mathbf{R}^*, \mathbf{t}^*)$ and use this to transform its points.
2. Move each point on the embedding lattice to the centroid of its transformed instances in all connected squares.

The only difference to the original shape deformation technique is that Wang et al. additionally simulate hard constraints by setting very large weights to points that represent manipulation handles. In fact their algorithm is nearly identical to [Sorkine and Alexa 2007], where instead of computing centroids a sparse linear system is solved. This modification clarifies the aforementioned difference between shape regularization and interactive shape deformation. In our case no points are fixed therefore after sufficient number of iterations the shape will return to the original configuration up to some global rigid body transformation. Such behavior is depicted in Figure 5 where one point is fixed at a different location and then the evolution of the deformation is captured during several shape regularization iterations. Initially the shape is flexible but with an increasing number of iterations, global rigidity is enforced so that the deformation gradually reduces to pure translation. This is caused by the diffusive nature of the averaging phase which gradually propagates rigidity through the whole shape.

This gradual diffusion of rigidity has several practical applications. By changing the number of shape regularization iterations we can smoothly vary between rigid and flexible deformation. It allows us to implement a smooth analogy to a hierarchy of deformation models [Bergen et al. 1992] (see Section 3.4) and also one-point interactive shape deformation (see Section 4).

3.3 Stopping criteria

Although our method does not explicitly minimize predefined energy function we can still estimate plausibility of the resulting registration by computing the average sum of absolute differences over all blocks during the block matching phase. In Figure 9 there are several graph plots of this average (blue curve) measured during a hundred iterations for different registration tasks. In most cases this value decreases monotonically and after several iterations the change is negligible. However, when a part of the shape undergoes a large non-overlapping deformation the change can be negligible for several iterations (see Figure 9b) and after that period the algorithm suddenly approaches new configurations with much lower value. This is caused by the fact that although a part of the image moves towards the desired pose it still remains in an area with no overlap where the sum of absolute differences is nearly constant. To overcome such ambiguity we instead monitor the average distance to the initial rest pose (red curve in Figure 9):

$$d_{\text{avg}} = \frac{1}{|P|} \sum_i \|\mathbf{p}_i - \mathbf{q}_i\| \quad (6)$$

This value informs us whether the control points on the embedding lattice are moving or not. We stop push-regularize iterations when d_{avg} has not changed considerably in the last 20 iterations.

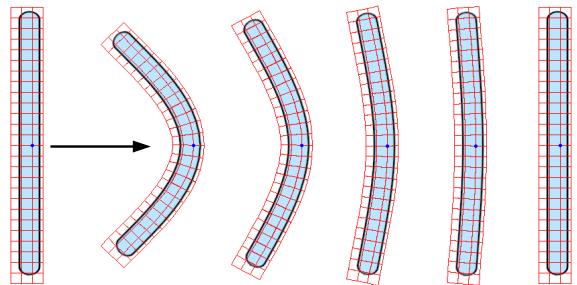


Figure 5: The evolution of the shape deformation through several shape regularization iterations – one point is fixed at different location (left). During the first iterations the shape is flexible but when the number of iterations increases the deformation gradually approaches pure translation (from left to right).

3.4 Limitations

Although our approach produces good results in most practical scenarios, there are some limiting factors which should be taken into account since they can lead to unexpected behavior. In this section we discuss these in more detail and address how they can be dealt with.

Limited resolution. Since we embed the image into a coarse lattice we cannot directly obtain pixel or even sub-pixel precision. Although a multi-scale extension is possible, increasing the number of squares makes the overall iterative process ineffective. This is caused mainly by an increasing number of block matching instances and shape regularization iterations. However, the coarse approximation produced by our algorithm is typically close enough to the desired pose so that classical energy-based approaches can be utilized to refine the registration to sub-pixel precision (we use a publicly available implementation of [Glocker et al. 2008]).

Occlusion and topology. The presence of occluders and topology variations in 2D projections of 3D articulated objects is a long-standing and challenging computer vision problem. It also limits

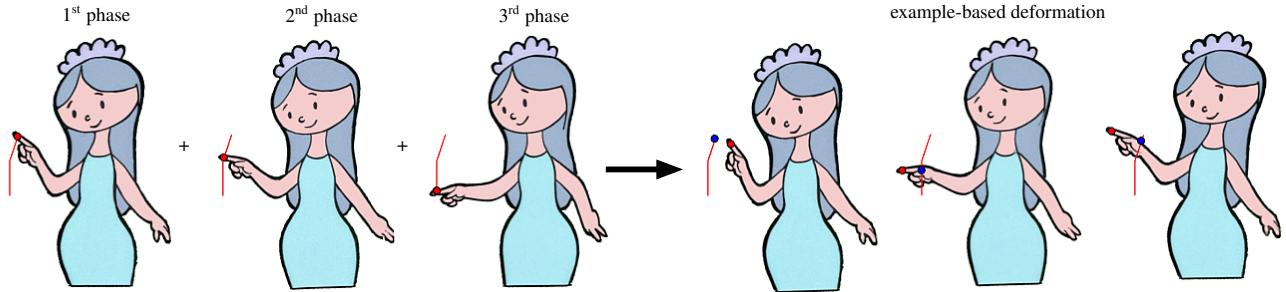


Figure 6: Example-based shape deformation – by registering several consecutive animation phases (left) a smooth sequence of intermediate frames can be generated. This can be utilized for a synthesis of new poses satisfying a user-given positional constraint (right): the current position of the dragged point (red dot) is projected (blue dot) on its key-frame trajectory (red curve) to retrieve the corresponding intermediate frame which is subsequently deformed to match the actual position of the dragged point.

the usage of our algorithm since occluded parts move together with their occluders and topology variations impose false connectivity. A possible solution to this problem is to reconstruct a layered representation of the image where each layer has its own depth information and shares common control points with other layers. Using this structure one can perform the pushing step only between layers having equal depth and then use the shape regularization phase to propagate these movements to other connected layers.

Scaling and shearing. As our method exploits the as-rigid-as-possible deformation model it is not able to handle deformations which do not preserve local rigidity (such as scaling or 3D rotation). This limitation can be partially reduced by exploiting an approach analogous to the use of a hierarchy of deformation models. Initially we can treat the image as more rigid and perform a higher number of rigid shape matching iterations. After that the number is gradually decreased to allow more flexible deformations. However, even using this extension, significant changes in scale and/or shearing are still intractable. In cases when such deformations are required we can switch to a different local deformation model allowing similarity or even affine transformations. According to [Schaefer et al. 2006] for similarity this can be done by replacing (4) with:

$$\mu = \sum_i \hat{\mathbf{p}}_i \hat{\mathbf{p}}_i^T \quad (7)$$

and for affine transform by replacing (3) and (5) with a full affine matrix:

$$\mathbf{A} = \left(\sum_i \hat{\mathbf{p}}_i^T \hat{\mathbf{p}}_i \right)^{-1} \sum_j \hat{\mathbf{p}}_j^T \hat{\mathbf{q}}_j \quad (8)$$

However, since similarity and affine models do not tend to preserve area they are not as stable as the original as-rigid-as-possible model therefore are suitable only for final refinement when the source and target pose are close enough, otherwise they can distort the image considerably and lead to unacceptable results.

Insufficient overlap. A key feature of our technique is the ability of the block matching phase to overcome configurations which correspond to local minima in energy-based techniques. However, to avoid matching ambiguity, the size of searching windows has to be limited. Because of this reason, our method requires partial overlap and consistent scale & orientation between source and target images. For images that do not satisfy these requirements we recommend that the initial rigid-body transformation be estimated by hand or that some automatic rigid-body registration technique should be used. When the insufficient overlap is caused by large free-form deformation (as in Figure 9h), the algorithm may get stuck in some inappropriate pose. In this case, the user can

provide additional hints by dragging a problematic part towards a desired position or use bidirectional registration, i.e. to alternate pushing and regularization steps on both source and target image. As compared to single image deformation where the target image is static, this approach provides better flexibility and so can overcome challenging configurations.

4 Results and Applications

We implemented our algorithm and tested it on various hand-drawn cartoon characters and human postures undergoing both small and large free-form deformations and changes in appearance. Selected results are presented in Figures 1 and 9. All examples are in PAL resolution. The width of squares on the embedding lattice is the same as the width of neighborhood N in equation (1), i.e. 16 pixels (blue squares in Figure 9) and the width of the search area M is 48 pixels (red squares). The number of inner iterations in the shape regularization phase is linearly decreased from 256 to 32 during the first 50 push-regularize steps.

To have an unified overview of the algorithm convergence we measured the average sum of absolute differences (blue curve) and the average distance to the starting pose (red curve) during the first 100 iterations for all examples in Figure 9. The actual number of iterations needed to reach stable configuration varies with the complexity of deformation. In simple cases it does not exceed 30, however, for large deformations such as human postures in Figures 9f and 9g it increases to 80. A typical processing speed is 20 iterations per second on a 3 GHz machine while the most demanding part is the block matching phase. However, it can be easily parallelized and thus much better processing speeds could be reached on some parallel architectures.

As stated in Section 3.4 the accuracy of our algorithm depends mainly on the resolution of the embedding lattice. Such precision is typically sufficient for applications where exact dense correspondences are not required such as auto-painting or motion capture. However, for inbetweening and example-based shape deformation, where smooth transitions are required, subsequent refinement is necessary to obtain sub-pixel accurate dense mapping. When a local appearance does not change considerably it is possible to take the result of our method as an initial guess for an energy-based approach (we use [Glockner et al. 2008]) and obtain refined sub-pixel accurate mapping. In Figure 9 we show both the registrations produced by our algorithm and also the corresponding refined results.

In the rest of this section we discuss several applications. Since our work is mainly motivated by the needs of the professional cartoon animation production pipeline we focus on this field. However, we believe that our algorithm is versatile enough to be applied

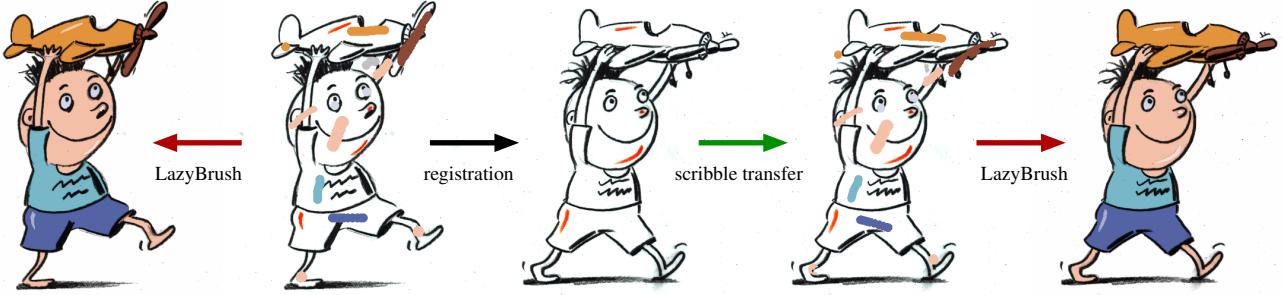


Figure 7: Auto-painting by unsupervised scribble transfer – color scribbles can be transferred from already painted to yet unpainted animation frames using our deformable image registration algorithm. The LazyBrush [Sýkora et al. 2009] algorithm is then utilized to compute the final painting.

in other contexts such as pedestrian registration, deformable object snapping or improving interactive shape deformation by providing dynamic feedback that looks like physical simulation.

Unsupervised inbetweening. The knowledge of dense correspondences between several consecutive frames allows us to create smooth intermediate transitions. One possibility for achieving this is to linearly interpolate positions of corresponding pixels. However, this is applicable only for small motions since the local rigidity is not preserved. A better solution is to divide the transition to coarse and fine level. The coarse level consists of the same square lattice as used for registration and the fine level is represented by two dense displacement maps (source-target and target-source) computed by the energy-based method [Glocker et al. 2008]. To generate the intermediate frame we first linearly interpolate the coarse lattices of the source and target frame and perform several shape regularization iterations to enforce rigidity. On the pixel level we scale transformed source-target and target-source displacements and resample source and target images accordingly. Finally we blend co-located pixels to obtain C^0 continuity.

Example-based shape deformation. User-driven shape deformation based on intuitive positional constraints has recently become popular particularly due to the work of Igarashi et al. [Igarashi et al. 2005]. Although many researchers attempt to improve this technique [Schaefer et al. 2006; Weng et al. 2006; Wang et al. 2008] they still offer only single image deformation. Thanks to our deformable image registration algorithm, we can easily extend this technique to multiple images (see Figure 6). By registering several animation phases we obtain smooth transitions as we do for inbetweening, however, a key difference here is that we allow the user to drag a specific point and move it to a different location. We project this new location on its inbetweening trajectory and generate a closest transition frame that is subsequently deformed to match the user-specified position. This enables interactive shape deformation which respects the original animation but is more flexible than simple inbetweening. Moreover, the ease of manipulation is improved considerably since in contrast to classical approaches we do not need to place other positional constraints to fix the global pose. Instead we apply a lower number of shape regularization iterations as described in Section 3.2 to suppress the diffusion of rigidity so that parts of the shape having long geodesic distances from the selected point remain untouched.

Auto-painting and editing. The process of adding colors to hand-drawn images is one of the most challenging tasks in the classical cartoon animation pipeline. In the last decade researchers have developed various auto-painting approaches allowing significant reduction of manual effort [Madeira et al. 1996; Chang and Lee 1997; Seah and Feng 2000; Sýkora et al. 2005; Qiu et al. 2008]. As these

techniques exploit similarity of regions they require drawing styles that can be easily converted to a set of homogenous regions. Recently, Sykora et al. [2009] introduced a more general approach based on color scribbles that is applicable to a broad class of different drawing styles. By registering painted and yet unpainted frames, we can transfer scribbles between animation frames and considerably speed up the process (see Figure 7). Besides painting, similar workflow can be utilized to perform various editing operations, e.g. retouching, insertion, and deletion.

Motion capture and retargeting. In this application pioneered by [Bregler et al. 2002] the aim is to transfer specific motion captured in a sequence of images to a novel animation having a different visual appearance. In our case this can be done by superimposing a skeleton on a reference pose and then using deformable image registration to find corresponding positions of joints and bones in subsequent animation frames (see Figure 8). Moreover, the superimposed skeleton can be utilized to form a set of rigid clusters and perform skeletal-like deformation via rigid square matching as in [Wang et al. 2008].

5 Conclusions and Future work

We presented a new approach to deformable image registration based on an approach analogous to the dynamic simulation of deformable objects. In contrast to previous techniques it handles large free-form deformations and notable changes in appearance. Although the algorithm prevails in challenging situations it is surprisingly easy to implement. We believe that, due to its simplicity and robustness, it will find numerous applications in tasks where the knowledge of correspondences between images plays an important role. As an example of such usage we presented several use cases in the context of the cartoon animation production pipeline.

As future work we plan to extend our approach to handle occlusions and also to develop efficient multi-resolution schemes to avoid dependence on energy-based techniques for applications where a pixel or sub-pixel precision is required.

Acknowledgements

We are grateful to Ladislav Kavan for numerous fruitful discussions. Thanks must also go to anonymous reviewers for their insightful comments and to Tomáš Jarkovský, Vojtěch Votýpka, and Tomáš Rychecký from AniFilm studio for being initiators of this work. Cartoon images used in this paper are courtesy of Pavel Koutský / AniFilm, and studios Universal Production Partners & Digital Media Production. This work has been supported by the Marie Curie action IEF, No. PIEF-GA-2008-221320.

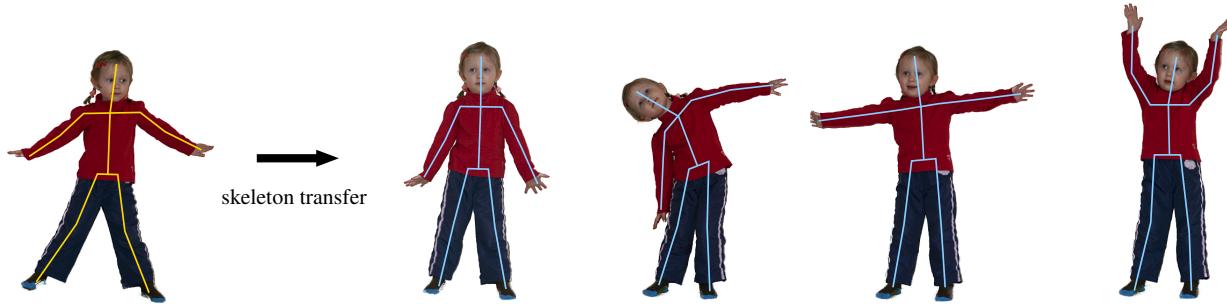


Figure 8: Motion capture by skeleton transfer – the image of the rest pose was manually annotated by a skeleton (left). Its corresponding positions on several new postures were obtained without user intervention using our deformable image registration algorithm (right).

References

- ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *ACM SIGGRAPH Conference Proceedings*, 157–164.
- BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 24, 509–522.
- BERGEN, J. R., ANANDAN, P., HANNA, K. J., AND HINGORANI, R. 1992. Hierarchical model-based motion estimation. In *Proceedings of European Conference on Computer Vision*, 237–252.
- BOTSCH, M., PAULY, M., WICKE, M., AND GROSS, M. H. 2007. Adaptive space deformations based on rigid cells. *Computer Graphics Forum* 26, 3, 339–347.
- BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: Motion capturing cartoons. *ACM Transactions on Graphics* 21, 3, 399–407.
- CHANG, C. W., AND LEE, S. Y. 1997. Automatic cel painting in computer-assisted cartoon production using similarity recognition. *The Journal of Visualization and Computer Animation* 8, 3, 165–185.
- GHOLIPOUR, A., KEHTARNAVAZ, N., BRIGGS, R. W., DEVous, M., AND GOPINATH, K. S. 2007. Brain functional localization: A survey of image registration techniques. *IEEE Transactions on Medical Imaging* 26, 4, 427–451.
- GLOCKER, B., KOMODAKIS, N., TZIRITAS, G., NAVAB, N., AND PARAGIOS, N. 2008. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis* 12, 6, 731–741.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* 24, 3, 1134–1141.
- DE JUAN, C. N., AND BODENHEIMER, B. 2006. Re-using traditional animation: methods for semi-automatic segmentation and inbetweening. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 223–232.
- KLEIN, S., STARING, M., AND PLUIM, J. P. W. 2007. Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Transactions on Image Processing* 16, 12, 2879–2890.
- KORT, A. 2002. Computer aided inbetweening. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*, 125–132.
- LI, W., AND SALARI, E. 1995. Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing* 4, 1, 105–107.
- LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 2, 91–110.
- LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, 674–679.
- MADEIRA, J. S., STORK, A., AND GROSS, M. H. 1996. An approach to computer-supported cartooning. *The Visual Computer* 12, 1, 1–17.
- MAINTZ, J. B. A., AND VIERGEVER, M. A. 1998. A survey of medical image registration. *Medical Image Analysis* 2, 1, 1–16.
- MODERSITZKI, J. 2004. *Numerical Methods for Image Registration*. Oxford University Press, UK.
- MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. *ACM Transactions on Graphics* 24, 3, 471–478.
- PAPENBERG, N., SCHUMACHER, H., HELDMANN, S., WIRTZ, S., BOMMERSHEIM, S., ENS, K., MODERSITZKI, J., AND FISCHER, B. 2007. A fast and flexible image registration toolbox. In *Bildverarbeitung für die Medizin*, 106–110.
- QIU, J., SEAH, H. S., TIAN, F., CHEN, Q., AND WU, Z. 2005. Enhanced auto coloring with hierarchical region matching. *Computer Animation and Virtual Worlds* 16, 3–4, 463–473.
- QIU, J., SEAH, H. S., TIAN, F., CHEN, Q., WU, Z., AND MELIKHOV, K. 2008. Auto coloring with enhanced character registration. *International Journal of Computer Games Technology*, 1, 2.
- RIVERS, A. R., AND JAMES, D. L. 2007. FastLSM: Fast lattice shape matching for robust real-time deformation. *ACM Transactions on Graphics* 26, 3, 82.
- SCHAFFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Transactions on Graphics* 25, 3, 533–540.
- SEAH, H. S., AND FENG, T. 2000. Computer-assisted coloring by matching line drawings. *The Visual Computer* 16, 5, 289–304.
- SHEKHOVTSOV, A., KOVTUN, I., AND HLAVÁČ, V. 2008. Efficient MRF deformation model for non-rigid image matching. *Computer Vision and Image Understanding* 112, 1, 91–99.
- SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 109–116.
- SUMNER, R. W., SCHMID, J., AND PAULY, M. 2007. Embedded deformation for shape manipulation. *ACM Transactions on Graphics* 26, 3, 80.
- SÝKORA, D., BURIÁNEK, J., AND ŽÁRA, J. 2005. Colorization of black-and-white cartoons. *Image and Vision Computing* 23, 9, 767–782.
- SÝKORA, D., DINGLIANA, J., AND COLLINS, S. 2009. LazyBrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum* 28, 2, 599–608.
- SZELISKI, R. S., ZABIH, R., SCHARSTEIN, D., VEKSLER, O., KOLMOGOROV, V., AGARWALA, A., TAPPEN, M., AND ROTHER, C. 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 6, 1068–1080.
- WANG, Y., XU, K., XIONG, Y., AND CHENG, Z.-Q. 2008. 2D shape deformation based on rigid square matching. *Computer Animation and Virtual Worlds* 19, 3–4, 411–420.
- WENG, Y., XU, W., WU, Y., ZHOU, K., AND GUO, B. 2006. 2D shape deformation using nonlinear least squares optimization. *The Visual Computer* 22, 9, 653–660.
- WIRTZ, S., FISCHER, G., MODERSITZKI, J., AND SCHMITT, O. 2004. Superfast elastic registration of histologic images of a whole rat brain for 3d reconstruction. In *Proceedings of the SPIE*, vol. 5370, 328–334.
- XIE, M. 1995. Feature matching and affine transformation for 2D cell animation. *The Visual Computer* 11, 8, 419–428.
- XU, X., WAN, L., LIU, X., WONG, T.-T., WANG, L., AND LEUNG, C.-S. 2008. Animating animal motion from still. *ACM Transactions on Graphics* 27, 5, 117.

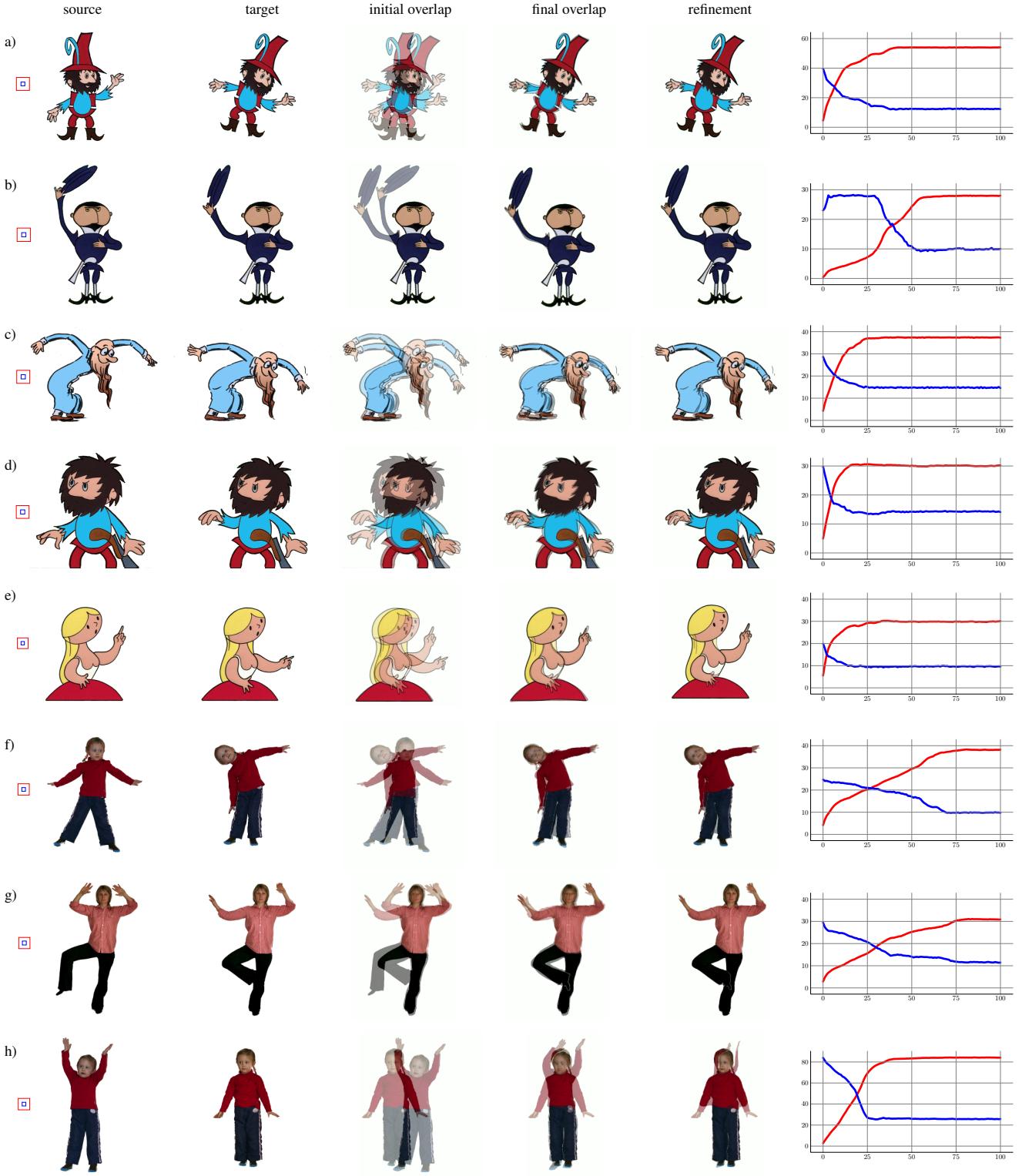


Figure 9: Selected examples of deformable image registration produced by our algorithm – each example contains (from left to right): size of the square on the embedding lattice equal to the local neighborhood N (blue square), size of the local searching area M (red square), source & target image, their initial overlap, resulting overlap after registration using our approach, overlap refined by energy-based approach [Glocker et al. 2008], and the graph of the average sum of absolute differences (blue curve) and the average distance to starting pose (red curve) for first 100 iterations. The last registration result (h) presents a failure example when our algorithm gets stuck in an undesirable pose due to very large free-form deformation.