

# Fast Musculoskeletal Registration Based on Shape Matching

Benjamin Gilles and Dinesh K. Pai

Department of Computer Science, University of British Columbia, Canada  
{bgilles,pai}@cs.ubc.ca

**Abstract.** This paper presents a new method for computing elastic and plastic deformations in the context of discrete deformable model-based registration. Internal forces are estimated by averaging local transforms between reference and current particle positions. Our technique can accommodate large non-linear deformations, and is unconditionally stable. Moreover, it is simple to implement and versatile. We show how to tune model stiffness and computational cost, which is important for efficient registration, and demonstrate our technique in the complex problem of inter-patient musculoskeletal registration.

## 1 Introduction

Registration is a fundamental problem in image and geometry processing. The problem of matching two surfaces, two images, or a surface and an image is often expressed as an energy minimization and is generally difficult to solve due to the presence of local solutions. To tackle this, researchers are focusing on finding, in specific contexts, adequate registration features, similarity measures, parameterizations of the deformation and optimization methods [1,2]. The goal is to evolve the solution (i.e. transformation) in a spatially coherent way through the introduction of priors, and along an as-convex-as-possible matching energy.

Musculoskeletal registration is a particularly challenging domain since it involves a large number of interrelated components undergoing large non-linear deformations with large anatomical variations in the population. In this context, allowable deformations need to be carefully parameterized to avoid falling into one of the numerous local solutions, *and* to present sufficient degrees of freedom. While the animation of articulated bodies has been widely explored based on skeleton-driven deformations, example-based techniques or physically-based methods [3], few works have been dedicated to musculoskeletal modeling and tracking from medical images [4,5]. However, as stressed in the survey of Blemker et al. [6], studies on (neuro)musculoskeletal disabilities could greatly benefit from using patient-specific volumetric models obtained from MR data.

The parameterization of elastic deformations from multiple global transforms has been successfully used in various domains such as shape editing [7], computer animation [8,9], computer vision [10] and image registration [11]. Spatially coherent (e.g. quasi-rigid) flow can be designed through generalized gradients

as shown by Charpiat et al. [10] (Eulerian setting) and Eckstein et al. [12] (Lagrangian setting). In [11], Arsigny et al. fuse locally rigid or affine transforms into a global transform by infinitesimally averaging their logarithms. This ensures invertibility and thus removes artifacts such as folding.

Our work extends a computer animation technique called *shape matching* introduced by Müller et al. [8] to efficiently approximate large soft-tissue elastic deformations. Non-rigid deformations are simulated by blending rigid transforms of overlapping mesh regions (clusters). But instead of lattice control points [9], our clusters directly contain object vertices. We show how to compute and optimize clusters, and prioritize particles so that the complexity can be tuned online. We compare two techniques for estimating rigid transforms and simulate plasticity through reference positions update. Like *generalized gradients* [10,12], this enforces spatially coherent deformations. We use shape matching as a regularization step that filters out noise from external forces, and thus propose a modified force-based evolution scheme. Finally we demonstrate and evaluate our technique in the complex problem of inter-patient musculoskeletal registration.

## 2 Methods

### 2.1 Background

Due to the large datasets and displacements involved in musculoskeletal imaging, our registration technique is based on discrete deformable models [13], that offer both flexibility and efficiency. The model, initially aligned to the segmented reference dataset is iteratively deformed until it matches the target image. At each time step, we trilinearly interpolate intensities at sampled locations along surface normals and perform, on one side, 1D profile registration, and on the other side, maximal gradient search (see [14,5] for more details). The two resulting goal positions, obtained within a certain search distance  $d$ , are averaged to compute the external image forces:  $\mathbf{f}^e(t) = \alpha_e(\bar{\mathbf{x}} - \mathbf{x}(t))$  where  $\alpha_e$  is the stiffness,  $\mathbf{x}(t)$  the current vertex positions and  $\bar{\mathbf{x}}$  the average goal position.

Because of noise, local solutions and aperture problems, the registration process need to be constrained through internal forces. The idea of shape matching deformation techniques is to compute the least squares rigid transform  $\mathbf{T}$  between  $\mathbf{x}(t)$  and the reference positions  $\mathbf{x}^r$  [8]. Applying the internal forces  $\mathbf{f}^i(t) = \alpha_i(\bar{\mathbf{x}} - \mathbf{x}(t)) = \alpha_i(\mathbf{T}\mathbf{x}^r - \mathbf{x}(t))$ , we introduce shape memory constraints that are rotation and translation invariant, which is a desired property (as in continuum mechanics, internal forces should be independent from object pose). Using initial positions as a reference ( $\mathbf{x}^r = \mathbf{x}(0)$ ) [8,9], shape matching forces act as elastic forces, proportional to the strain. When updating reference positions at each time step ( $\mathbf{x}^r = \mathbf{x}(t - dt)$ ), the effect, similarly to a plastic deformation, is to enforce spatially coherent deformations like generalized gradients [10,12]. We will show how to combine the two approaches in Section 3. To give some flexibility to the model, we do not apply this technique to the entire model but instead localize it to sets of vertices (clusters) around each vertex [9].

## 2.2 Preprocessing

**Clustering.** Because external forces are evaluated at model vertex positions, we use the model itself to compute deformations instead of an underlying regular lattice [9]. We define each cluster  $\zeta_i$  associated with a vertex  $i$ , as:  $\zeta_i = \{j : d(\mathbf{x}_i, \mathbf{x}_j) \leq s\}$  where  $s$  is the cluster size (or radius). Increasing cluster sizes makes the model more rigid. Here, we use Euclidean distances because objects are volumetric. However, in a different context, another measure could be used such as the geodesic distance.

**Cluster Optimization.** When we deal with large sizes  $s$ , index lists are very redundant across neighboring vertices. We exploit this redundancy to reduce computation and memory costs. Following [9], we only store the boolean difference in the index list between a vertex  $i$  and one of its neighbors  $k$  so that we have  $\zeta_i^+ = \{j : j \in \zeta_i - (\zeta_k \cap \zeta_i)\}$  and  $\zeta_i^- = \{j : j \in \zeta_k - (\zeta_k \cap \zeta_i)\}$ . Then, summation of some field data  $\mathbf{v}$ , within the cluster  $i$  can be quickly performed through:  $\Sigma_i(\mathbf{v}) = \sum_{j \in \zeta_i} \mathbf{v}_j = \Sigma_k(\mathbf{v}) + \sum_{j \in \zeta_i^+} \mathbf{v}_j - \sum_{j \in \zeta_i^-} \mathbf{v}_j$ . Unlike [9], our clusters contain irregularly sampled vertices, making list optimization unsystematic. Instead, we need an ordered list  $\mathbf{L}$  for updating sums so that  $\Sigma_k(\mathbf{v})$  is available for computing  $\Sigma_i(\mathbf{v})$  (the vertex  $k$  is a parent of  $i$ ). Our approach is to use the smallest non-empty clusters that are defined by  $\xi_i = \{j : d(\mathbf{x}_i, \mathbf{x}_j) \leq s_{min}\}$  with  $s_{min} = \max_i(\min_j d(\mathbf{x}_i, \mathbf{x}_j))$ . These *one-ring* clusters contain vertices sharing highly correlated clusters. The list is initialized with the vertex having the highest number of one-ring neighbors:  $L_0 = \operatorname{argmax}_i |\xi_i|$ . Then, all indexes in  $\xi_{L_0}$  are added to the list, with  $L_0$  as a parent. The propagation continues by pushing back indexes of  $\xi_{L_j} : 1 \leq j \leq |\xi_{L_0}|$  (with  $L_j$  as a parent), and so on. In some cases (e.g. separate models), new seeds need to be added, so that all indexes can finally be inserted into the list.

**Complexity Tuning.** To save computational time, we discard some vertices by removing them from the  $\zeta_i$ . We expect a limited loss of accuracy when clusters keep a sufficient number of vertices for estimating transformations. We propose to create a list  $\mathbf{S}$  containing indexes ordered from the less-significant to the high priority ones. A constraint is to always keep cluster cardinality over 3 ( $|\zeta_i| \geq 3$ ) to be able to estimate a rigid transform. To fill  $\mathbf{S}$ , our criterion is hence the minimum cardinality in each cluster:  $m_i = \min_{j \in \zeta_i} |\zeta_j|$  (so,  $i$  can be discarded if  $m_i > 3$ ). Starting with the ones of largest  $m_i$ , we successively insert vertices in  $\mathbf{S}$ , until  $m_i = 3, \forall i$ .

## 2.3 Registration

**Shape Matching Using Polar Decomposition.** For each cluster  $\zeta_i$ , the least square estimation of the rigid transform between particles weighted by their mass  $m_j$  is:  $\mathbf{T}_i = [\mathbf{R}_i, \mathbf{t}_i] = \operatorname{argmin} \sum_{j \in \zeta_i} m_j \|\mathbf{T} \mathbf{x}_j^r - \mathbf{x}_j\|^2$ . Several methods provide a closed-form computation of rigid transformations [15]. The most popular ones are based on quaternions, SVD decomposition and polar decomposition. For dynamic simulation, many of such computations are required at each time step, and

the polar decomposition has been preferred for its efficiency [8,9]. The solution of the polar decomposition approach is given by [8]:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{A}_i(\mathbf{A}_i^T \mathbf{A}_i)^{-1/2} = \mathbf{A}_i(\mathbf{V}_i \boldsymbol{\lambda}_i^{-1/2} \mathbf{V}_i^T) \quad \text{and} \quad \mathbf{t}_i = \mathbf{c}_i - \mathbf{R}_i \mathbf{c}_i^r \\ \mathbf{A}_i &= \boldsymbol{\Sigma}_i(m \mathbf{x} \mathbf{x}^T) - M_i \mathbf{c}_i \mathbf{c}_i^r{}^T = \boldsymbol{\Sigma}_i(m \mathbf{x} \mathbf{x}^T) - \boldsymbol{\Sigma}_i(m \mathbf{x}) \mathbf{c}_i^r{}^T \end{aligned}$$

Here,  $\mathbf{c}_i$  and  $\mathbf{c}_i^r$  denote cluster centers of mass in the current (resp. reference) configuration. This method is efficient since it involves the diagonalization  $\mathbf{A}_i^T \mathbf{A}_i = \mathbf{V}_i \boldsymbol{\lambda}_i \mathbf{V}_i^T$  that can be done using only a small number of cyclic Jacobi iterations. Following [9], the number of iterations is even more reduced by preconditioning the diagonalization with the previous rotations  $\mathbf{V}_i(t - dt)$ .

**Shape Matching Using Rigid Body Kinematics.** In the context of small rotational displacements ( $< 20^\circ$ ), the least square estimation can be linearized by  $\boldsymbol{\omega}_i^* \simeq \mathbf{R}_i - \mathbf{I}$  where  $\boldsymbol{\omega}_i^*$  is the cross-product matrix of the angular velocity. The solution is well known from rigid body mechanics:  $\boldsymbol{\omega}_i = \mathbf{I}_i^r{}^{-1} \mathbf{L}_i$  where  $\mathbf{I}_i^r$  is the inertia tensor and  $\mathbf{L}_i$  the angular momentum. Note that here, the displacements  $(\mathbf{x}_i - \mathbf{x}_i^r)$  act as instantaneous velocities. To reduce distortions due to non-orthonormal rotation matrices, we convert angular velocities into rotation matrices via the Rodrigues formula:  $\mathbf{R}_i = \mathbf{r}(\boldsymbol{\omega}_i)$ . All quantities can be quickly evaluated through summation within individual clusters:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{r}(\boldsymbol{\omega}_i) = \mathbf{r}(\mathbf{I}_i^r{}^{-1} \mathbf{L}_i) \quad \text{and} \quad \mathbf{t}_i = \mathbf{c}_i - \mathbf{R}_i \mathbf{c}_i^r \\ \mathbf{L}_i &= \boldsymbol{\Sigma}_i(m \mathbf{x}^r \times \mathbf{x}) - M_i \mathbf{c}_i^r \times \mathbf{c}_i = \boldsymbol{\Sigma}_i(m \mathbf{x}^r \times \mathbf{x}) - \mathbf{c}_i^r \times \boldsymbol{\Sigma}_i(m \mathbf{x}) \\ \mathbf{I}_i^r &= \text{tr}(\mathbf{C}_i^r) \mathbf{I} - \mathbf{C}_i^r \quad \text{and} \quad \mathbf{C}_i^r = \boldsymbol{\Sigma}_i(m \mathbf{x}^r \mathbf{x}^r{}^T) - M_i \mathbf{c}_i^r \mathbf{c}_i^r{}^T \end{aligned}$$

**Evolution.** Vertices receive different goal positions from the different clusters they belong to. So, we average these positions to produce smooth deformations (as in [9]) through the fast summation technique:  $\tilde{\mathbf{x}}_i = \boldsymbol{\Sigma}_i(\mathbf{T}) \mathbf{x}_i^r / |\zeta_i|$ . The effect of internal and external forces can now be combined to compute the new model state. Other forces could be added at this point as done in [5] to enforce, for instance, volume preservation, smoothing, damping, or non-penetration between models (e.g. when simulating sliding objects owning separate clusters). When summing all forces ( $\mathbf{f} = \mathbf{f}^i + \mathbf{f}^e$ ), the noise, which corrupts external forces, has a direct impact on model deformation. In this case, the condition  $\alpha_e \ll \alpha_i$  is needed to maintain model regularity. Instead, we prefer to apply shape matching as a *post regularization* step:  $\mathbf{f} = \mathbf{f}^i = (\boldsymbol{\Sigma}_i(\mathbf{T}) \mathbf{x}_i^r / |\zeta_i| - \mathbf{x})$  where transforms are estimated between the reference positions  $\mathbf{x}^r$  and the current positions *augmented by the external forces*  $(\mathbf{x} + \mathbf{f}^e)$ . So instead of adding forces that penalize undesired particle configurations, we project external forces onto the shape matching deformation space to filter out the noise. In practice, this has a stabilizing effect and removes constraints related to force stiffness. When a dynamic aspect is desired, the computation of the new positions from  $\mathbf{f}$  can be performed using classical integration schemes (such as the implicit Euler scheme as in [5]). In this paper, as we want to demonstrate the effect of shape

matching forces, we do not use velocities to extrapolate results. We rather apply the simplest relaxation scheme (gradient descent with unitary time step):  $\mathbf{x}(t + dt) = \mathbf{f} + \mathbf{x}(t)$ . To summarize, an iteration of our registration process involves:

- Compute external forces  $\mathbf{f}^e$  (including any other custom forces)
- Compute shape matching forces  $\mathbf{f}^i$ :
  - for each cluster  $\zeta_i$ , compute  $\mathbf{T}_i = \operatorname{argmin} \sum_{j \in \zeta_i} \|\mathbf{T}\mathbf{x}_j^r - \mathbf{x}_j - \mathbf{f}_j^e\|^2$
  - average goal positions  $\tilde{\mathbf{x}}_i = \boldsymbol{\Sigma}_i(\mathbf{T})\mathbf{x}_i^r/|\zeta_i|$  for each vertex  $i$
- Evolve particle positions:  $\mathbf{x} = \tilde{\mathbf{x}}$
- (Optionally, update reference positions to simulate plasticity:  $\mathbf{x}^r = \mathbf{x}$ )

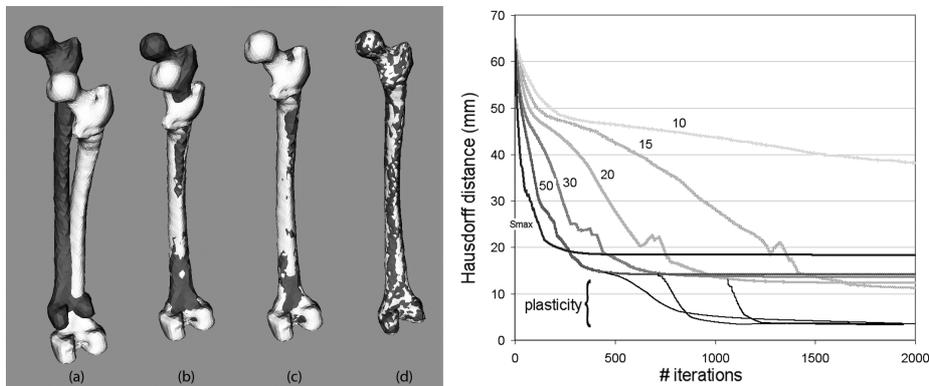
### 3 Results

**Implementation.** The implementation has been done in C, using parallel techniques when possible. Despite its sequential nature, the fast summation method could be mostly parallelized (the part involving  $\zeta^+$  and  $\zeta^-$ ). All timings have been measured on a 2.66Ghz QuadCore machine, and exclude visualization time.

**Performance Evaluation.** Our first experiments deal with computational speed measurement. Our test model is made of 24 surfaces (3 bones, 21 muscles) consisting of 23488 clusters/ vertices in total (see Fig. 2). For a cluster size of  $s = 3cm$  ( $s_{min} = 0.7cm$ ,  $s_{max} = 63cm$ ), its preprocessing lasts about  $3min$  for which most of the time is spent on scalability list computation ( $2min$ ). Cluster optimization reduces the number of indexes to store by more than four times (from 390 to 90 in average). In Fig. 2, we evaluate our complexity tuning method and compare the two shape matching techniques. The *rigid body* method is about 30% faster than the *preconditioned polar decomposition*. It was expected since the later involves multiple Jacobi iterations in addition to the computation of  $\mathbf{A}_i$  (equivalent complexity than  $\mathbf{L}_i$ ). The computational time decreases linearly with the augmentation of ignored vertices but does not go down to 0, since a minimal amount of time is spent on handling ignored vertices for which sums still need to be computed. However, time for evaluating other forces is strictly proportional to the number of active vertices. Because external forces are the main burden, discarding vertices significantly improves computational speed. In this example, intensity profiles are made of 30 values sampled every  $0.5mm$  and forces are evaluated in the range  $d = 1cm$  (20 samples).

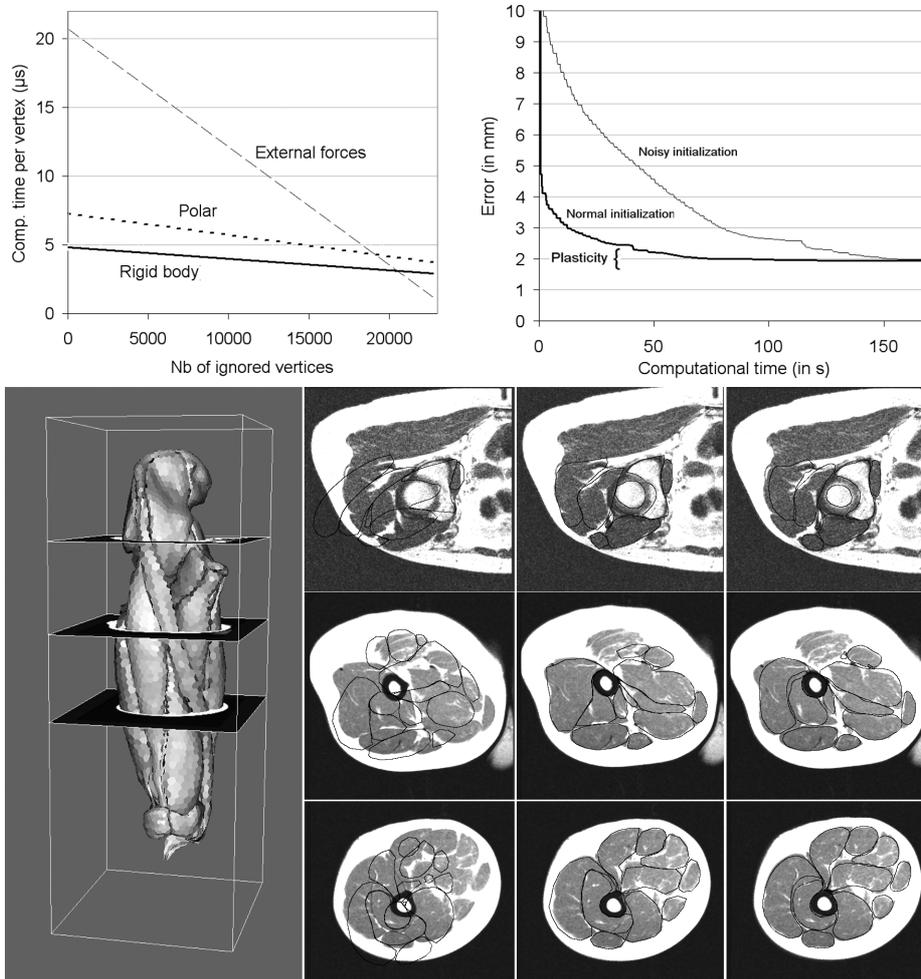
**Geometric Registration.** We now demonstrate our method in the context of surface registration through a representative example (two patient-specific femurs). Here, external forces are computed by projecting source vertices onto the target surface. The goal is to evaluate the required flexibility for covering anatomical variability while being robust against local solutions. As expected, model variations are limited by internal forces, making the alignment uncomplete (see Fig. 1.c). A higher rigidity produces a faster matching, due to a smaller number of local minima and to a steeper energy, but larger errors after convergence

(the deformation space is too small). To release some constraints, one could progressively decrease the rigidity (coarse to fine approach). But, using many rigidity levels is unpractical for large models (due to memory and preprocessing time charges). Our approach is to simulate plasticity by updating reference positions, when convergence has been reached (i.e. deformations  $< 50\mu m$ ). Here, we use the *rigid body* method which is theoretically *not* rotational invariant. But since reference positions are updated from time to time (i.e. due to plasticity) with small rotational changes, strict invariance is not mandatory. We did not notice significant differences from using the polar decomposition.



**Fig. 1.** Left: example of geometric registration with  $s = 5cm$ , changes between (c) and (d) illustrate plastic deformations. Right: Hausdorff distance in function of time for different cluster sizes/ rigidities, with and without plasticity.

**Iconic Registration.** We perform medical image registration using the *rigid body* method. Our data consists of four MRI volumes of the right hip and thigh (Axial T1-weighted spin echo sequence, matrix:  $512 \times 512 \times 150$ , in plane resolution:  $0.8 \times 0.8mm$ , slice thickness:  $2mm$  to  $1cm$ ) where the 24 surfaces have been interactively segmented [5]. Models from one subject are automatically aligned to the other datasets and segmentation error is estimated by measuring the average distance with the interactively segmented surfaces. A coarse initialization is performed using 15 landmarks and thin-plate-spline interpolation. External forces (stiffness  $\alpha_e = 0.8$ ) are based on intensity profile registration (normalized cross correlation similarity measure, 30 samples, decreasing search distance from  $1cm$  to 0) and gradient magnitude maximization (search distance from  $0.5cm$  to 0). The number of active vertices is progressively increased to 100%, and cluster sizes maintained at  $s = 3cm$ . For the three registrations, the average segmentation error was  $2.0mm$  ( $0.5mm$  more than [5], but the difference was not noticeable visually) and the computational time  $2min$  (compared to  $15min$ ). Our method was found to be relatively robust against parameter changes (for stiffnesses  $< 1$ , search distances  $< 2cm$ , and nb. of active vertices  $> 20\%$ ). We evaluated our method against large initialization defects (random perturbations



**Fig. 2.** Top: Complexity tuning and segmentation error evolution; Bottom: Registered 3D model and segmentation results on different slices (Columns: Noisy initialization/ Automatic segmentation/ Interactive segmentation)

in landmark locations between  $\pm 2cm$ ) and noticed almost no differences in the results (see Fig. 2). There was also no large fall into local solutions (except for the “floating” distal tendons that should have been driven by the tibia).

#### 4 Discussion and Future Work

We have shown that registration processes could be valuably constrained through fast deformation methods commonly used in computer graphics, especially when large displacements and models are involved. Our method does not make any assumption about the input geometry and mesh connectivity, so models from

various sources could be directly used (e.g. CAD, computer animation, finite-element models). Simulating plasticity is relevant for updating the reference model, but unrealistic deviations can be accumulated in presence of unreliable image forces. In future, we believe that allowable deformations could be limited by selecting reference clusters from different subjects in the population.

**Acknowledgments.** We would like to thank Qi Wei and Danny Kaufman for their helpful comments, and the University of Geneva for providing the data. This work is funded by the Swiss National Research Foundation.

## References

1. Maintz, J., Viergever, M.: A survey of medical registration. *MIA* 2(1), 1–36 (1998)
2. Audette, M., Ferrie, F., Peters, T.: An algorithmic overview of surface registration techniques for medical imaging. *MIA* 4, 201–217 (2000)
3. Teran, J., Sifakis, E., Blemker, S., Hing, V.N.T., Lau, C., Fedkiw, R.: Creating and simulating skeletal muscle from the visible human dataset. *IEEE TVCG* 11, 317–328 (2005)
4. Papademetris, X., Dione, D., Dobrucki, L., Staib, L., Sinusas, A.: Articulated rigid registration for serial lower-limb mouse imaging. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3750, pp. 919–926. Springer, Heidelberg (2005)
5. Gilles, B., Moccozet, L., Magnenat-Thalmann, N.: Anatomical modelling of the musculoskeletal system from MRI. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) *MICCAI 2006*. LNCS, vol. 4190, pp. 289–296. Springer, Heidelberg (2006)
6. Blemker, S., Asakawa, D., Gold, G., Delp, S.: Image-based musculoskeletal modeling: Applications, advances, and future opportunities. *J. of Magnetic Resonance Imaging* 25(2), 441–451 (2007)
7. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. *IEEE TVCG* 14, 213–230 (2008)
8. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. *SIGGRAPH*, 471–478 (2005)
9. Rivers, A., James, D.: Fastlsm: Fast lattice shape matching for robust real-time deformation. *SIGGRAPH* 26 (2007)
10. Charpiat, G., Maurel, P., Pons, J., Keriven, R., Faugeras, O.: Generalized gradients: Priors on minimization flows. *Int. J. of Computer Vision* 73, 325–344 (2007)
11. Ayache, N., Pennec, X., Arsigny, V., Commowick, O.: A log-Euclidean polyaffine framework for locally rigid or affine registration. In: Pluim, J.P.W., Likar, B., Geritsen, F.A. (eds.) *WBIR 2006*. LNCS, vol. 4057, pp. 120–127. Springer, Heidelberg (2006)
12. Eckstein, I., Joshi, A., Kuo, C., Leahy, R., Desbrun, M.: Generalized surface flows for deformable registration and cortical matching. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part I*. LNCS, vol. 4791(1), pp. 692–700. Springer, Heidelberg (2007)
13. Montagnat, J., Delingette, H.: A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing* 19, 1023–1040 (2001)
14. Montagnat, J., Delingette, H.: 4D deformable models with temporal constraints: application to 4D cardiac image segmentation. *MIA* 9, 87–100 (2005)
15. Lorusso, A., Eggert, D.W., Fisher, R.B.: A comparison of four algorithms for estimating 3D rigid transformations. *BMVC*, 237–246 (1995)