

Stable Dynamic 3D Shape Models

Lars Dornheim, Klaus D. Tönnies, Jana Dornheim

Institut für Simulation und Graphik, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg

Abstract—Shape models used for the segmentation of 3D image data often suffer from high instability of shape. Current approaches to avoid this instability often result in models with high computation times and few possibilities for interaction and modelling.

We present a 3D mass-spring model which has been extended by torsion forces and the capability of explicit rotation. These models are stable with respect to shape collapse and contortion. Stability is achieved even if the model is only sparsely connected. This makes the computation efficient enough for real-time interaction.

The extended model has been successfully applied to the segmentation of the left ventricle of the human heart in 3D SPECT data.

I. INTRODUCTION

The segmentation of three-dimensional data by means of dynamic shape models is currently not standard, although it attracts increasing interest. 2D dynamic models are often not easily adaptable to the third dimension, which causes additional problems:

- The search space increases significantly in 3D.
- Due to the additional degree of freedom, many models are significantly more instable in 3D with respect to shape and contortion. In this context, figure 1 gives an idea of this problem occurring already in 2D.

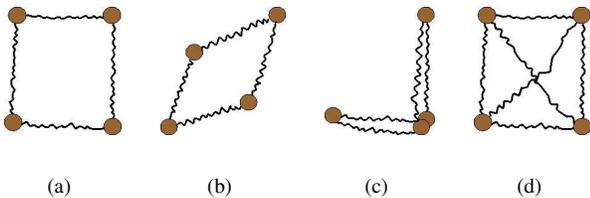


Fig. 1. (a) shows a simple mass-spring system modelling a square. In (b) this system is contorted, although no spring has changed its length. In (c) the same system is even collapsed with no single spring length changed. (d) is the same system but fully interconnected. So it tries to avoid such shape instabilities, but loses flexibility in the same moment.

Implicit (analytical) models (see [1] for a distinction of implicit and explicit dynamic models) are a common choice for the three-dimensional case, as they reduce the problem of shape instability by their way of parameterization. Their implicit parameterization mostly controls the model in a global manner and is therefore hard to control locally.

Explicit (prototype-based) approaches in 3D do not have this drawback. They allow the use of explicit knowledge about the object to segment and they simplify local user interaction. However, the lack of global control makes the definition of stable models difficult.

With the segmentation of 3D data becoming increasingly important for many applications, it would be desirable to have a stable dynamic shape model with good modelling and interaction qualities.

II. RELATED WORK

Since the introduction of dynamic models into image analysis, shape instabilities have been an important problem, particularly in 3D, where the additional degree of freedom is hard to control. Since then, different approaches to the solution of this problem have been developed.

One solution is to create *mass-spring models* with a dense mesh of springs ([2]), which require a large computational effort and therefore do not allow for real-time interaction.

The other solution is to add global constraints, which are difficult to control locally. Typical examples for this approach are implicit models like the *superquadrics* used in [3] with their drawbacks mentioned in section I or *balloons* ([4]), which are using an additional force component that lets the model expand like an inflated balloon, so that their stability is associated with a loss of concrete shape and size information in the model.

This is similar with the *Simplex Meshes* in [5], which do not contain concrete shape and size information either, as they were designed as a 3D mass-spring surface mesh. Here, the model's masses move in a self-directed manner toward the closest data points, which leads to a quasi-stability of the dynamic behaviour, provided that an appropriate initialization of the model was achieved.

Explicit dynamic models containing a specific shape and structure have apparently not been successfully applied to 3D data yet.

III. MASS-SPRING MODELS

Mass-spring models are dynamic, physics-based models known from image analysis literature (e.g. [2]). They are a theoretical model representing a dynamic system of *mass points* interconnected by elastic *springs*. Besides the physical parameters, i.e. the masses m_i of the mass points i , and the rest lengths $l_{0_{ij}}$ and spring constants k_{ij} (force exerted per length difference) of the single springs, the connection topology of the masses and springs plays a central role.

The dynamics of such a system can be described by Newtonian Mechanics. The motion of the mass points is thus solely influenced by the forces acting on the mass points. In this theoretical, ideal system, these are the spring forces $\vec{F}_{i,j}$ (equation 1), which are exerted on the mass points by the elastically deformed springs. Their value depends on the positions of the incident sensors i and j (here represented by

position vectors s_i and s_j), that are spatially associated with the mass points i and j . We assume that there is exactly one sensor per mass point, which is not a loss of generality.

$$\vec{F}_{ij} = k_{ij} \cdot (\|\vec{s}_j - \vec{s}_i\| - l_{0_{ij}}) \cdot \frac{\vec{s}_j - \vec{s}_i}{\|\vec{s}_j - \vec{s}_i\|} \quad (1)$$

As it is common in literature (e.g. in [4]), the model dynamics shall not be calculated exactly by differential equations. Instead, it will be simulated in discrete time steps of distance Δt . Starting from the speed \vec{v}_{i_t} of sensor i at time t , its speed at the time $t + \Delta t$ is calculated. Equation 2 shows the motion equation involving the spring forces, summed up for all sensors j which are connected to the sensor i . This notation of the summation of connected sensors shall be kept throughout this paper.

$$\vec{v}_{i_{t+\Delta t}} = \vec{v}_{i_t} + \frac{\sum_j \vec{F}_{ij}}{m_i} \cdot \Delta t \quad (2)$$

We recommend the addition of a general damping effect, which is motivated by the time-discrete simulation of the system. This significantly reduces problems in reaching a stable equilibrium. Equation 3 shows the extended motion equation, in which the speed of motion is dampened by a factor d .

$$\vec{v}_{i_{t+\Delta t}} = (\vec{v}_{i_t} + \frac{\sum_j \vec{F}_{ij}}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (3)$$

The damping effect represents *dynamic forces*, which only occur when the model is in motion. In contrast spring forces do not necessarily occur during the motion of the model. They shall be referred to as *internal forces* and code the shape information contained in a mass-spring model.

Image information is integrated through *external forces* exerted on mass points through sensors. Each sensor i creates a force \vec{F}_i depending on the image data. Theroto, the position of sensor \vec{s}_i as well as the type and parameterization of the sensor are relevant. Extending equation 3 by external forces results in the new motion equation 4.

$$\vec{v}_{i_{t+\Delta t}} = (\vec{v}_{i_t} + \frac{\sum_j \vec{F}_{ij} + \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (4)$$

This motion equation is extended by weight coefficients (equation 5), where w_f weights the spring force component, and $w_s(i)$ is the coefficient for the sensor force component. The latter depends on i insofar, as different types of sensors suggest different and independent weight coefficients.

$$\vec{v}_{i_{t+\Delta t}} = (\vec{v}_{i_t} + \frac{w_f \cdot \sum_j \vec{F}_{ij} + w_s(i) \cdot \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (5)$$

Once the model reaches its state of equilibrium according to equation 5, the model's adaptation to the data is complete. The equilibrium of the model's shape information and the incorporated data information has then led to a model-based segmentation of the data.

IV. EXTENDED MASS-SPRING MODELS

In this section, the mass-spring models of section III will be extended by stabilizing torsion forces, so that the problem of shape collapse and contortion in higher dimensions is no longer relevant (for details see [6]). Because of the inflexibility regarding rotation caused by these forces, possibilities are presented to let a mass-spring model perform controlled rotation while maintaining its stability.

A. Torsion Forces

Bergner already recognized in [2] the instability problem of explicit dynamic 2D models and suggests to its solution, besides a dense crosslinking, the use of an angle force which he refers to as *torque force*.

Dense crosslinking (especially in higher dimensions) solves the instability problem only at the expense of high model complexity and only indirectly, for no additional control over the degree of stability is gained.

The *angle forces* in [2] do only consider angles between two springs with a common sensor. In higher dimensions, this allows the angle plane to turn, so that a multitude of additional angle forces would be necessary leading to the same complexity problems as the dense crosslinking.

The solution to this problem is the introduction of (normalized) *rest directions* $r_{0_{ij}}$ of the springs starting from one sensor i to all its adjacent sensors j along the lines of the spring rest lengths. Spring contortions, i.e. deviations from their rest directions, can then be compensated by opposed torque moments as figure 2 illustrates.

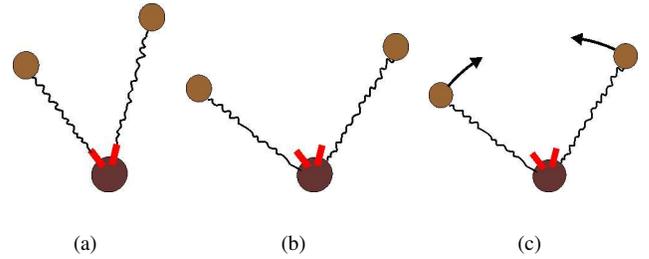


Fig. 2. (a) shows the springs in their rest directions (marked at the sensor). In (b), these springs are contorted by some external force. As a result, in (c) a torque moment acts on them in direction of their rest directions.

These torque moments manifest themselves in *torsion forces* $\vec{F}_{(i,j)}$ (equation 6), whose values depend on the sensor-specific torsion constant t_i (torque moment per torsion angle).

$$\vec{F}_{(i,j)} = \frac{t_i \cdot |\angle(\vec{r}_{ij}, \vec{r}_{0_{ij}})|}{\|\vec{r}_{ij}\|} \cdot \frac{\vec{n}_{ij}}{\|\vec{n}_{ij}\|} \quad \text{with} \quad \vec{r}_{ij} = \vec{s}_j - \vec{s}_i \quad (6)$$

They act upon the sensors j tangentially to the motion curve, in order to compensate the torsion. The working direction of the torsion forces \vec{n}_{ij} is calculated as equation 7 shows.

$$\vec{n}_{ij} = \vec{r}_{0_{ij}} - \frac{\langle \vec{r}_{ij}, \vec{r}_{0_{ij}} \rangle}{\|\vec{r}_{ij}\|^2} \cdot \vec{r}_{ij} \quad (7)$$

Stabilization works successfully in arbitrarily high dimensions. It allows an exact control of the shape stability via a weight coefficient w_t of the torsion forces. It requires as many torsion calculations per sensor, as springs incide with this sensor. Thanks to the absolute rest directions of all springs, there is no danger of mutually shifting angle planes. Equation 8 shows the motion equation extended by the torsion forces.

$$\vec{v}_{i_t+\Delta t} = (\vec{v}_{i_t} + \frac{w_f \cdot \sum_j \vec{F}_{ij} + w_t \cdot \sum_j \vec{F}_{(j,i)} + w_s(i) \cdot \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (8)$$

The torsion forces belong to the internal model forces, as they code knowledge about the model's shape. With sparse crosslinking, it can be assumed that the springs define the size of the modelled object, whereas the spring directions model its shape. Shape and size of a model can therefore be weighted individually with respect to their influence on the model adaptation.

B. Model Rotations

Because of the torsion forces' dependence on the rest directions described in the previous paragraph, these rest directions determine the orientation of the model. If these directions are specified in an absolute manner in relation to the dataset, a rotation of the model is prevented. This behaviour can be of advantage in specific applications, e.g. when the orientation of the object to segment is known in advance.

On the other hand, in order to enable model rotations, the rest directions must not be specified absolutely. Instead, it is reasonable to specify them relatively to the corresponding sensor. This sensor must be rotatable as well. When it is turned, then all rest directions of the incident springs are turned respectively.

The rotation of the sensor i should ideally result from all contortions of its incident springs, i.e. from their instantaneous spring directions \vec{r}_{ij} (see equation 7). In order to transfer the sensor at once into a state of equilibrium with respect to the contortions of all m incident springs, the inverse of the average of all its incident springs' compensation rotations is applied to it. A spring's compensation rotation is the rotation that would be required to turn a spring j back to its rest direction \vec{r}_{0ij} . Figure 3 shows the sensor rotation in the case of two contorted springs.

These compensation rotations are thereto modelled by means of a quaternion Q_{ij} , in order to exploit their simple and efficient ability to represent and combine rotations in 3D. Equation 9 shows the construction of this quaternion from the rotation angle φ_{ij} (in mathematically positive direction) and the vectored rotation axis \vec{a}_{ij} .

$$Q_{ij} = Q(\varphi_{ij}, \vec{a}_{ij}) = \begin{pmatrix} \cos \frac{\varphi_{ij}}{2} \\ \sin \frac{\varphi_{ij}}{2} \cdot \vec{a}_{ij} \end{pmatrix} \quad \text{with } \varphi_{ij} = \angle(\vec{r}_{0ij}, \vec{r}_{ij}) \quad \text{and} \quad \vec{a}_{ij} = \vec{r}_{0ij} \times \vec{r}_{ij} \quad (9)$$

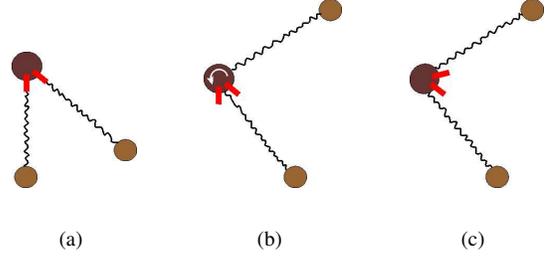


Fig. 3. (a) shows the springs in their rest directions (marked at the sensor). In (b), these spring are contorted by some external force. As a result, in (c) the sensor executes a compensatory rotation to minimize the overall contortions of the springs.

Calculating the average of all m_i individual rotations of a sensor i is nontrivial, as the successive execution of rotations in three-dimensional space is not commutative. This average is rather comparable to a superposition (simultaneous application) of the m_i -th part of each individual rotation, so that for large n the average quaternion Q_i can adequately be approximated by equation 10.

$$Q_i = \left(\prod_j Q\left(\frac{\varphi_{ij}}{m_i \cdot n}, \vec{a}_{ij}\right) \right)^n \quad (10)$$

By applying the normalized average rotation $Q_{i_{\text{norm}}}$ (equation 11) to all rest directions \vec{r}_{0ij} of the current sensor i , this sensor is aligned optimally according to the incident contortions. This is not a contradiction to its physically based model assumption, as a mass point has a torque moment of zero at its center of rotation and can therefore reach arbitrarily high rotation speeds according to Newtonian Mechanics. In this context, equation 12 shows the recalculation \vec{r}_{1ij} of the rest direction \vec{r}_{0ij} .

$$Q_{i_{\text{norm}}} = \left(\sqrt{\frac{1-\varphi^2}{\|\vec{a}\|^2}} \cdot \vec{a} \right) \quad \text{with} \quad Q_i = \begin{pmatrix} \varphi \\ \vec{a} \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} 0 \\ \vec{r}_{1ij} \end{pmatrix} = Q_{i_{\text{norm}}} \cdot \begin{pmatrix} 0 \\ \vec{r}_{0ij} \end{pmatrix} \cdot \overline{Q_{i_{\text{norm}}}} \quad \text{with} \quad \overline{Q_{i_{\text{norm}}}} = \begin{pmatrix} \varphi \\ \vec{a} \end{pmatrix} = \begin{pmatrix} \varphi \\ -\vec{a} \end{pmatrix} \quad (12)$$

If all sensors in a mass-spring model are individually rotated in the described manner, they can contort against each other, depending on the degree of crosslinking. This may make sense with respect to an exact adaptation to the segmentation target object. However, if it is known that the object to segment does not require such contortions, a rotation of the entire model would be sensible, in which all sensors are rotated equally.

Such a model behaviour can directly be derived from the free sensor rotation described above. The sensor rotations are calculated individually as described above. However, instead of applying them, the average rotation Q of all k sensors of the model (equation 13) is applied to all sensors after

normalization. This way, only the entire model rotates and aligns optimally according to the spring contortions.

$$Q = \left(\prod_{i,j} Q\left(\frac{\varphi_{ij}}{k \cdot m_i \cdot n}, \vec{a}_{ij}\right) \right)^n \quad (13)$$

V. APPLICATION AND RESULTS

The extended mass-spring model presented in paragraph IV was applied to the segmentation of the *left ventricle* (LV) of the human heart in *3D SPECT data*. The functional 3D SPECT data in this application often exhibited a low signal in the myocardium caused by infarcts. The goal was to define the anatomy of the LV by the model and adapt it to the functional image data.

The model for 41 datasets from 25 different patients was generated automatically ([7]). This produced a sparsely connected, three-dimensional mass-spring model, which allowed for a high simulation speed of the model dynamics (less than 1 minute for the complete, fully automatic segmentation process on a modern PC). It was achieved by the new stabilizing torsion forces. Interactive intervention of the user into the model adaptation process (e.g. moving of a mass) was possible in real-time.

The model behavior was extremely stable and robust. Even with initialization positions deviating from the LV positions (by up to half of the LV diameter), the model quickly and directedly moved onto the LV. A collapsed model with the torsion forces switched off was able to regain its stable shape similar to the original model shape very quickly after the torsion forces had been reactivated; this way it was still able to segment the LV successfully (figure 4).

Moreover the usefulness of the model's introduced explicit ability to rotate could be proofed. Figure 5 illustrates a case, where this ability was necessary to segment the LV correctly.

The LV has been segmented successfully in all datasets. Manual segmentations of medical experts were available for 7 datasets. Based on this gold standard, the average contour deviation of the segmentation results from the manual segmentation were measured. This deviation never extended half a voxel (with 4.795 mm voxel size). The respective Hausdorff distance of the contours was always smaller than 3 voxels.

VI. SUMMARY

The extended mass-spring model presented here is a stable dynamic 3D shape model offering good modelling and interaction capabilities. Due to their mechanics, they are more stable in higher dimensions and therefore more goal-directed than previous mass-spring models. For the first time, they allow for a controlled rotation of the whole model without contorting it.

As a result of the introduction of torsion forces, sparsely connected models can be used. For such models, an almost independent and direct weighting of model size and shape becomes possible. Furthermore, in higher dimensions, use of sparsely connected models results in a significantly reduced computational effort. That way, a simulation of the 3D model dynamics in real time becomes possible.

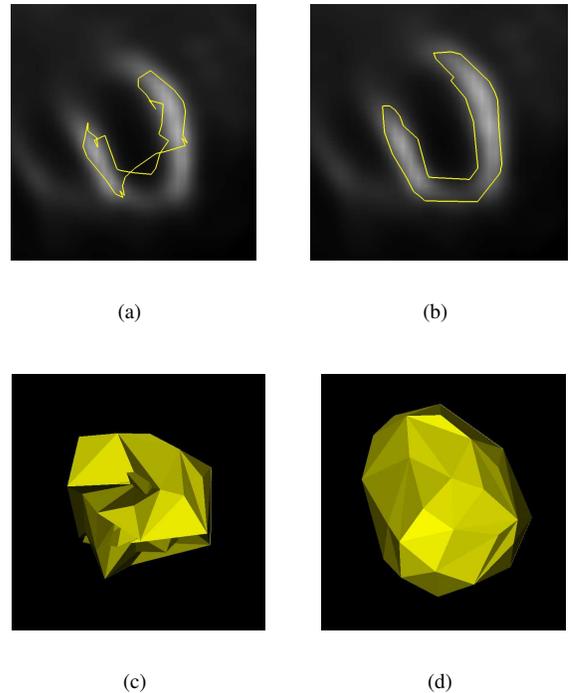


Fig. 4. The model contour on the LV (top) and the corresponding 3D model (below). (a), (c): Without torsion forces, parts of the model leave the low signal infarct region. (b), (d): With torsion forces, the segmentation is correct.

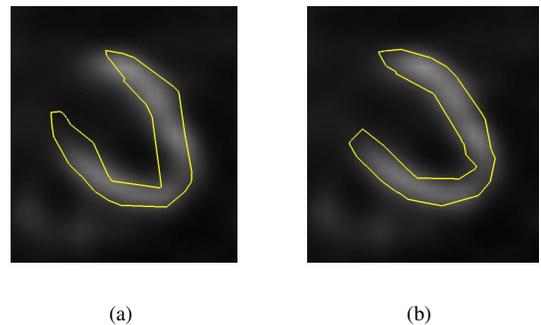


Fig. 5. 2D views of the final adaption of the LV model using torsion forces to a LV with different orientation. (a) Without explicit model rotation ability. (b) With explicit model rotation ability.

REFERENCES

- [1] A. K. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, vol. 71, pp. 109–129, 1998.
- [2] S. Bergner, S. Al-Zubi, and K. Tönnies, "Deformable structural models," in *ICIP*, 2004.
- [3] E. Bardin, L. D. Cohen, and N. Ayache, "A parametric deformable model to fit unstructured 3D data," *CVIU*, vol. 71, no. 1, pp. 39–54, 1998.
- [4] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.
- [5] H. Delingette, "Simplex meshes: a general representation for 3D shape reconstruction," in *CVPR*, 1994, pp. 856–857.
- [6] L. Dornheim, "Generierung und Dynamik physikalisch basierter 3D-Modelle zur Segmentierung des linken Ventrikels in SPECT-Daten," Diplomarbeit, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, 2005.
- [7] L. Dornheim and K. D. Tönnies, "Automatische Generierung dynamischer 3D-Modelle zur Segmentierung des linken Ventrikels in 3D-SPECT-Daten," in *Bildverarbeitung für die Medizin*, 2005.