Generalized deformation energy for reduced deformable models

Will Chang
Department of Computer Science and Engineering
University of California, San Diego
wychang@cs.ucsd.edu

October 1, 2007

1 Deformation model

A *reduced deformable model* [3, 5, 8] is a function $D(x) : \mathbb{R}^3 \to \mathbb{R}^3$ representing the deformation of space as a weighted sum of rigid transformations (picture here)

$$D(x) = \sum_j w_j(x) T_j(x).$$

Here, T_j is a rigid transformation (rotation and translation) belonging to $SE(3)$, and $w_j(x)$ is a spatially varying weighting function that defines the *region of influence* for transformation T_j. The weighting function we use must be normalized; this means that the weighting functions influencing a point $x \in \mathbb{R}^3$ must sum up to 1:

$$\sum_{k \in \{j \mid x \in \Omega_j\}} w_k(x) = 1,$$

where Ω_j is the *(effective) support* of w_j. Normalized weighting functions – also called *partition of unity functions* – can be generated from any set of weighting functions using the formula [6]

$$\varphi_j(x) = \frac{w_j(x)}{\sum_k w_k(x)}.$$

These partition of unity functions can be pre-generated or applied when deforming the space by dividing out the sum of the weights at each point. (Pre-generating POUs lead to complicated expressions, so we work with unnormalized weighting functions when we describe the deformation energy in the subsequent sections.)

2 Two previous approaches

Reduced deformable models have been used by Sumner et al. [8] and Botsch et al. [2] to express deformations of 2D and 3D shapes. The goal of their methods is to compute a deformation of *space* given user constraints on the embedded object. In the context of reduced deformable models, this amounts to solving for the transformations T_j given that they are already distributed in space according to the weighting functions $w_j(x)$. Generally there are two objectives in solving for the deformation:

1. The smoothness objective, or *deformation energy,* which guarantees that the deformation transitions smoothly throughout space and is physically plausible, and

2. The constraint objective, which specifies that the deformation matches the constrains given by the user.

In this report we describe a novel deformation energy which generalizes the previous work of Sumner et al. and Botsch et al. In Sumner et al., the reduced deformable model is described using a *deformation graph,* where the nodes g_j are the locations of the transformations T_j, and the edges connect nodes whose influences overlap. They define a regularization term, or smoothness energy for each neighbor k of node j as (picture here)

$$\sum_j \sum_{k \in N(j)} \alpha_{jk} \|T_j(g_k) - T_k(g_k)\|^2. \tag{1}$$

Here, $N(j)$ is the set of neighbors at cell j, and α_{jk} is a weighting term that measures the extent of overlap of nodes j, k. Intuitively, this constraint measures the difference between node j's transformation applied to the location of its neighbor g_k and node k's transformation applied to itself. This idea of comparing the transformations by applying them to representative sample points was proposed by Pottmann et al. as an alternative to the Frobenius norm [2, 7].

Botsch et al. define a similar deformation energy, where the deformations are compared at all points within a prescribed volume. This is in contrast to Sumner et al. which compares the transformation at a single point g_k. Specifically, Botsch et al. define *cells* C_j that subdivide space, and the transformations T_j assigned to each cell are thought to influence the region within the cell. Neighboring cells C_j, C_k define an *elastic energy* which measures the difference between the transformations T_j and T_k applied to all points in the volume $C_j \cup C_k$:

$$\sum_{\text{each pair } (j,k)} \frac{w_{jk}}{V_j + V_k} \int_{C_j \cup C_k} \|T_j(x) - T_k(x)\|^2 dx, \tag{2}$$

where V_j, V_k are the volumes of the cells C_j, C_k respectively, and w_{jk} measures the overlap between the cells (the face area shared by cells C_j and C_k divided by the sum of the distances from the centers to the face).

We can reformulate this deformation energy in two different ways. First, consider the indicator function $I_V : \mathbb{R}^3 \to \{0,1\}$, $V \subseteq \mathbb{R}^3$,

$$I_V(x) = \begin{cases} 1 & \text{if } x \in V, \\ 0 & \text{otherwise.} \end{cases}$$

We can rewrite the above equation as

$$\sum_{\text{each pair } (j,k)} \frac{w_{jk}}{\int_{\mathbb{R}^3} I_{C_j \cup C_k}(x)dx} \int_{\mathbb{R}^3} I_{C_j \cup C_k}(x) \|T_j(x) - T_k(x)\|^2 dx,$$

and in the spirit of Sumner et al., the deformation energy is defined for each neighbor k of cell j as

$$\sum_j \sum_{k \in N(j)} \frac{w_{jk}}{2\int_{\mathbb{R}^3} I_{C_j \cup C_k}(x)dx} \int_{\mathbb{R}^3} I_{C_k}(x) \|T_j(x) - T_k(x)\|^2 dx. \tag{3}$$

Second, we can also interpret this energy in terms of basis functions defined at each cell. In this case, consider the indicator $I_{\mathcal{V}_j}(x)$ on the total neighboring volume $\mathcal{V}_j = \bigcup_{k \in N(j)} C_k$ as representing the influence of cell j on other cell. We can then write the pairwise energy

$$\sum_{\text{each pair } (j,k)} \frac{w_{jk}}{\int_{\mathbb{R}^3} I_{\mathcal{V}_j}(x) I_{\mathcal{V}_k}(x)dx} \int_{\mathbb{R}^3} I_{\mathcal{V}_j}(x) I_{\mathcal{V}_k}(x) \|T_j(x) - T_k(x)\|^2 dx. \tag{4}$$

Note that we can also reformulate the regularization in Sumner et al. using an indicator at the point g_k:

$$\sum_j \sum_{k \in N(j)} \alpha_{jk} \int_{\mathbb{R}^3} I_{g_k}(x) \|T_j(x) - T_k(x)\|^2 dx \quad = \quad \sum_j \sum_{k \in N(j)} \alpha_{jk} \|T_j(g_k) - T_k(g_k)\|^2. \tag{5}$$

3 Generalized deformation energy

The observations above lead us to define a generalization of the deformation energy that uses the weighting function $w_j(x)$ from the deformation model instead of indicator functions. As above, we define the generalized deformation energy in the spirit of equation 4 as

$$\sum_{\text{each pair } (j,k)} \tau_{jk} \int_{\mathbb{R}^3} w_j(x) w_k(x) \|T_j(x) - T_k(x)\|^2 dx, \tag{6}$$

where τ_{jk} is a normalization constant defined as the inner product

$$\tau_{jk} = \langle w_j, w_k \rangle = \left(\int_{\mathbb{R}^3} w_j(x) w_k(x) dx \right)^{-1}.$$

This inner product measures the extent of overlap of the two weighting functions w_j and w_k. Alternatively, we can define the energy in the spirit of equation 3 as

$$\sum_j \sum_{k \in N(j)} \frac{\tau_{jk}}{\int_{\mathbb{R}^3} w_k(x)dx} \int_{\mathbb{R}^3} w_k(x) \|T_j(x) - T_k(x)\|^2 dx. \tag{7}$$

Equation 6 compares the transformations where the influences of T_j and T_k overlap, but equation 7 compares the them in the region of the neighbor T_k. The latter constrains in pairs in sense that node j constrains node k and vice versa, whereas the former is defined once for each unique pair (to avoid double-counting). Now, given this deformation energy, our goal is to find the rigid transformations $T_1, \ldots, T_n$ that minimize the energy:

$$\text{argmin}_{\{T_1,\ldots,T_n\}} \sum_{\text{each pair } (j,k)} \tau_{jk} \int_{\mathbb{R}^3} w_j(x) w_k(x) \|T_j(x) - T_k(x)\|^2 dx.$$

4 Constraint term

We would like to give positional constraints for select vertices rather than prescribing transformations directly. Following the formulation in Sumner et al., we have

$$\sum_l \|u'_l - D(u_l)\|^2 = \sum_l \left\| u'_l - \sum_j w_j(u_l) T_j(u_l) \right\|^2, \tag{8}$$

where (u_l, u'_l) is the original / translated constraint pair, and k iterates in the set $j \in \{k \,|\, u_l \in \Omega_k\}$.

5 Numerical solution & implementation

We use a Newton solver in the same way as Botsch et al. to solve this optimization problem. To constrain the transformations to be rigid, we linearize the transformations using affine approximations

$$T_j(x) \approx A_j(x) = x + \bar{c}_j + (c_j \times x),$$

where $\bar{c}$ is a translation and c is an angular velocity. This equation is derived by taking the first derivative of a rigid motion, and represents uniform translations, uniform rotations, and uniform helical motions [7]. At each step of the solver, we replace the transformations with their linearized versions, yielding a sparse linear system that can be solved in closed form (calculation details to follow?):

$$\sum_{\text{each pair } (j,k)} \tau_{jk} \int_{\mathbb{R}^3} w_j(x) w_k(x) \|A_j(x) - A_k(x)\|^2 dx.$$

Note that the A_j are not rigid, so we map them to their closest rigid counterpart using a simple local shape matching technique [1, 4]. The closest rigid transformation $T_j = (R_j, t_j)$ is found by solving another minimization problem:

$$\operatorname{argmin}_{T_j} \int_{\mathbb{R}^3} w_j(x) \|T_j(x) - A_j(x)\|^2 dx.$$

For the Gaussian weight function

$$w_j(x) = \exp\left\{-\frac{\|x - g_j\|^2}{2{\sigma_j}^2}\right\}$$

Following Horn's procedure [4], we can derive the closed form solution for the rotation R_j. This solution is given as a quaternion of the form:

$$q = \left[1 + \sqrt{1 + \|c_j\|^2}\ , \quad c_{jx}, \quad c_{jy}, \quad c_{jz}\right]^\top.$$

We normalize this quaternion and find the corresponding rotation matrix R_j. This rotation is applied with respect to the centroid of Ω_j as the origin, i.e.

$$T_j(x) = R_j(x - g_j) + g_j + t_j.$$

Then, the translation t_j is the difference between the centroids of the two transformations, which results in

$$t_j = \bar{c_j} + (c_j \times g_j).$$

6 Misc. Implementation Notes

Replacing sums with more equations. Note that instead of minimizing the expression

$$\sum_{\text{each pair } (j,k)} \tau_{jk} \int_{\mathbb{R}^3} w_j(x) w_k(x) \|A_j(x) - A_k(x)\|^2 dx$$

we can instead independently minimize each summand

$$\tau_{jk} \int_{\mathbb{R}^3} w_j(x) w_k(x) \|A_j(x) - A_k(x)\|^2 dx$$

since they are positive. This also holds with the constraint term, i.e.

$$\sum_l \left\| u_l' - \sum_j w_j(u_l) T_j(u_l) \right\|^2$$

can be broken into minimizing each summand

$$\left\| u_l' - \sum_j w_j(u_l) T_j(u_l) \right\|^2.$$

However, note that we cannot separate the individual terms in the sum within the norm.

The derivatives of the vertex constraint term. The vertex constraint term is

$$\|u_l' - D(u_l)\|^2 = \Big\|u_l' - \sum_j D_j(u_l)\Big\|^2 = \Big\|u_l' - \sum_j w_j(u_l)T_j(u_l)\Big\|^2,$$

where $j \in \{k \,|\, u_l \in \Omega_k\}$. We compute derivatives of this function below:

$$\begin{aligned}
\Big\|u_l' - \sum_j D_j(u_l)\Big\|^2 &= \Big[u_l' - \sum_j D_j(u_l)\Big]^\top \Big[u_l' - \sum_j D_j(u_l)\Big] \\
&= \Big[{u_l'}^\top - \sum_j D_j^\top(u_l)\Big]\Big[u_l' - \sum_j D_j(u_l)\Big] \\
&= {u_l'}^\top u_l' - 2\sum_j {u_l'}^\top D_j(u_l) + \Big[\sum_j D_j^\top(u_l)\Big]\Big[\sum_j D_j(u_l)\Big] \\
&= {u_l'}^\top u_l' - 2\sum_j {u_l'}^\top D_j(u_l) + \sum_{\forall \text{ pairs } (j,k)} D_k^\top(u_l)D_j(u_l).
\end{aligned}$$

(The sum in the last equation iterates between all possible pairs of $j, k \in \{i \,|\, u_l \in \Omega_i\}$.) To minimize this term, we take each partial derivative and find the value that make it equal to zero. So each derivative contributes a constraint in the linear system. When we compute the derivative with respect to node j, the only relevant terms are

$$-2{u_l'}^\top D_j(u_l) + 2\sum_{k\neq j} D_k^\top(u_l)D_j(u_l) + D_j^\top(u_l)D_j(u_l).$$

(Here the sum iterates in the set of all $\{k \,|\, u_l \in \Omega_k\}$ except for $k = j$.) Now, taking the derivative with respect to c_j and $\bar{c}_j$, we obtain:

$$\partial/\partial c_j \Rightarrow -2w_j(u_l)(u_l \times u_l') + 2w_j(u_l)\sum_k w_k(u_l)\Big[(u_l \times \bar{c}_k) + ({u_l}^\top u_l)c_k - ({u_l}^\top c_k)u_l\Big],$$

$$\partial/\partial \bar{c}_j \Rightarrow -2w_j(u_l)u_l' + 2w_j(u_l)\sum_k w_k(u_l)\Big[u_l + \bar{c}_k + (c_k \times u_l)\Big].$$

Setting this equal to zero, we get the two constraints:

$$\begin{aligned}
\sum_k w_k(u_l)\Big[(u_l \times \bar{c}_k) + ({u_l}^\top u_l)c_k - ({u_l}^\top c_k)u_l\Big] &= u_l \times u_l', \\
\sum_k w_k(u_l)\Big[u_l + \bar{c}_k + (c_k \times u_l)\Big] &= u_l'.
\end{aligned}$$

Simplifying this further, this amounts to the two constraints:

$$\begin{aligned}
u_l \times D(u_l) &= u_l \times u_l', \\
D(u_l) &= u_l'.
\end{aligned}$$

7 Results

References

[1] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 11–20, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[2] Mario Botsch, Mark Pauly, Martin Wicke, and Markus Gross. Adaptive space deformations based on rigid cells. In *Computer Graphics Forum: EUROGRAPHICS 2007 Papers*, volume 26, pages 339–347, 2007.

[3] Kevin G. Der, Robert W. Sumner, and Jovan Popović. Inverse kinematics for reduced deformable models. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1174–1179, New York, NY, USA, 2006. ACM Press.

[4] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4), 1987.

[5] Doug L. James and Christopher D. Twigg. Skinning mesh animations. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 399–407, New York, NY, USA, 2005. ACM Press.

[6] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 463–470, New York, NY, USA, 2003. ACM Press.

[7] Helmut Pottmann, Qi-Xing Huang, Yong-Liang Yang, and Shi-Min Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *Int. J. Comput. Vision*, 67(3):277–296, 2006.

[8] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 80, New York, NY, USA, 2007. ACM Press.