Pacific Graphics 2012
C. Bregler, P. Sander, and M. Wimmer
(Guest Editors)

Volume 31 (2012), Number 7

Homunculus Warping: Conveying importance using self-intersection-free non-homogeneous mesh deformation

Bernhard Reinert Tobias Ritschel Hans-Peter Seidel
MPI Informatik

Figure 1: Left to right: *Input geometry and importance on a human figure and five views of the according Homunculus Warping.*

Abstract

Size matters. Human perception most naturally relates relative extent, area or volume to importance, nearness and weight. Reversely, conveying importance of something by depicting it at a different size is a classic artistic principle, in particular when importance varies across a domain. One striking example is the neuronal homunculus; a human figure where the size of each body part is proportional to the neural density on that part. In this work we propose an approach which changes local size of a 2D image or 3D surface and, at the same time, minimizes distortion, prevails smoothness, and, most importantly, avoids fold-overs (collisions). We employ a parallel, two-stage optimization process, that scales the shape non-uniformly according to an interactively-defined importance map and then solves for a nearby, self-intersection-free configuration. The results include an interactive 3D-rendered version of the classic sensorical homunculus but also a range of images and surfaces with different importance maps.

1. Introduction

Simple and effective visual representations of complex relations are hard to come by [TWM97]. While there are many different approaches and diagram types, the problem is to pick one that is suitable for the current task and is understood immediately. To clarify the relationship between different data dimensions it is sometimes useful to visualize one dimension on the domain of a different one, e. g., to illustrate the number of inhabitants of a country on a world map. A common way of visualizing this data is to use false color coding, despite all its known shortcomings [BTI07]. The main shortcoming is the complex human color perception that varies considerably among different viewers and even penalizes achromates as well as the conflicts it causes for three-dimensional surfaces in combination with shading.

Generally perceived differences in size are remarkably invariant under different viewing conditions [Cut87, Gre63] especially for familiar objects [Bin93] e. g., the human body. Hence an alternative to false color coding is to encode the scalar field directly as size.

A prominent contemporary example of this type of visual coding is a didactical concept from neuroscience: the homunculus (Fig. 2). Here the three-dimensional surface of the human body is locally changed to reflect the density of sensorical and motorical neural density [PR50]. The idea of encoding importance in size however is not a novel idea. It dates back to examples from early renaissance (Fig. 2, c) where a donor of a painting is depicted smaller, as he is less important than the depicted saint or even before that, such as in medieval book illustrations (Fig. 2, d). Considering that the

Figure 2: *Four examples of local size changes to convey importance* (Left to right)*: The motorical* (a) *and sensorical* (b) *homunculus (Natural History Museum, London).* c)*: A renaissance painting by Gentile da Fabriano (1370—1427): "Mary Enthroned with the Child, Saints and a Donor" depicting the donors as smaller and less important.* d)*: Miniature from the "Tetraevangelia of Ivan Alexander" (14th century, Bulgaria) depicting the tsar and his wife as more important than his children.* e)*: Limestone engraving from the tomb of Akhenaten (2nd century BC, Cairo Museum, Egypt) where cascading size conveys importance.*

metaphor is as old as art itself, e. g., used in Egypt (Fig. 2, e), it seems worthwhile to make it accessible for digital media.

In this paper we introduce an approach to locally deform a domain such as a 2D image or 3D surface such that it conveys importance. The computational challenge, is to effectively minimize distortions, such as fold-overs, while fulfilling the goals imposed by the importance field to convey. Besides directly depicting importance, our approach can be used for focus+context visualization [LA94], as a 2D or 3D editing metaphor on its own or for the design of 2D and 3D aggregate texture design.

2. Related work

Deformation Deforming images [Wol98] and surfaces [SP86] is a classic computer graphics topic. Recently, deformations that minimize distortions [ACOL00, IMH05, SA07, SMW06] have received considerable interest. The deformations resulting from those methods aim to follow artistic constraints, be smooth and to locally behave like rotations. Different from our approach, the usual input to such methods are positional constraints: a set of handles that are moved to another location. In our approach, no single location is deformed but the entire domain changes its local size.

Furthermore, we have to avoid self-collisions that occur by scaling deformations in free space. The volume-preserving approach of von Funck et al. [VFTS06] also produces a self-intersection-free vector field. The PriMo modeling system [BPGK06] is a rare example that allows for local scaling of surface area, but without an account for intersections that occur for drastic and complex local size changes such as we target (Fig. 1). The approach of Harmon et al. [HPSZ11] includes continuous intersection handling into the modelling process, that allows for sliding motions such as the one of cloth on a character by eliminating space-time interference.

Non-homogenous scaling was addressed in the context of retargeting [KSSCO08], where the bounding box of one object is fit into another one, while trying to locally preserve salient regions and structure, e. g., such that spheres remain spheres. Our approach seeks to do the opposite, as it locally wants to change scaling but has to prevent intersections and minimize distortion.

Visualization Besides the technical challenges that need to be solved, the visualization community has identified the opportunities provided by local size changes under the label of "focus+context" visualizations (FCV) [LA94]. The underlying magnifying glass metaphor was extended to 3D surfaces and volumes [WZMK05]. Recent works combine all levels of detail in a single image with smooth transitions between the different levels [HMC11]. Different viewports and the area each one of them should cover in the final image can be defined and the method smoothly interpolates between these viewports. The result however is viewport dependent and it is not clear if and how an extrapolation for different viewports could be added.

Beyond lenses, other deformations can be applied in 3D to achieve focus+contex. Wang et al. [WLT08, WWLM11] deform a 3D surface to achieve focus+context by allowing to deform free space to deform more and minimize distortions of the surface. Our problem is different in two ways. First, their technique handles all voxels inside of a bounding cube whereas our technique ignores voxels not occupied by mesh vertices. Second, their focus+context deformation field is intentionally low-frequency (e. g., a single large gradient proportional to distance from a focus point). Our solution can be applied to both high-frequency deformation, e. g., the crocodile teeth in Fig. 6 as well as focus+context deformation as used by Wang et al., e. g., Fig. 7, i.

Physically-based animation Our solution draws inspiration from the way deformations are computed for physical animation [MHHR07]. The objective of our approach is a static rest state, the time-dependent change of a deformation subject to certain forces (i. e., of a piece of cloth) and momenta are not important for our application. Including collision detection in physical animation is common, however it was not adressed

for shape manipulation or focus+context visualization to our knowledge.

3. Approach

An overview of our approach is given in Fig. 3. The input boundary (Sec. 3.1) is first voxelized (Sec. 3.2) and subsequently the importance is defined on the voxelization (Sec. 3.2). Next, a two step optimization (Sec. 3.3) is performed to fulfill the importance goals, retain the object appearance and smoothly avoid self-collisions. Finally, a deformation transfer (Sec. 3.5) from the voxelization back onto the boundary is performed.

3.1. Input

Input to our approach is a detailed $d-1$-dimensional boundary $\mathcal{B}$ in $\mathbb{R}^d$ and a scalar importance field $\mathcal{F}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^+$ defined on its voxelization. Output is a new boundary $\mathcal{B}'$, where the local size is proportional to the importance, with an intersection-free deformation. In practice we deal with a discrete d-dimensional polygonal mesh $B = (V^{\mathrm{B}}, E^{\mathrm{B}})$ defined by its vertex positions V^{B} and its edges E^{B}.

3.2. Voxelization

We perform a voxelization to become independent of the actual underlying boundary and its representation, to achieve a more uniform sampling, to become robust to potential (erroneous) self-intersections present in the input B and to better reflect the solid nature of the shape inside the boundary B.

The first step is to voxelize B into an edge complex $C = (V, E)$ with vertices $V = \{v_0 \ldots, v_n \in \mathbb{R}^d\}$ and edges $E = \{e_0, \ldots, e_m \in \mathbb{N}^2\}$ as well as into a tetrahedral complex $C_{\mathrm{t}} = (V, T)$ with the same vertices V and tertrahedra $T = \{t_0, \ldots, t_o \in \mathbb{N}^4\}$. For images the alpha channel is used to find all covered voxels. To this end, a finite discretization is chosen that partitions space into virtual cells. For each cell that is at least partially inside B, all its edges, its vertices and its tetrahedra are appended to C and C_{t} respectively. The scalar field importance values $F = \{f_0, \ldots, f_m \in \mathbb{R}\}$ are assigned to each edge $e \in E$. Also, let $\Delta_i \in \mathbb{R}^d$ to be the i-th edge's difference vector, i. e., $\Delta_i = v_j - v_k$.

In practice it cannot be expected that B is a closed manifold as done for (GPU) voxelization [DCB*04]. Instead we perform solid voxelization that tolerates small holes (Fig. 4). First, the boundary is voxelized into a grid, by simply rasterizing its triangles. Doing so all cells that are partially covered by B are marked and consequently, holes smaller than one cell are closed. Holes larger than a cell, e. g., from non-orientable objects such as cloth lead to shell-like objects without volume. We did not further investigate deformation of such objects and report results restricted to surfaces with holes no larger than one cell. We assume, that the cell size is small enough to resolve all details, e. g., to separate hand and body in Fig. 3. In a final step a flood fill finds all outside voxels: Every voxel that is not outside is considered to belong to the inside.

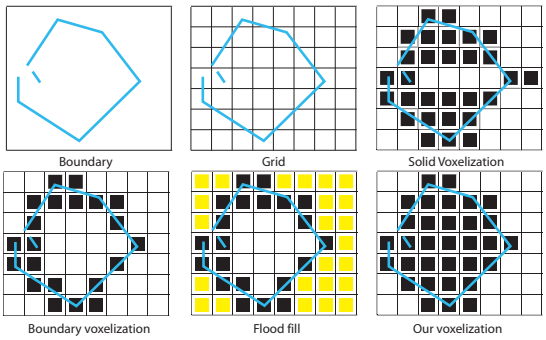

Figure 4: *Our voxelization is more robust to meshing defects.*

3.3. Optimization

Our optimization is performed in two steps: The first step finds a mesh configuration that tries to preserve edge directions and set edge lengths according to importance (edge optimization). The second step finds a collision-free configuration of the first step's solution which evenly distributes the distortions introduced by collision response over the complex (collision optimization).

Edge optimization Here, we seek to find vertex positions V' such that edges in (V', E) have lengths proportional to their importance values F. In other words, that for each edge $e_i \in E$ with importance $f_i \in F$ it holds, that

$$\Delta_i' = f_i \Delta_i$$

which leads to a linear system of equations. Please note, how this formulation also tries to preserve initial edge directions. Since the importance values f_i are user-defined and thus not restricted, in general not all edge lengths can be satisfied simultaneously. Hence the goal becomes to find the minimal sum of errors in a least squares sense, i. e.,

$$\underset{V'}{\operatorname{argmin}} \sum_{i=1}^{m} \|\Delta_i' - f_i \Delta_i\|_2^2. \qquad (1)$$

The global minimum of this equation can be found by setting its directional derivatives to zero. This step of our pipeline generalizes the approach of Sorkine and Alexa [SA07] by adding a per-edge scaling factor.

Collision optimization After the previous step, the edges in the complex (V', E) are scaled optimal while reducing distortions but the complex (V', T) is potentially self-intersecting. A second optimization is used to resolve self-intersections while preserving the appearance of the previous step as much as possible. To this end, collisions need to be detected, resolved and the resulting deformation needs to be distributed. To fulfill both goals at the same time, each individual goal is iteratively resolved in turn as in [MHHR07]. While the first optimization strived to preserve edge directions the second optimization should just maintain edge lengths. Doing so

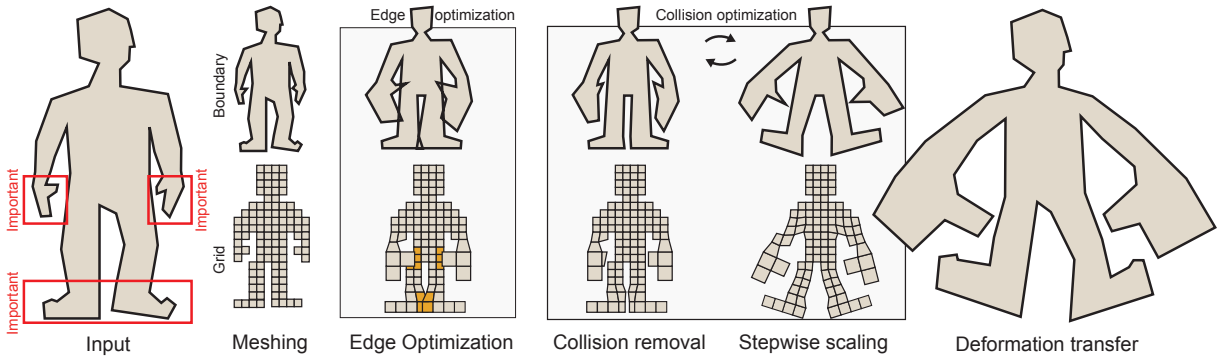

Figure 3: *Overview of our approach* (Left to right). *Input is a detailed boundary representation such as a 3D mesh with an annotation of importance. First, the boundary is simplified and voxelized into a regular grid. For didactical purposes the optimization is illustrated both for the the boundary* (Top) *and the regular grid actually used in our computation* (Bottom). *The deformation consists of an edge and a collision optimization. The edge part first locally scales the elements proportional to their importance and distributes the distortion by preserving relative coordinates. While the resulting shape is smooth and conveys the importance well, it results in collisions* (Yellow cells). *In the collision part, collisions are removed, which re-introduces distortions. Those distortions are removed by locally preserving edge lengths. This can again result in new collisions, so the collision step is iterated. Finally, the deformation of the grid is propagated onto the original input mesh. Note, how the resulting shape follows the prescribed importance, has no collisions, smoothly distributes collision response over the complex and parts (e. g., the hands) were rotated to avoid collisions.*

distributes the collision response over the complex since it allows edges to rotate.

Since collision detection is a continuous process only small scalings can be established concurrently; introducing the full scaling of Eq. 1 at once would lead to many severe collisions which are harder to resolve. As a solution we propose to introduce only a small portion of the optimal length $l_i = \|\Delta_i'\|_2$ for the i-th edge in multiple iteration steps, i. e., the target length for each edge e_i becomes λl_i with incrementally increasing $\lambda \in [0,1]$. Doing so, will result in much fewer collisions in every step that are much easier to resolve.

For the stepwise scaling of the mesh we solve the equation

$$\|\Delta_i''\|_2 = \lambda l_i$$

for every i. Since the values l_i are derived from a satisfiable configuration of the mesh the resulting system of equations has at least one solution (namely V'). However V'' should be collision-free which introduces distortions to the mesh and turns the system into a potentially overdetermined system of equations that has to be minimized, i. e.,

$$\underset{V''}{\operatorname{argmin}} \sum_{i=1}^{m} \|\|\Delta_i''\|_2 - \lambda\, l_i\|_2^2 \qquad (2)$$

with collision-free V''. Since this system involves finding a constrained solution V'' it has to be minimized iteratively. Solving this equation has a two-fold purpose: First it incrementally scales the complex to the designated size and second it distributes the distortions introduced by collision response over the complex.

For higher similarity to the input complex collisions should not be resolved precisely but rather vertices should keep a certain distance to the boundary. We propose to approximate this behaviour by defining for every vertex i a sphere with a radius $r_i \in \mathbb{R}^+$ that equals the minimal distance to every adjacent vertex, i. e., $r_i = \min\{\|v_i'' - v_j''\|_2 | (i,j) \in E\}$. Collision between all vertices and spheres are found using spatial hashing build in every iteration [THM*03]. When a collision of a vertex v_i'' with a sphere surrounding vertex v_j'' is found, the vertex is projected to the surface of the sphere in the direction $v_i'' - v_j''$, i. e., the vertex is corrected by

$$v_i'' += \left(\frac{r_i}{\|v_i'' - v_j''\|_2} - 1\right) \frac{v_i'' - v_j''}{2}.$$

The same correction is performed for vertex v_j in the opposite direction.

3.4. Equation minimization

We minimize Eq. 1 and Eq. 2 based on parallel operations (using OpenCL) as follows.

Equation 1 leads to a linear system $\mathsf{A}\mathbf{x} = \mathbf{b}$, where the rows a_i of A express the differences between vertex i and its edge-connected vertices; the values b_i of $\mathbf{b}$ reflect the sum of the signed distances for every coordinate between vertex i and its edge-connected vertices. A therefore is sparse, symmetric and positive-semidefinite, which allows to use efficient solvers such as Cholesky decomposition or the Gauss-Seidel method.

Minimizing Eq. 2 involves solving a constrained system of equations of second degree polynomials. We solve this iteratively using constraint projection as described by Müller et al. [MHHR07], which allows to treat edges independently

```
V^(0) := V;
// Edge optimization
for t from 0 to n_e − 1
  for i from 0 to n in parallel
    v_i^(t+1) := (b_i + a_ii v_i^(t) − (A v^(t))_i)/a_ii;
// Remember optimal edge length
for all edges e = (i, j)
  l_i := ‖Δ'_i^(n_e)‖_2;
V^(n_e) := V;
// Collision optimization
for t from n_e to n_e + n_c − 1
  λ := (t − n_e)/(n_c − 1);
  // Stepwise scaling
  P := 0;
  for i from 0 to n in parallel
    p_i := v_i^(t);
    for all edges e = (i, j) connected to v_i
      p += ((λ l_i/‖p_i − v_j^(t)‖_2) − 1)(p_i − v_j^(t))/2;
  // Calculate radii
  R^(t) := FLT_MAX;
  for i from 0 to n in parallel
    for all edges e = (i, j) connected to v_i
      r_i^(t) := min(r_i^(t), ‖p_i − p_j‖_2);
  // Handle collisions
  for i from 0 to n in parallel
    for all v_j with ‖p_i − p_j‖_2 < r_i
      p += ((r_i^(t)/‖p_i − p_j‖_2) − 1)(p_i − p_j)/2;
    v'_i^(t+1) := p;
```

Figure 5: *Pseudo-code of our approach (Please see text).*

and admits to handle scaling and collision response in a unified way. Again, since the system of equations only specifies edge lengths, it is underdetermined. This approach is compatible with the solution to Eq. 1.

Since both our solvers work iteratively they turn the solutions V' and V'' into sequences $(V'_0,\ldots,V'_{n_e})$ and $(V''_0,\ldots,V''_{n_c})$ with $V'_0 = V''_0 = V$, $n_e \in \mathbb{N}^+$ and $n_c \in \mathbb{N}^+$ as the number of iterations for Eq. 1 and Eq. 2 respectively. We combine both sequences in a single sequence $V^{(i)} : \mathbb{N} \to \left(\mathbb{R}^d\right)^n = (V'_0,\ldots,V'_{n_e},V''_0,\ldots,V''_{n_c})$. The pseudo-code of our approach is given in Fig. 5. After a user interaction, the last solution V'' is used as the next initial guess.

Theoretically the increment in edge lengths must not be larger than the minimum distance between two vertices divided by the maximum distance between two vertices of the complex. However this value highly depends on the scalar field F and in practice that much larger λ are feasible.

3.5. Deformation transfer

In a pre-process, the barycentric coordinates $\alpha_i,\beta_i,\gamma_i,\delta_i \in \mathbb{R}$ with $\delta_i = 1-\alpha_i-\beta_i-\gamma_i$ for the containing tetrahedron $t_j = (a_j,b_j,c_j,d_j) \in \mathbb{N}^4$ defined by the vertices v_{a_j}, v_{b_j}, v_{c_j} and v_{d_j} are computed for every vertex v_i^{B} in B [MG04]. Finally, the deformed vertex positions V'' are used to deform the original boundary B. At runtime, the new position is compute as $v_i^{\mathrm{B}'} = \alpha_i v''_{a_j} + \beta_i v''_{b_j} + \gamma_i v''_{c_j} + \delta_i v''_{d_j}$. Using the barycentric coordinates the scalar field F can also be resampled on the actual boundary B to visualize the importance on it.

4. Results

First, typical outputs of our approach for 3D surfaces are shown in Fig. 7 and for 2D images in Fig. 6. All results can be manipulated interactively and the solution is found incrementally. A reasonable linear deformation feedback is returned in a fraction of a second before the solution converges in less than a second. Typically the iterative approach for the edge optimization converges in less than $n_e = 20$ iterations. The collision optimization takes in the order a few minutes (typically $n_c = 1000$ iterations). The high-frequency importance field (and by that, also detailed deformation field) requires a high number of voxels, typically $128 \times 128 \times 128$.

5. Discussion and Conclusion

In this paper we proposed an approach to change the local scale of a surface to convey importance. The resulting shapes are free of self-intersections and avoid distortions.

Compared to other visualizations of scalar (importance) fields, using size is immediately understood and could be used to illustrate scalar information on complex domains even for naïve observers. This is, because a twice-as-large hand in Fig. 1 naturally is perceived twice-as-much-of-something (e. g., neural density), whereas there definitely is no color that is twice as much as another color. Color-coded scalar fields are superior to our approach when qualitatively comparing locations on the domain i. e., finding all locations with the same color label.

In future work, the performance should be further increased to allow fully interactive focus+context navigation. Human perception of area, weight, size, scale and its dependence on spatial organization of the underlying domain certainly have an influence on our approach but yet remain to be modeled computationally. Avoiding distortion of salient features such as faces (Fig. 1) or symmetry, as well as rigid components is potential future work. Finally, our approach could be evaluated in a perceptual study that would lead to a more formal understanding of human task performance when using size vs. color-coded information.

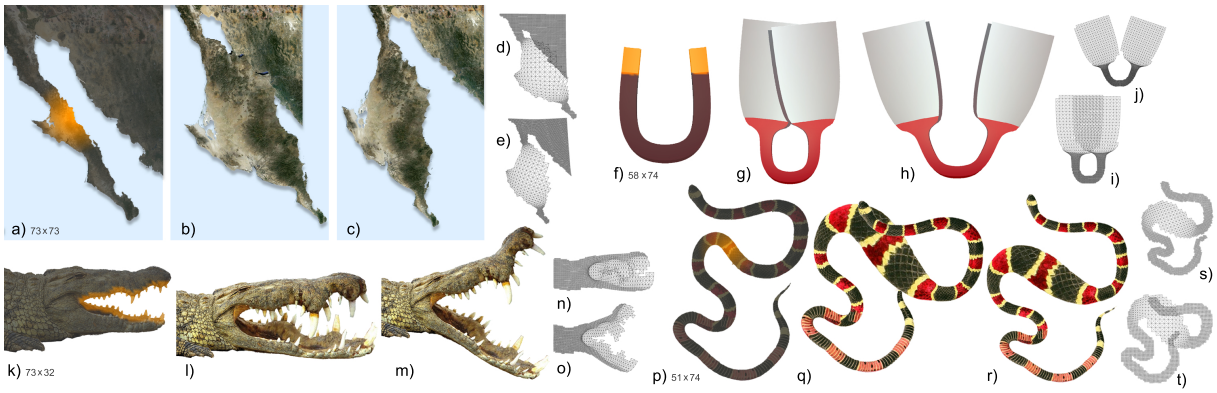

Figure 6: *Our approach applied to two-dimensional images. Deformation for geographical map data with importance* (a) *where classic deformation leads to intersections* (b,d) *avoided by our approach* (c,e). *Conveying magnetism* (f) *on the end of the u-shape results in interference* (g,i) *wheras our approach bends the domain to make room* (h,j). *High-frequency deformation on the crocodile scales only the teeth* (k) *leading to many collisions* (l,n). *Our method results in opening the mouth* (m,o). *As only the teeth grow, the jaw globally and locally bends to make more room for the individual teeth. A snake digesting its prey* (p) *self-intersects when applying classic deformation* (q,t) *that is prevent by our approach* (r,s). *For every example the resolution of the voxelization is listed.*

References

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH* (2000), pp. 157–164. 2

[Bin93] BINGHAM G.: Perceiving the size of trees: Form as information about scale. *J Exp. Psych.: Human Perception and Performance 19*, 6 (1993), 1139. 1

[BPGK06] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: PriMo: Coupled prisms for intuitive surface modeling. In *Proc. SGP* (2006), pp. 11–20. 2

[BTI07] BORLAND D., TAYLOR II R.: Rainbow color map (still) considered harmful. *IEEE CG & App.* (2007), 14–17. 1

[Cut87] CUTTING J.: Rigidity in cinema seen from the front row, side aisle. *J Exp. Psych.: Human Perception and Performance 13*, 3 (1987), 323. 1

[DCB*04] DONG Z., CHEN W., BAO H., ZHANG H., PENG Q.: Real-time voxelization for complex polygonal models. In *Proc. Pacific Graphics* (2004), pp. 43–50. 3

[Gre63] GREGORY R.: Distortion of visual space as inappropriate constancy scaling. *Nature 199*, 678-91 (1963). 1

[HMC11] HSU W.-H., MA K.-L., CORREA C.: A rendering framework for multiscale views of 3d models. In *Proc. SIGGRAPH Asia* (2011), pp. 131:1–131:10. 2

[HPSZ11] HARMON D., PANOZZO D., SORKINE O., ZORIN D.: Interference-aware geometric modeling. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 6 (2011), 137. 2

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J.: As-rigid-as-possible shape manipulation. In *ACM Trans. Graph* (2005), vol. 24, pp. 1134–1141. 2

[KSSCO08] KRAEVOY V., SHEFFER A., SHAMIR A., COHEN-OR D.: Non-homogeneous resizing of complex models. In *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* (2008), vol. 27, p. 111. 2

[LA94] LEUNG Y., APPERLEY M.: A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. CHI) 1*, 2 (1994), 126–160. 2

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proc. Graphics Interface* (2004), pp. 239–246. 5

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (2007), 109–118. 2, 3, 4

[PR50] PENFIELD W., RASMUSSEN T.: The cerebral cortex of man; a clinical study of localization of function. 1

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proc. SGP* (2007), pp. 109–116. 2, 3

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. In *ACM Trans. Graph. (Proc. SIGGRAPH)* (2006), vol. 25, pp. 533–540. 2

[SP86] SEDERBERG T., PARRY S.: Free-form deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics 20*, 4 (1986), 151–160. 2

[THM*03] TESCHNER M., HEIDELBERGER B., MÜLLER M., POMERANETS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. In *Proc. VMV* (2003), pp. 47–54. 4

[TWM97] TUFTE E., WEISE MOELLER E.: *Visual explanations: images and quantities, evidence and narrative*. Graphics Press Cheshire, CT, 1997. 1

[VFTS06] VON FUNCK W., THEISEL H., SEIDEL H.: Vector field based shape deformations. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 1118–1125. 2

[WLT08] WANG Y.-S., LEE T.-Y., TAI C.-L.: Focus+context visualization with distortion minimization. *IEEE Trans. Vis. and Computer Graphics 14*, 6 (2008), 1731–1738. 2, 7

[Wol98] WOLBERG G.: Image morphing: a survey. *The visual computer 14*, 8 (1998), 360–372. 2

[WWLM11] WANG Y.-S., WANG C., LEE T.-Y., MA K.-L.: Feature-preserving volume data reduction and focus+context visualization. *IEEE Trans. Vis. and Computer Graphics 17* (2011), 171–181. 2

[WZMK05] WANG L., ZHAO Y., MUELLER K., KAUFMAN A.: The magic volume lens: An interactive focus+context technique for volume rendering. In *Proc. IEEE VIS* (2005), pp. 367–374. 2

Figure 7: *Results of classic deformation* (Left) *compared to our collision-aware approach* (Right). a)*: Noise sources of an airplane.* b)*: Injury probability of a goal keeper* c)*: Black market price of body parts of an elephant.* d)*: Force field on a robot.* e)*: Pollen concentration of flowers.* f)*: Homunculus.* g)*: Heat field on a street lamp.* h)*: Diluent concentration of bike parts in contact with human body.* i)*: Focus+context visualization of a colon model (cf. [WLT08]).* j)*: Comic character deformation. For every example the resolution of the voxelization is listed.*