Focus+Context Visualization with Distortion Minimization

Yu-Shuen Wang, Tong-Yee Lee *Member, IEEE*, and Chiew-Lan Tai

Abstract— The need to examine and manipulate large surface models is commonly found in many science, engineering, and medical applications. On a desktop monitor, however, seeing the whole model in detail is not possible. In this paper, we present a new, interactive Focus+Context method for visualizing large surface models. Our method, based on an energy optimization model, allows the user to magnify an area of interest to see it in detail while deforming the rest of the area without perceivable distortion. The rest of the surface area is essentially shrunk to use as little of the screen space as possible in order to keep the entire model displayed on screen. We demonstrate the efficacy and robustness of our method with a variety of models.

Index Terms— Focus+Context visualization, magnification, bounding space

1 Introduction

Using a magnifying lens to enlarge the focal region of an object is very common in the real world. If the object contains tiny details or is itself very small, observing the object through a magnifying lens is usually helpful and perhaps even necessary. Such local magnification is also useful for the visualization of a high-resolution model on a low-resolution display device. The user moves the mouse cursor to specify a region that he/she wants to observe in more detail, and the system displays an enlarged version of that region in another part of the screen. However, such a straightforward sub-window technique requires the user to interpret translation relation between the sub-window and the main window. Another approach is to zoom in on the region of interest while cropping off parts of object that are farther away. Such local magnification visualization, however, cannot maintain a full view of the model (see Figure 1).

To display complex models on the screen which has limited resolution, researchers have proposed Focus+Context frameworks, which magnify the area of interest without clipping off the other parts [2,3,5–8,14]. These methods expand the region of interest through the theory of optical lens or other distortion methods to achieve this aim. These methods allow the user to clearly observe the model's detail in the region of interest while not losing the overall view of the model's shape and topology. This kind of visualization conveys a complete visual message to the user and reminds the user of the overall perception of the model at all time, while the user's attention is focused on a local region.

We propose a framework to interactively deform a polyhedral model to achieve Focus+Context visualization. Our goal is to non-homogeneously rescale different regions while preventing the global bounding space of the model from being expanded (see Figure 2) and thus keeping the entire model displayed on the screen. We construct a uniform grid space for a given model such that the model vertices are embedded in the grid cubes. While enlarging the cubes covering the user-specified focal region, our system automatically reduces the other cubes to keep the entire model within the global bounding space. The deformed model is reconstructed by computing each model vertex as a linear combination of its respective set of cube vertices in the deformed grid space.

Since our goal is to constrain the model to be within a bounding space, when the focal region is expanded, some regions must be reduced. We wish to scale each local region of the model but keep it similar to its original appearance. That is, we aim to preserve the aspect ratio of each local region, avoiding squeezing or squashing which would result in distortion. Specifically, we design an optimization procedure that allows cubes covering the model to undergo uniform scales, while letting empty cubes to be stretched to absorb the resulting distortion. Since the transformations of connected cubes are not identical, the deformation inevitably causes distortion. To minimize this resulting distortion, we propose a set of energy terms to form an optimization system and solve for the grid vertex positions of the deformed space in a least-squares sense.

Previous Work

Many related algorithms have been developed to visualize complicated information of 3D volumetric models. The outer opaque layer always overlays the internal information and results in visualization problems. Hence, Viola et al. [13] automatically compute the importance of each voxel to avoid hiding the important regions by the outer trivial voxels. Zhou et al. [15] advocate a feature-based method to enhance the volumetric features and render the parts of the model inside and outside the focal region in different styles. McGuffin et al. [11] applied deformation techniques to browse volumetric data. Their approach opens up, spreads apart, or peels away the outer layers to reveal the hidden structures.

To visualize tiny information, such as the bump surface of a human colon, in a clear and detailed view, researchers have proposed many methods to magnify the focal area and either distort or overlay the neighboring regions to highlight the region of interest. Keahey et al. [5–7] deformed the texts or 2D images by a transformation grid, determined by nonlinear magnification fields. Bier et al. [1] presented an intuitive interface for the user to specify the focal region and render the information inside the focal region with a different style to enhance the feature of interest. Carpendale et al. [2, 3] proposed several distortion patterns, such as stretch orthogonal and nonlinear radial, to demand more space for the focal region to achieve 3D distortion that is independent of the viewpoint. LaMar et al. [8] applied hardware acceleration to deform the rendered 2D images or 3D volumes. They accomplished the goal by dynamically computing the texture coordinates for the grid vertices of the applied mask and rendering the texture with the coordinates that are projected onto the homogeneous space to make the results desirable. Unlike the above-mentioned methods, Wang et al. [14] proposed an interactive technique to render volumetric models according to optical-lens theory. Their method simulates the ray direction that is determined by the position of the focal point and displays the expanded image within the magnifying lens. Both [8] and [14] provide different shapes of bounded lens for the user to magnify the regions of interest.

- *Yu-Shuen Wang and Tong-Yee Lee are with the Computer Graphics Group/Visual System Laboratory, Department of Computer Science and Information Engineering, National Cheng-Kung University, No. 1, Ta-Hsueh Road, Tainan 701, Taiwan, R.O.C., E-mail: braveheart@csie.ncku.edu.tw; tonylee@mail.ncku.edu.tw.*
- *Chiew-Lan Tai is with the Department of Computer Science & Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: taicl@cse.ust.hk*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

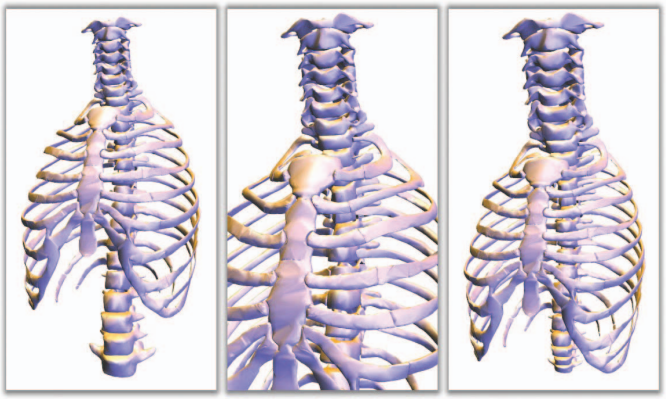

Fig. 1. (left) Original view of the thorax model. For a detailed observation of the cervical vertebra, an intuitive approach is to shorten the distance between the model and the camera. However, the other regions, such as the lower part of the spine, will be clipped off due to the limited screen space (middle). In contrast, our method magnifies the focal region while keeping the entire model displayed on the same screen (right).

Contribution

Some previous techniques are similar to our work in that they deform a grid space to magnify the region of interest. However, none of them has addressed the issue of stretching and distorting the remaining parts of the model while magnifying a specific region. While expanding the focal region, these methods simply let the distortion occur in the surrounding area and ignore the artifact. In contrast, our method lets the free space absorb the resulting distortion rather than letting the distortion uniformly spread throughout the nearby spaces (see Figure 3). We resize each local region of the given model by an approximate uniform scale such that the locally magnified model resembles the original shape, except that their local sizes are different. Furthermore, our method accomplishes Focus+Context visualization with the regions close to the focal region automatically expanded to reduce the distortion. This is a desirable property since these surrounding areas are likely to be the next focal region during interactive visualization and thus are important as well. Overall, our algorithm can visualize Focus+Context information on the screen while keeping the distortion under control and making it unnoticeable.

2 Focus+Context Visualization

Given an input model, we first rescale the model to be of unit size. To ensure that the entire model remains displayed on the screen, the given model should be placed at a proper position such that the eight vertices of its $1 \times 1 \times 1$ bounding box are all projected within the screen. Then, we partition the model using a n^3 uniform grid space, $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{C}\}$, with vertices $\mathbf{V}$, edges $\mathbf{E}$, and cubes $\mathbf{C}$, where $\mathbf{V} = [\mathbf{v}_0^T, \mathbf{v}_1^T, ...\mathbf{v}_{m-1}^T]^T$, $m = (n+1)^3$, and $\mathbf{v}_i \in \Re^3$ denotes the position of vertex i. In other words, there are n^3 identical cubes inside the bounding space.

Using our 2D graphical interface, the user moves a magnifying lens represented by a red dotted circle to specify a *focal region* to magnify. The user also specifies a parameter λ as the magnification factor of the focal region. We refer to the cubes covering the focal region (i.e., those whose centroids are projected within the red dotted circle) as the *focal cubes*. The focal cubes are expanded according to the parameter λ, and the remaining cubes are automatically expanded or reduced to obtain a deformed grid space $\mathbf{G}'$ while maintaining the global bounding size. Our system solves for the grid vertex positions by minimizing a set of energy functions retaining the aspect ratios of the cubes covering the given model as well as constraining the grid vertices to be within the bounding space. After deforming the embedding space, the model is reconstructed by computing each vertex position according to its mean value coordinates within the grid space [4]. Since the global bounding space remains unchanged during the magnification of the focal region (see Figure 4), and the distance between the model and the camera also remains the same, our system accomplishes the aim of Focus+Context visualization.

2.1 Space Deformation

We compute the deformed grid space by determining the scaling transformation of each cube. Clearly, since the cubes are connected and are not scaled by the same factor, it is impossible to magnify a specific region without causing distortion to other regions. Therefore we strive to minimize the deviation of each local transformation from a uniform scale to keep the resulting distortion under control and unnoticeable. To satisfy the above requirements, we propose several energy functions and formulate an optimization system to compute the deformed grid space.

2.1.1 Individual Cube Rescaling

Given a cube $\mathbf{c}_k$, we compute its deformed version $\mathbf{c}'_k = \mathbf{s}_k \mathbf{c}_k$, where $\mathbf{s}_k$ is a 3×3 uniform scaling matrix. To solve for the entire grid space in

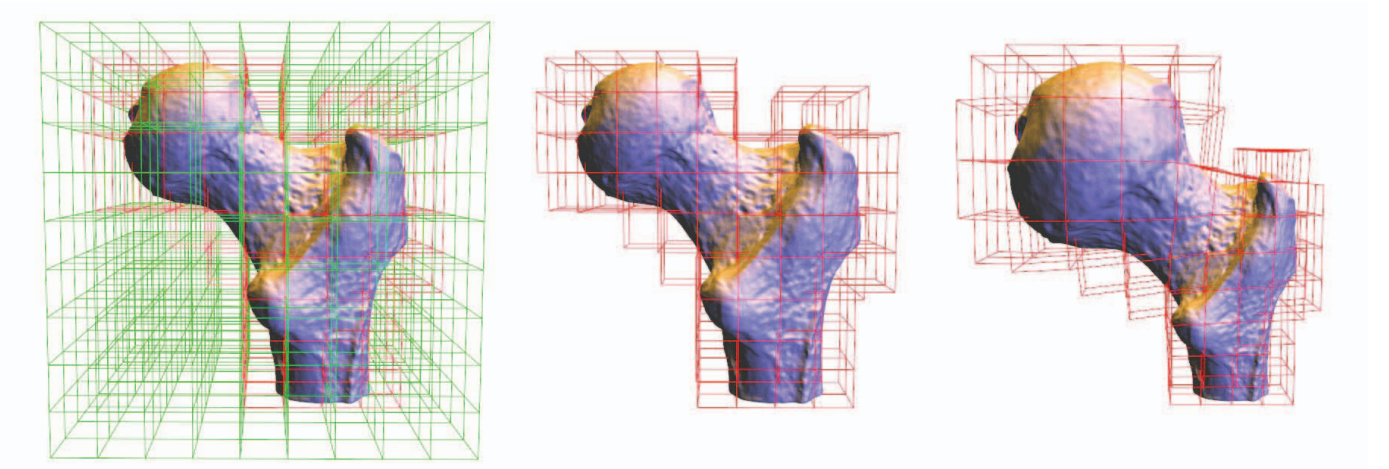

Fig. 2. (left) The original model partitioned with a uniform grid space. Since solving for those deformed cubes that do not contain any model information (in green) provides zero contribution to the magnification result, we omit their energy terms from our optimization system (middle). That is, we solve only for the deformed grid vertices of the cubes occupied by the model (in red) and reconstruct the embedded model by space interpolation to achieve magnification visualization (right).

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

Fig. 3. (left) Original model. The focal region (inside the red dotted circle) is magnified to observe in detail the teeth model using the stretch orthogonal method [2, 3] (middle left), radial Gaussian distortion [2, 3] with faster (middle) and slower (middle right) fall-off and our method (right). Note that, with the stretch orthogonal method and the radial Gaussian, the surroundings of the focal region are seriously stretched due to the distortion being uniformly distributed on the model; wider Gaussian fall-off only distributes the distortion to a larger region. In contrast, our method minimizes distortion and preserves the shape of the local feature such that each tooth remains similar to its original shape.

a least-squares sense, we integrate the equations into a linear system

$$\|\mathbf{C}' - \mathbf{S}\mathbf{C}\|^2 = 0, \quad where$$
$$\mathbf{S}_{uv} = \begin{cases} \mathbf{s}_k & if \quad u = v \\ 0 & otherwise \end{cases}, \ and \ \mathbf{C} = [c_0, c_1, ..., c_{n^3-1}]^T. \tag{1}$$

Rather than uniformly spreading the distortion over the grid space, we float distortion to the cubes that are not occupied by the model because the distortion of those cubes does not influence the magnification result. Thus, we classify the cubes into two groups: the *principle cubes* which cover the input model $\mathbf{C}_P \subset \mathbf{C}$ and the *trivial cubes* $\mathbf{C}_T \subset \mathbf{C}$ which do not. Our goal is to prevent the principle cubes from being squeezed. While solving for the deformed grid space by minimizing the objective function, our algorithm gives higher penalties to the principle cubes $\mathbf{C}_P$ if their transformations deviate from uniform scales so as to better preserve their aspect ratios. The trivial cubes $\mathbf{C}_T$ have zero penalties, sacrificing the uniformity of scaling transformation to absorb the distortion. To implement this idea, we rewrite Equation 1 into the form:

$$\|\mathbf{C}'_P - \mathbf{S}_P\mathbf{C}_P\|^2 + \gamma\|\mathbf{C}'_T - \mathbf{S}_T\mathbf{C}_T\|^2 = 0. \tag{2}$$

By setting $\gamma = 0$ since the distortion on the trivial cubes C_T does not influence the model's shape, our system computes only the principle cubes C_P by solving the following simplified equation:

$$\|\mathbf{C}'_P - \mathbf{S}_P\mathbf{C}_P\|^2 = 0. \tag{3}$$

Note that a cube is formed by a set of vertices and edges. That is, if a cube is uniformly resized, then its 12 edges would be uniformly expanded or contracted. While deforming the grid space, our algorithm retains the connectivity of the original grid structure and only moves the grid vertices to scale individual cubes to different sizes. Let $\mathbf{e}_p = \{i, j\}$ be the p^{th} edge of the cube $\mathbf{c}_k$ and $\varepsilon_p = \mathbf{v}_i - \mathbf{v}_j$, we can represent the cube as $\mathbf{c}_k^T = \mathbf{q}_k\mathbf{V}$, where

$$\mathbf{q}_{k,uv} = \begin{cases} 1 & if \quad u = p, \ v = i \\ -1 & if \quad u = p, \ v = j \\ 0 & otherwise \end{cases}, \quad \mathbf{c}_k = [\varepsilon_0^T, \varepsilon_1^T, ..., \varepsilon_{11}^T].$$

Putting the equations together, we can denote $\mathbf{Q} = [q_0^T, q_1^T, ..., q_{n^3-1}^T]^T$ and replace the matrix $\mathbf{C}$ by $\mathbf{QV}$. Thus, Equation 3 can be reformulated as

$$\|\mathbf{Q}_P\mathbf{V}'_P - \mathbf{S}_P\mathbf{Q}_P\mathbf{V}_P\|^2 = 0. \tag{4}$$

where $\mathbf{V}_P$ denotes the vertices of the principle cubes.

2.1.2 Position Constraint

To solve Equation 4, we need at least one absolute position to locate the deformed grid space since the equation only expresses relations among vertex positions. However, constraining only one vertex at a specific position may move the model when the focal region is magnified. This is because the optimization may transform individual cubes to different positions to reduce their distortion. Figure 5 demonstrates this effect. To enhance the stability of the visualization, we want all the cubes to be close to their original positions. Specifically, we constrain each grid vertex to be at its original position with a small weighting factor and introduce the following energy term:

$$\omega\|\mathbf{I}\mathbf{V}'_P - \mathbf{V}_P\|^2 = 0. \tag{5}$$

We set $\omega = 0.001$ in our experiments to retain the overall model's position. Larger ω will lead to immovable grid vertices and failure to deform the grid space.

2.1.3 Non-Linear Constrained Optimization

We solve for the deformed vertex positions by minimizing the integration of the above energy terms

$$arg\min_{\mathbf{V}'_P, \mathbf{S}_P} \quad \|\mathbf{Q}_P\mathbf{V}'_P - \mathbf{S}_P\mathbf{Q}_P\mathbf{V}_P\|^2 + \omega\|\mathbf{I}\mathbf{V}'_P - \mathbf{V}_P\|^2, \tag{6}$$

which can be formulated as an over-determined linear system $\mathbf{A}\mathbf{V}'_P = \mathbf{b}(\mathbf{V}_P)$, where $\mathbf{A} = [\mathbf{Q}_P^T, \omega I^T]^T$ and $\mathbf{b}(\mathbf{V}_P) = [\mathbf{S}_P^T\mathbf{Q}_P^T\mathbf{V}_P^T, \mathbf{V}_P^T]^T$. To prevent the global bounding space from expanding, we add inequality constraints to the objective function such that the grid vertices move only within the bounding space. That is, we minimize

$$arg\min_{\mathbf{V}'_P} \quad \|\mathbf{A}\mathbf{V}'_P - \mathbf{b}(\mathbf{V}_P)\|^2, \quad \text{subject to}$$
$$x_l \le \mathbf{v}_x \le x_u, \quad y_l \le \mathbf{v}_y \le y_u, \quad and \quad z_l \le \mathbf{v}_z \le z_u, \tag{7}$$

where x_l, x_u, y_l, y_u, z_l, and z_u are the lower and upper bounds of x, y, and z coordinate, respectively. Since $\mathbf{A}$ is not a positive definite matrix, we multiply the equation by $\mathbf{A}^T$ and solve the system $(\mathbf{A}^T\mathbf{A})\mathbf{V}'_P = \mathbf{A}^T\mathbf{b}(\mathbf{V}_P)$ to obtain the unknown variables $\mathbf{V}'_P$. Note that the uniform scaling transformations of the cubes are still unknown when we determine the deformed grid space. Therefore, we can only solve this non-linear optimization problem by iteratively updating the vertex positions [9].

The scaling factor for the focal cubes is obtained from the input parameter λ. For the remaining (non-focal principle) cubes, we compute their scaling transformations according to their deformed (M'_k) and original (M_k) volumes. Specifically, $\mathbf{s}_k = (M'_k/M_k)^{1/3}\mathbf{I}$. In the beginning, we set $\mathbf{V}_P^0 = \mathbf{V}_P$ to start the iteration. We then compute the scaling transformations for the non-focal cubes from $\mathbf{V}_P^t$, and use the obtained $\mathbf{b}(\mathbf{V}_P^t)$ to solve for the new vertex positions $\mathbf{V}_P^{t+1}$. Although the scaling transformation obtained from $\mathbf{V}_P^0$ is merely an identity matrix, the least-squares solver will still reduce the volumes of some cubes so as to expand the focal region. However, the set of vertex positions computed in the first iteration is not the optimum solution of our proposed objective function because the applied scales are only determined by the initial guess. Therefore, we repeat the process to update the vertex positions until the solution converges.

While iteratively updating the grid vertex positions, we detect whether any vertex violates the inequality constraints. Since only a

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

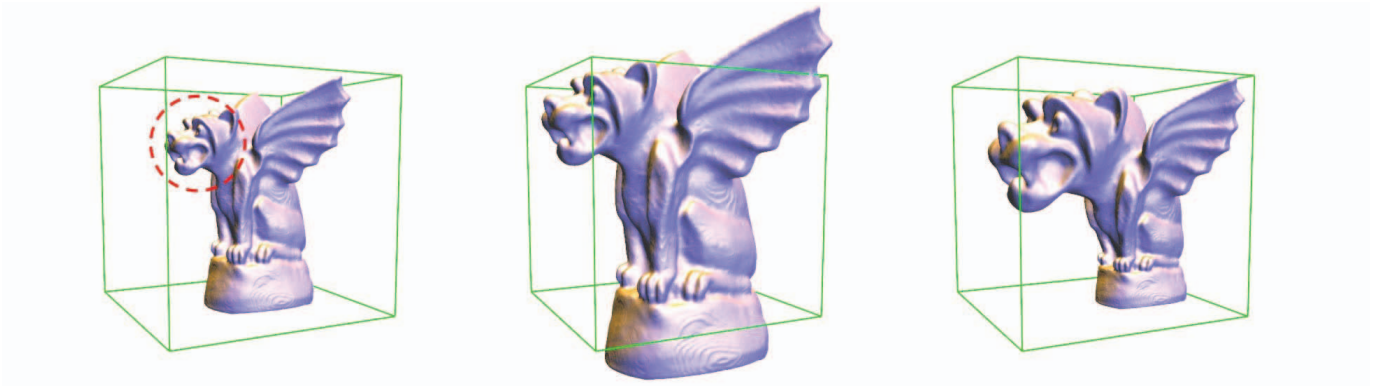

Fig. 4. (left) Original model and its global bounding space. While magnifying the gargoyle's head without imposing inequality constraints, the entire model is uniformly expanded and some regions may go out of the bounding space (middle). We constrain the outer grid vertices to be located on the bounding surface (right), and thus ensure that the deformed model is always displayed on screen in its entirety.

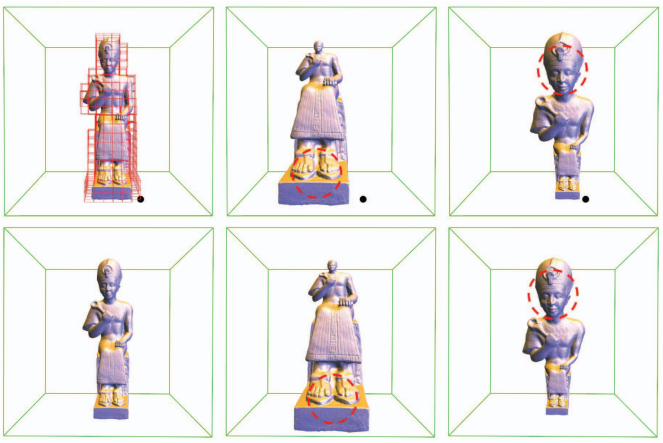

Fig. 5. (top row) Only one position constraint is applied at the bottom right of the Ramesses model (black point). While magnifying his feet (top middle) and head (top right), his body would move to the right or left as our algorithm tries to minimize distortion. (bottom row) All the grid vertices are slightly constrained to their original positions. The model's position is now more stable after local magnification.

few vertices might be transformed outside the bounding space, not all the inequality constraints have influence on the optimization problem. Therefore, we consider an inequality constraint $f_x(\mathbf{v})$ as

$$\begin{array}{llll} active & if & \mathbf{v}_x > x_u & or \quad \mathbf{v}_x < x_l \\ inactive & if & x_l \leq \mathbf{v}_x \leq x_u & \end{array}.$$

Similarly, $f_y(\mathbf{v})$ and $f_z(\mathbf{v})$ are inactive if $\mathbf{v}_y$ and $\mathbf{v}_z$ satisfy their respective constraints. In the beginning, we consider all the inequality constraints as inactive since all the grid vertices are located within the bounding space. After computing the new vertex positions in each iteration, our algorithm detects if any inequality constraint has become active, and if so, restricts that vertex to be sliding on the space boundary in the next iteration. To combine the inequality constraints with our proposed objective function, these constraints are transformed into energy terms [10] (for example, $(\mathbf{v}_x - x_l)^2 = 0$ if $\mathbf{v}_x < x_l$) and are added into the linear system if they are active. Note that the active constraints of x, y, and z coordinates are different; therefore, the three coordinates of the grid vertices should be solved separately. Let $\mathbf{F}_x = \{f_{x,0}, f_{x,1}, f_{x,2}, ...\}$ be the index set of active inequality constraints of the x coordinate. We can solve the linear system $\mathbf{A}_x \mathbf{V}'_{P,x} = \mathbf{b}_x(\mathbf{V}_P)$ to obtain the x coordinates of the grid vertices, where

$$\mathbf{A}_x = [\mathbf{Q}_P^T, \omega \mathbf{I}^T, \mathbf{R}_x^T]^T, \mathbf{b}_x(\mathbf{V}_P) = [\mathbf{S}_P^T \mathbf{Q}_P^T \mathbf{V}_{P,x}^T, \mathbf{V}_{P,x}^T, \mathbf{H}_x]^T,$$

$$\mathbf{R}_{x,uv} = \begin{cases} \delta & if\ v = f_{x,u} \in \mathbf{F}_x \\ 0 & otherwise \end{cases}, \mathbf{H}_x = \begin{cases} \delta x_l & if \quad \mathbf{v}_x < x_l \\ \delta x_u & if \quad \mathbf{v}_x > x_u \end{cases},$$

and δ is a large number to enforce the soft constraint (we set $\delta = 100$ in our experiments). The other two coordinates of the vertices can be determined in the same manner. Since the number of equations changes whenever any inequality constraint becomes active, causing the size and the structure of the linear system to change as well, our system needs to re-factorize the matrix $\mathbf{A}_x$ (or $\mathbf{A}_y$, $\mathbf{A}_z$) to compute the new vertex set. Fortunately, the factorization only takes place when an inequality constraint is activated or deactivated, which occurs very infrequently because only the boundary vertices of the principle cubes might violate the inequality constraints. Although our algorithm deforms the grid space in an iterative manner, the computation is still efficient because the factorization of the linear system is not necessary in every step. In most steps, we need only to apply back substitutions to compute the new vertex set and thus the grid space can be deformed in real time.

For grid vertices whose inequality constraints are active, we examine their positions to decide when to deactivate them. Since our system solves for the deformed grid space using soft constraints, the vertices with active inequality constraints are not exactly located on the grid space boundary, rather they may be slightly inside or outside the boundary. Specifically, the activated inequality constraint of an outside vertex pulls the vertex close to the space boundary, but the vertex remains outside. Then, subsequent movement of the focal region may cause the vertex to move in. When this happens, its inequality constraint can be deactivated.

2.1.4 Initial Guess

Solving a nonlinear optimization problem always needs a starting position, which is commonly called the initial guess. Obviously, an initial guess that is close to the optimum solution would lead to faster convergence. Choosing a good starting point is therefore an important issue. Although it is always possible to start deforming the grid space from its uniform shape, the extra iterations needed will increase the computation cost and thus slow down the interactive rate. Figure 6 shows an example to demonstrate the initial guess and the subsequent iterations. Here, we assume that the cursor movement of the user while specifying the focal region is continuous and the algorithm starts the iteration from the previous frame because the magnification results should be similar if their magnified regions are close to each other. In the beginning, we set the original grid space as the initial guess since there is no previous frame. Our system repeats the updating process until the movement of each grid vertex is less than 0.001, which takes 8 iterations on average to achieve the minimum solution.

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

Model	Model Vertex Number	Grid Vertex Number	Factorization (sec.)	Back substitution (sec.)	Reconstruction (sec.)
bonefoot	12612	605	0.032	0.001	0.002
ball joint	137062	343	0.021	0.001	0.023
Goddess	523578	565	0.106	0.001	0.087
thorax	99920	973	0.126	0.001	0.016
skull	63264	2714	0.145	0.001	0.011
Ramesses	826266	920	0.151	0.001	0.144
colon	44500	1351	0.101	0.001	0.008
hip	132538	1470	0.176	0.001	0.024
gargoyle	100002	1644	0.134	0.001	0.018
teeth	116604	1851	0.175	0.001	0.020
dancing children	100000	1697	0.131	0.001	0.018
Chinese dragon	437645	1480	0.177	0.001	0.076

Table 1. The second and third columns show the model information and the last three columns show the timing statistic. The computation cost of reconstructing the model depends on the model's vertex number while the computation cost of grid space deformation (i.e., factorization and back-substitution) depends on the grid's vertex number.

Fig. 6. We magnify the dragon's head (in the red dotted circle) to achieve Focus+Context visualization. From left to right are the original model and the locally magnified models computed by our algorithm after 1, 5, 10, 30 iterations. Notice that the surrounding region of the dragon's head is more distorted in the beginning but the distortion is rapidly reduced in subsequent iterations. The results obtained in 10 and 30 iterations are very similar because the iterative solver has converged.

2.2 Model Reconstruction

We reconstruct the given model by space interpolation after the grid space is deformed. Since each model vertex is embedded in a local cube, we can determine its new position by a combination of their respective 8 cube vertices. Let $\mathbf{u}$ be a vertex of the given model, and $\mathbf{v}_0 \sim \mathbf{v}_7$ be its surrounding cube vertices. The model vertex $\mathbf{u}$ can be represented as $\sum_{i=0}^{7} w_i \mathbf{v}_i$, where w_i is the weight determined by the mean value coordinates [4]. We compute the weighting factor w_i in the pre-computation step and use these values to reconstruct the model each time the bounding space is deformed since the w_i's remain the same.

3 Results and Discussion

We have implemented our algorithm on an Intel Core2 2.33 GHz PC with 2 Gb RAM. The linear system is solved with Cholesky Factorization provided by the Taucs library [12]. We partition each input model by a 20^3 grid space, except the ball joint model (Figure 2) by 10^3. Clearly, partitioning the model with a finer grid space (especially for complex models) will lead to better results but increase the computation cost. To balance between the quality and interactive rate, we found that a 20^3 grid space is a good compromise. Table 1 shows the timing statistic and the model information. The slowest part of our algorithm is the matrix factorization, which mainly depends on the number of grid cubes. The reconstruction of the given model is efficient because computing each vertex only requires a linear combination of the surrounding cube vertices. For example, the Ramesses model in Figure 5 contains 826,266 vertices, which requires only 0.144 seconds to determine the positions of all the model vertices.

Our system can provide Focus+Context information to the user, allowing the user to view detailed content of the focal region while maintaining the overall perception of the model. The results shown in Figure 9 demonstrate the effectiveness of our technique. For each input model, we expand different focal regions specified with a dotted red circle and the model is deformed smoothly to keep the deformed shape within the global bounding box. In addition, due to the distortion minimization in our formulation, our system not only retains the aspect ratios of non-focal regions but also results in the smooth magnification of the surroundings of the focal region. This means the user needs not carefully specifies the shape and size of the magnifying lens to fit the feature of interest, leading to an easy and intuitive interface.

3.1 Distortion Measurement

We determine the degree of similarity between the original and deformed models by considering the distortion of each local region. The distortion can be measured in terms of the amount of stretching of individual cubes because the embedded input model is deformed by space interpolation. Obviously, a deformed cube that is of the same shape, but different size, as the original cube suffers from zero distortion. Therefore, to determine the amount of distortion, we first scale each pair of original and deformed cubes to the same size and then accumulate the difference of their 12 edge vectors. Specifically, we compute the equation

$$\sum_{\{i,j\}\in E_k} |\delta(\mathbf{v}_i - \mathbf{v}_j) - (\mathbf{v}'_i - \mathbf{v}'_j)|^2, \tag{8}$$

where E_k denotes the 12 edges of the local cube k, and $\delta = (M'_k/M_k)^{1/3}$, with M'_k and M_k denoting the volume of its original and deformed cubes, respectively. The region with lower distortion value means that it is similar to its original shape and the region with higher distortion value is naturally stretched. For example, in Figure 7, we expand a portion of the colon and measure the resulting distortion due to the magnification. It can be clearly observed that, without distortion minimization, the regions surrounding the focal region are seriously distorted. In comparison, our system produces results with much smaller and unnoticeable distortion. In the case when all the

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

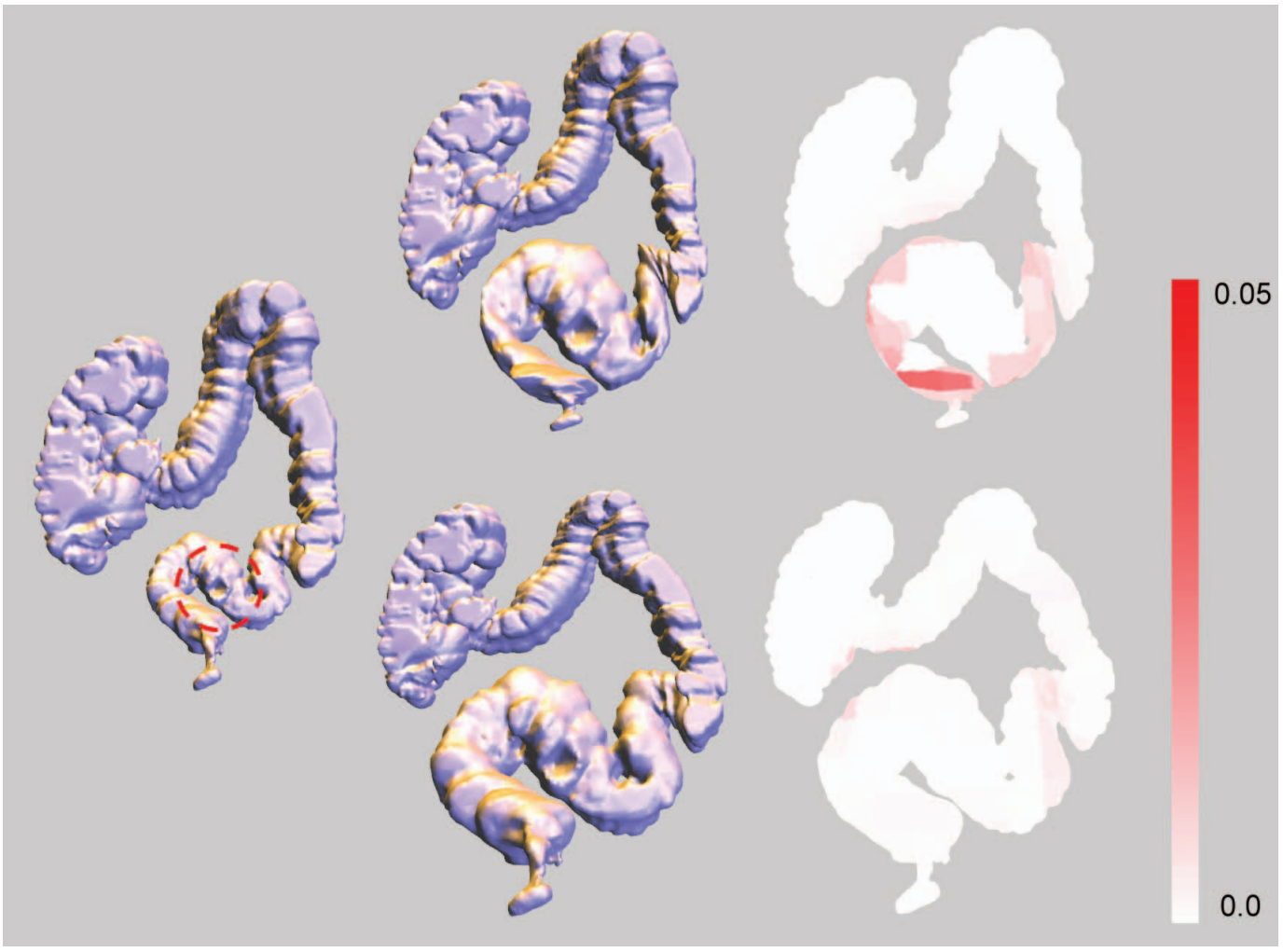

Fig. 7. (left) Original model. (middle) Deformed models by radial Gaussian distortion (top) [2, 3] and by our method (bottom). We use different shades of red to represent the distortion distribution (right). Notice that the distortion is much smaller with our method and does not gather around the focal region.

cubes are principle cubes, our system produces results similar to those with radial Gaussian distortion [2, 3] since there is no space to absorb distortion. Under this situation, the distortion is uniformly propagated outward from the focal region to other regions.

3.2 Soft Constraint

An interesting feature of our system is the use of soft constraints to solve for the deformed grid space in a least-squares sense. This leads to the actual size of the deformed focal region being slightly smaller than the user-specified magnifying factor for the focal region. We take the gargoyle model in Figure 4 as an example to illustrate this situation. Although we apply a $2 \times \mathbf{I}$ transformation on the focal region to determine the deformed grid space, the average volume of the deformed focal region is 1.46×10^{-4} while their original volume is 0.26×10^{-4}. The expanding ratio, i.e., 6.35, is less than the expected 2^3 because the uniformity and the magnification requirements conflict with each other (i.e., within the limited space, it is impossible for the focal region to expand with no distortion), and the least-squares solver finds the optimum solution to satisfy the overall requirements.

Solving for the deformed grid space using soft constraints inevitable causes a little distortion to the focal region because the system compromises the uniformity of the focal cubes to reduce the distortion of the overall model. Some applications may require the focal region to be expanded without distortion. Minimizing the objective function with hard constraints can achieve this aim and ensure that the size of the deformed focal cube is equal to the user's specification. Unfortunately, this approach would rapidly increase the cost if the focal region changes constantly since the modification of the hard constraints (i.e., focal region) changes the structure of the linear system and thus requires matrix refactorization. For efficiency, we solve the objective function with only soft constraints while the user is changing the focal region interactively and change the soft constraints into hard constraints to exactly retain the size and uniformity of the focal cubes after the user's specification remains fixed for some time.

3.3 Pros and Cons

Our system deforms the entire model to achieve Focus+Context visualization. While magnifying the focal region to display it with higher resolution, some regions are expanded while some are reduced to minimize distortion of every local region. Unlike the radial Gaussian distortion method [2, 3], which deforms only the cubes inside or close to the focal region, our system rescales almost all the cubes covering the model. Although the deformation is everywhere, the user can easily observe the continuous variation of the model's shape because the model is deformed interactively. Since every local region is rescaled, our method can not retain the model's global shape while minimizing the local distortion. For example, in Figure 8, the global shape of the foot bone model is changed with our approach but is better preserved by the radial Gaussian distortion. We argue that the change of the global shape is not necessarily a bad feature. According to the the-

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

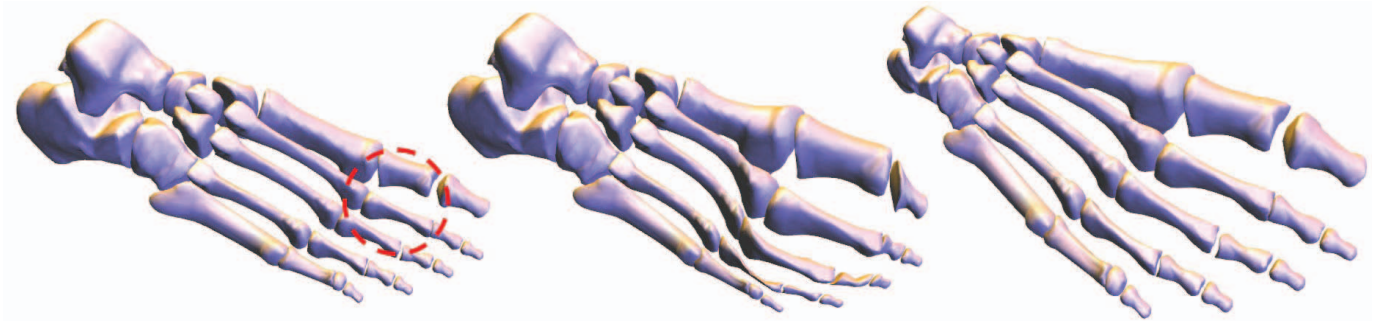

Fig. 8. Original model (left). Observe that the Gaussian distortion method (middle) better preserves the global shape of the foot bone than our method (right). However, the user usually pays more attention to the regions inside and surrounding the magnifying lens and our method produces very little artifact in these regions.

ory of human central vision, only a small area in the center of retina contains a rich collection of cone cells, which is sensitive to light, fine detail and color. Therefore, users usually concentrate on a region of interest and its surroundings when using a magnifying lens. This means that the model's overall shape is only a concept and the variation of the global shape is not disturbing to the user. In addition, since our system achieves real time performance, the user can easily move the magnifying lens to any free space not occupied by the model any time to see the model's undeformed global shape.

Our system offers another advantage for Focus+Context visualization. The magnifying lens is of limited size (i.e., smaller than the screen size), and during interactive visualization, the user is often also interested in the non-focal regions close to the lens border. Thanks to the distortion minimization, these surrounding regions, which are likely to be the next focal regions, are also enforced to expand because the cubes are connected.

4 Conclusions

We introduce a novel and interactive technique to achieve Focus+Context visualization of 3D models. The main contribution of our method is to minimize the resulting distortion so that each local region appears similar to its original counterpart. Our algorithm retains the original global size of the model by constraining the grid vertices to move within the global bounding space and prevents the local regions from squeezing through approximating the applied transformations to be close to uniform scales. Although solving this nonlinear constrained optimization problem takes several iterations to obtain the optimum solution, the computation is still efficient because updating the positions of grid vertices requires only back substitutions in most of the iterative steps. In addition, our algorithm has the potential to handle point models and volumetric data since the embedded model is deformed by space interpolation. While the cubes are transformed to satisfy the local magnification requirement, the positions of the point clouds or the voxels are recomputed as the combination of its surrounding cube vertices.

Acknowledgements

We thank the anonymous reviewers for their insightful comments that helps us improve the paper. We also thank AIM@SHAPE Shape Repository, Stanford 3D Scanning Repository and Cyberware for the 3D polyhedral models used in this paper. This work is supported in part by the Landmark Program of the NCKU Top University Project (Contract B0008), the National Science Council (Contracts NSC-95-2221-E-006-193-MY2 and NSC-96-2628-E-006-200-MY3), Taiwan, Republic of China and the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: 620107).

References

[1] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.

[2] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, page 46. IEEE Computer Society, 1996.

[3] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Extending distortion viewing from 2d to 3d. *IEEE Comput. Graph. Appl.*, 17(4):42–51, 1997.

[4] T. Ju, S. Schaefer, and J. Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3):561–566, 2005.

[5] T. A. Keahey. The generalized detail-in-context problem. In *INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization*, pages 44–51. IEEE Computer Society, 1998.

[6] T. A. Keahey and E. L. Robertson. Techniques for non-linear magnification transformations. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, page 38. IEEE Computer Society, 1996.

[7] T. A. Keahey and E. L. Robertson. Nonlinear magnification fields. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, page 51. IEEE Computer Society, 1997.

[8] E. LaMar, B. Hamann, and K. I. Joy. A magnification lens for interactive volume visualization. In *PG '01: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 223. IEEE Computer Society, 2001.

[9] K. Madsen, H. Nielsen, and O. Tingleff. Methods for nonlinear least squares problems. *Tech. rep., Informatics and Mathematical Modelling*, 2004.

[10] K. Madsen, H. Nielsen, and O. Tingleff. Optimization with constraints. *Tech. rep., Informatics and Mathematical Modelling*, 2004.

[11] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 53. IEEE Computer Society, 2003.

[12] S. Toledo, D. Chen, and V. Rotkin. Taucs: A library of sparse linear solvers, version 2.2. *Tel-Aviv University, Available online at http://www.tau.ac.il/ stoledo/taucs/*.

[13] I. Viola, A. Kanitsar, and M. E. Groller. Importance-driven volume rendering. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 139–146. IEEE Computer Society, 2004.

[14] L. Wang, Y. Zhao, K. Mueller, and A. E. Kaufman. The magic volume lens: An interactive focus+context technique for volume rendering. In *IEEE Visualization*, page 47, 2005.

[15] J. Zhou, M. Hinz, and K. D. Tonnies. Focal region-guided feature-based volume rendering. In *3DPVT*, pages 87–90. IEEE Computer Society, 2002.

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.

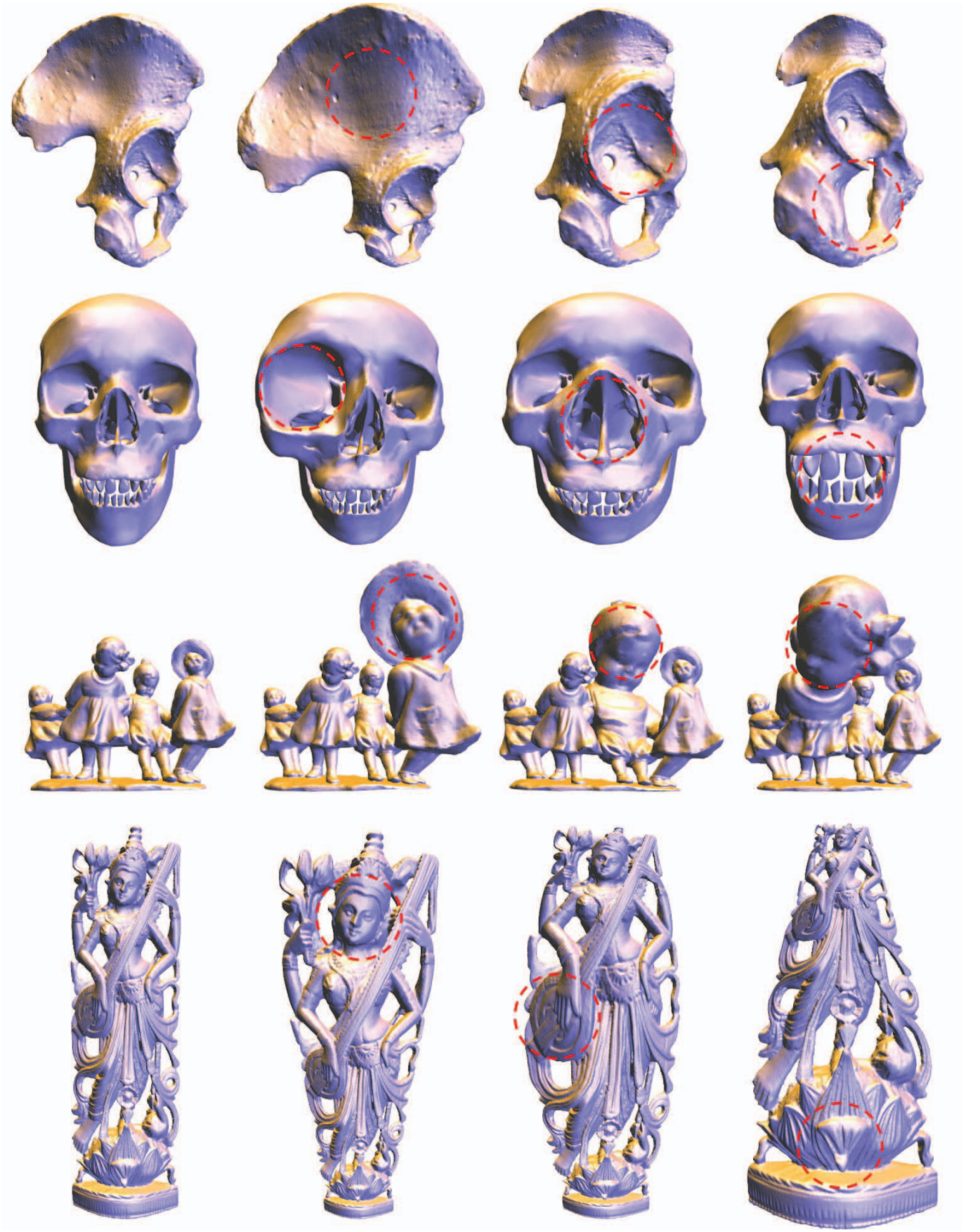

Fig. 9. Original models (leftmost column). Our algorithm expands the focal region and reconstructs the embedded model from the deformed grid space to achieve Focus+Context visualization (the remaining columns).

Authorized licensed use limited to: IEEE Xplore. Downloaded on November 3, 2008 at 00:27 from IEEE Xplore. Restrictions apply.