Distance Visualization for Interactive 3D Implant Planning

Christian Dick, Rainer Burgkart, and Rüdiger Westermann

Fig. 1. Distance visualization for interactive (> 30 fps) preoperative implant planning using color coding on the implant surface (left), oriented distance glyphs (middle), and color coding on the implant surface and on a set of additional axial slices (right).

Abstract—An instant and quantitative assessment of spatial distances between two objects plays an important role in interactive applications such as virtual model assembly, medical operation planning, or computational steering. While some research has been done on the development of distance-based measures between two objects, only very few attempts have been reported to visualize such measures in interactive scenarios. In this paper we present two different approaches for this purpose, and we investigate the effectiveness of these approaches for intuitive 3D implant positioning in a medical operation planning system. The first approach uses cylindrical glyphs to depict distances, which smoothly adapt their shape and color to changing distances when the objects are moved. This approach computes distances directly on the polygonal object representations by means of ray/triangle mesh intersection. The second approach introduces a set of slices as additional geometric structures, and uses color coding on surfaces to indicate distances. This approach obtains distances from a precomputed distance field of each object. The major findings of the performed user study indicate that a visualization that can facilitate an instant and quantitative analysis of distances between two objects in interactive 3D scenarios is demanding, yet can be achieved by including additional monocular cues into the visualization.

Index Terms—Distance visualization, biomedical visualization, implant planning, glyphs, distance fields.

1 INTRODUCTION AND RELATED WORK

Whenever a user in an interactive computer environment is faced with the problem of positioning an object relative to another object, additional cues are required to support an instant communication of the absolute physical distance of the controlled object to the other object. Even though a number of such cues can be used in general, for example, audio cues or haptic feedback, in most environments only binocular or monoscopic views and mouse-based interaction is available. Furthermore, as for spatial stimuli the visual sense predominates, followed by audition and touch [19], binocular or monocular visual cues are usually the preferred indicators.

Psychovisual experiments, however, have indicated that stereopsis provides only relative depth information, which allows inferring on the location of an object relative to another object rather than the ab-

- Christian Dick and R\u00eddidger Westermann are with the Computer Graphics and Visualization Group, Technische Universit\u00e4t M\u00fcnchen, Germany, E-mail: {dick, westermann}@tum.de.
- Rainer Burgkart is with the Klinik für Orthopädie u. Unfallchirurgie am Klinikum Rechts der Isar, Technische Universität München, Germany, E-mail: burgkart@tum.de.

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.

solute distance between them. Monocular cues, such as motion effects or additional geometric structures embedded into a visualization, on the other hand, can effectively indicate absolute estimates of distance. Without such direct cues, the estimation of absolute spatial distance between objects becomes very difficult, and typically requires a mental indirection to infer an accurate estimate. Let us refer to the textbooks by Steinman et al. [16] and Schwartz [14] for a scientific review of these arguments.

An example where color is used to indicate distance is given in Figure 1 (left): The minimum distances between points on an implant and the surface of a bone into which the implant should be placed are color-coded on the implant. In previous work, color (and iso-contours in color distributions) has been shown to be quite effective at indicating where distance falls below a critical value, and in combination with color legends distance-based coloring even supports quantitative investigations [2, 8, 18]. As can be seen, however, such visualizations fail in communicating contextual information that is required to effectively interpret distance. Regardless the specific distance measure used, such as shortest distances via distance transforms [5, 7, 12], spatial distance always 'connects' two reference points, and the human is used to interpret distance in the context of such references. Due to this observation, distance coding in still images via connecting structures such as lines or glyphs has been shown to be very effective. Pang et al. [10] used line primitives of varying length to indicate distances between two surfaces, and in [11, 12] point-to-point distances between two selected anatomical features were visualized via arrow glyphs ac-

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

companied by textual annotations. Without such connecting structures the user has difficulties to associate the points the color is referring to and to use her experience for interpretation.

Apart from the question for an adequate visual encoding of absolute distances in still images, another aspect becomes very important in interactive environments where the user's eyes fixate on the controlled objects and track their motion and orientation. In this case, distances to other objects have to be communicated via peripheral vision, which is the ability to gather information from the environment other than the point of focus. Peripheral vision directs our attention to slight movements in the environment around us, and it is processed significantly faster than vision requiring color. To the best of our knowledge, an investigation of the potential of glyphs for revealing distances between two moving objects in interactive scenarios—taking into account the possibility to dynamically change their shape, size, color, and density— has not been performed so far.

Based on the aforementioned observations, in this paper we provide novel monocular cues for effectively revealing point-to-point distances between two objects in interactive environments. Based on the mentioned previous techniques using static lines and glyphs we propose a number of extensions to exploit the human's peripheral vision, and we investigate the effectiveness of these extensions in an applicationspecific user study. Specifically, we address an application in which the user has to position an object with respect to another object that is fixed in space. The underlying application is implant planning for hip joint replacements, where in a preoperative planning phase a surgeon tries to find the patient-specific optimal implant shape, size and position. In this application, the user can interactively translate and rotate both models together or only the implant relative to the bone.

The monocular cues we present indicate distance via additional geometric structures that smoothly adapt their shape, appearance, or position to changing distance distributions. Two different kinds of monocular cues are provided: For interactive scenarios we have developed dynamically changing and colored distance glyphs to effectively employ peripheral vision for gathering distance information. An example demonstrating the visual sensation produced by these glyphs is given in Figure 1 (middle). Note that even in a still image the absolute distances can be revealed much more intuitively compared to pure color coding as in Figure 1 (left). One important reason is that the glyphs serve as 'bridging' structures between the implant and the bone surface, enabling the user to associate the distance values to geometric points on the surfaces.

For still images, where peripheral vision is not relevant for perceiving distances, we embed slices into the 3D visualization, and we further augment these slices by colors indicating distance (see Figure 1, right). The advantage of this kind of visualization is that it allows for a better observation of the implant due to the sparsity of the used geometric elements, yet providing visual indicators of the spatial relationships between the two objects.

For both types of visual cues, by carefully selecting the shape and appearance attributes of the used graphical primitives, critical regions where the moved implant comes close to the bone surface are identified pre-attentively, i.e., they are detected very rapidly and accurately by the low-level visual system. The low-level visual system is associated with the lateral geniculate nucleus, or simply called visual thalamus, which is part of the thalamus and responsible for reducing the optical input that is further processed by the visual cortex. In our application, the pre-attentive visual features we are exploiting to accommodate a rapid analysis of distances are hue, form, and density of glyphs. For a more elaborate discussion of the pre-attentive mechanisms of the low-level visual system let us refer to [6, 17].

Throughout this paper we will present the methodology underlying the two new approaches for visualizing spatial distances, and we will analyze the particular strengths as well as the differences and limitations of these approaches in the context of preoperative implant planning. To the best of our knowledge, this is the first time that spatial distance visualization between two objects in interactive environments is addressed other than by color.

The remainder of the paper is as follows: In the next section we



Fig. 2. Cross section through a human femur, showing the bone shell consisting of stiff cortical bone and the bone interior consisting of spongious trabecular bone.

present the medical application for which our distance visualization techniques are developed. Section 3 outlines the specific distance measures we use and describes how these measures are mapped onto meaningful graphical representations, such as glyphs and slices. The dedicated GPU implementation of all approaches, by which it is possible to achieve interactive frame rates even for large and complicated data sets, is discussed in Section 4. The result section demonstrates the applications of our techniques and compares the results to each other. We conclude the paper with a summary of the basic advantages and disadvantages of both techniques.

2 MEDICAL BACKGROUND

A human femur consists of two types of tissue: The outer shell of the bone consists of stiff cortical bone, whereas the interior consists of spongious trabecular bone (see Figure 2). This particular construction results from a natural optimization process, which allows the femur to bear maximum load at minimum weight. For total hip joint replacement, the femoral head is removed, and the inner trabecular bone has to be partially removed in order to be able to insert the implant.

During preoperative planning, the patient-specific optimal implant shape, size, and position has to be determined according to certain geometrical as well as biomechanical criteria. The geometrical criteria include that the implant has to fit into the interior of the cortical bone shell, without penetrating this shell, and that the joint rotation center has to be preserved. The main biomechanical criterion is related to the stress distribution in the loaded femur, in that the stress distribution after insertion of the implant should closely match the preoperative, physiological stress distribution. Bone is a living tissue and adapts to changes of the mechanical load situation by bone formation or resorption, dependent on whether the load has increased or decreased, respectively. Paradoxically, bone resorption also occurs if the load exceeds a certain critical magnitude. The insertion of an implant changes the stress distribution in the bone, in that stresses are bypassed by the implant, which leads to a removal of stresses from certain regions of the bone (so-called stress shielding).

In addition, for an optimal load transmission from the implant to the adjacent bone stock, it is important that the implant has tight press fit contact to the remaining bone shell, in particular in the calcar region and—especially for short stemmed implant designs—the lateral wall of the femur. If the implant has not sufficient contact in these regions, the stress distribution is changed significantly in that stresses are moved almost completely from regions with no contact to regions with contact. An unphysiological stress distribution can lead to a substantial resorption of bone tissue, with the consequences of fracture or loosening of the implant. Therefore, the selection of a patient-specific optimal implant shape, size and position during preoperative planning is essential for the long-term stability of the implant.

In previous works, Dick and coworkers presented a computational



Fig. 3. Distance rendering using oriented glyphs, located at certain seed points on the implant surface: Left: Orienting the glyphs orthogonally to the implant surface enables an effective perception of the spatial relationships. Right: Orienting the glyphs toward the respective closest point on the bone surface makes perception of the spatial relationships rather difficult.

steering environment for implant planning in orthopedics [3, 4], which computes and visualizes the stress distribution in the femur at interactive update rates, and thus allows the surgeon to interactively analyze the effect of different implant shapes, sizes, and positions on the stress distribution in the patient-specific bone. This environment so far uses semi-transparent rendering of the bone and implant surfaces to visualize the implant position three-dimensionally within the bone, which only allows for a very limited perception of the geometric relationships and absolute distances, i.e., the distances between bone and implant.

In the considered application, however, the perception of the geometric relationships and absolute distances is highly important for a precise and intuitive navigation of the implant in the virtual 3D environment, for validating if the current implant configuration meets the described geometrical criteria, as well as for analyzing if a particular unphysiological stress distribution results from the implant not having sufficient contact with the bone, and in this case, how the implant has to be moved or changed in size in order to establish this contact. Therefore, the goal of the work presented in this paper is to develop dedicated visualization methods that allow for a rapid perception of the geometric relationships and absolute distances between implant and bone. In the computational steering environment, the surgeon can then flexibly switch between the distance and the stress visualization (or is provided with both visualizations simultaneously on a split screen) in order to find the patient-specific optimal implant shape, size, and position.

3 DISTANCE VISUALIZATION

For the particular application of preoperative implant planning, we employ a triangle mesh of the outer bone surface, a triangle mesh enclosing the trabecular region of the bone and thus separating the cortical and trabecular regions, and a triangle mesh of the implant surface. The first two meshes, which are in the following referred to as the outer and inner bone mesh, are obtained by segmentation from a CT scan. The distances which are of primary relevance for preoperative planning are those between the inner bone mesh and the implant mesh. To simplify the notation, we typically omit the differentiation between inner and outer bone mesh in the following.

3.1 Glyph-based Distance Visualization

Our first approach for the visualization of distances between bone and implant mesh is based on rendering cylindrical glyphs at selected seed points on the implant surface.

We use cylindrical glyphs since these provide an intuitive means to depict distances: The height of the cylinder visualizes the represented distance, and its axis specifies the direction along which the distance



Fig. 4. Distance rendering using oriented glyphs: Left: Rendering glyphs for all distances leads to visual cluttering. Right: By rendering glyphs only for distances up to a certain maximum value, visual cluttering can be avoided.

is measured. There are distinct strategies to choose the radius of the cylinder: For example, by using a radius proportional to the cylinder's height, the glyph is scaled isotropically with a scaling factor proportional to distance. In contrast, by using a radius inverse proportional to the height (i.e., the product of radius and height is constant), the radius is increasing if the distance is decreasing, and vice versa. This suggests that the glyphs are clamped between the two surfaces, and that they are elastically deforming according to the relative movements of the surfaces, i.e., the glyphs are squeezed with decreasing distances. We use the latter strategy due to this intuitive—since physically plausible—relation between changes of distances and effects on the glyphs. In addition, for the considered medical application, small distance values are of high relevance. By increasing the radius with decreasing height, it is ensured that the glyphs representing small distances remain visible.

For the glyph-based approach, distances between the two surfaces are measured orthogonally to the implant surface and are determined by means of ray/triangle mesh intersection. By orienting the glyphs orthogonally to the implant surface, their orientations are changing smoothly with the implant surface, which enables an effective perception of the spatial relationships. In contrast, orienting the glyphs toward the respective closest point on the bone surface leads to abrupt changes of the glyphs' orientations, which introduces visual clutter and makes perception of the spatial relationships rather difficult (see Figure 3).

Since large distances are of minor relevance, we only render glyphs representing distances below a certain maximum value d_{max} . In this way, we also avoid visual cluttering of the glyphs (see Figure 4). To avoid that glyphs suddenly pop up or disappear when the represented distance crosses d_{max} , we fade out the glyphs by rendering them as semi-transparent objects. In particular, we linearly interpolate a glyph's alpha value from 1 to 0 for distances from $d_{max}/2$ to d_{max} . Furthermore, to avoid that the glyphs' radii become arbitrary large, we clamp the radii at a certain maximum value r_{max} . To ensure that glyphs do not overlap, we prescribe a minimum distance between each pair of seed points of $\delta = 2r_{max}$.

We only render glyphs on those parts of the implant surface which are lying in the interior of the bone. To avoid that the glyphs are hidden by the bone surface, we render the bone meshes semi-transparently. If the implant sticks out of the bone surface, the implant is in a geometrically invalid position. Therefore, at the transition from bone interior to exterior and vice versa, we abruptly remove the glyphs (no fading out) to attract the attention of the user. In addition, we colorize the parts of the implant surface which are lying outside of the bone mesh using blue color, which is in high contrast to the other colors used for the bone and implant surfaces and the glyphs.



Fig. 5. Left: Distance measure (red) at points (yellow) in the volume between bone and implant (for simplicity, the 2D case is shown). Right: Approximation of this distance measure using distance fields (blue: bone distance field, magenta: implant distance field).

Besides of adapting the shape of the glyphs, we additionally colorize the glyphs to encode the represented distance. We linearly interpolate between colors green and red, where green color is used for a distance of d_{max} , and the highlighting red color (corresponding to the importance of small distances) for a distance of 0. This smoothly varying color map was chosen to avoid a rapid change of colors when the glyphs' sizes are changing, which would otherwise disturb the perception of size changes.

The glyph seed points are computed once when a particular implant mesh is selected. In particular, we use spatially fixed seed points and a uniform seeding density. Besides the important goal of achieving visual coherence, these choices are motivated as follows: By using spatially fixed seed points, we avoid that changes of visualized distances due to moving of the implant are overlaid by changes of visualized distances due to moving of the seed points, which would distract from precisely monitoring the implant's position. By using a fixed seeding density, increasing distances (corresponding to decreasing radii of the glyphs) lead to a visual thinning out of the glyphs. For the considered application, this is in accordance with the fact that large distances are of minor interest.

To determine the seed points, we proceed as follows: In the first step, we build a list of potential seed points L' by iterating over the individual triangles of the implant surface mesh. For each triangle, we test if all edge lengths are at most δ . If this is the case, we add the triangle's barycenter to L'. Otherwise, we perform a 1:4 split of the triangle, and proceed recursively on the newly created triangles. In the second step, we build the list of seed points L by iterating over L'. For each potential seed point P, we test if all seed points already contained in L have at least a distance of δ from P (measured in 3D space). If this is the case, P is added to L, otherwise it is discarded. Since this algorithm considers the individual triangles of the implant mesh separately, it is very robust in that it is independent of the mesh quality.

3.2 Slice-based Distance Visualization

One approach for the visualization of the distances between two surfaces is to display at each surface point the distance to the nearest point on the respective other surface via color coding [2, 8, 18]. This can be realized by using a 3D distance field [7] for each of the two surfaces. Given a set of feature points $\Sigma \subset \mathbb{R}^3$, a distance field is a mapping $d_{\Sigma} : \mathbb{R}^3 \to \mathbb{R}$ which specifies for each point *x* in three-dimensional space the distance to the nearest feature point: $d_{\Sigma}(x) = \inf_{y \in \Sigma} ||y - x||_2$. A distance field for a particular surface is obtained by choosing Σ to be the set of points on this surface. To create the described visualization, for the first surface, the second surface's distance field is sampled at the points on the first surface, and vice versa. Algorithmically, a distance field can be obtained by means of a distance transform [7].

For our application of interactive 3D implant planning, this visualization approach is shown in Figure 1 (left). Here, the distance volume of the bone surface is visualized on the surface of the implant. To provide a large number of distinguishable colors for conveying absolute distance values effectively, a rainbow color map ranging from red (small distance) to violet (large distance) is used. This choice was motivated by our observation that the known shortcomings of rainbow color maps, as for instance discussed in [1], are much less severe in



Fig. 6. Focus+context visualization: Left: To avoid that colors are washed out by the bone surface, the surface's opacity is reduced. However, this reduces the spatial context information. Right: By using a focus+context approach, the colors are saturated in the important regions, and at the same time the spatial context information is preserved.

our application scenario. Since here the user's perception is supported by 3D geometry that provides additional depth cues, and the color variation on the implant changes dynamically according to the issued movements, the rainbow color map can convey a very accurate and intuitive image of absolute distances.

Figure 1 (left), however, demonstrates that the spatial relationships between the nested meshes along the viewing direction cannot be recognized. As a consequence, it is not intuitively obvious how the implant has to be moved in order to optimize the distances between bone and implant with respect to the specific medical requirements, which makes navigation in an interactive setting rather difficult.

To address this issue, we propose to augment the visualization by rendering additional geometric structures that bridge the volume between implant and bone. Since slicing planes are still the predominant tool for analyzing 3D data in medical diagnosis, and thus are graphical elements to which doctors are accustomed through experience, we render a set of parallel, axial slices passing through the bone and implant, as shown in Figure 7 (left). From the intersection pattern of the slices with the bone and implant the spatial relationships become directly visible. By using an orthographic projection, leading to a parallel rendering of the slices in screen space, occlusions introduced by the slices can be kept at a minimum.

However, it is still difficult to relate the color-coded distances on the implant to the spatial relationships. This can be improved by visualizing color-coded distances also on the slices. Rendering the distance field of the bone not only on the implant but also on the slices results in ring-shaped structures, as illustrated in Figure 7 (middle left), with the number of rings between implant and bone being a visual clue for the distance. However, on the slices the color coding then only provides the distance of the respective point to the bone, without incorporating the distance to the implant. In particular, the colors on the slices do not change upon movements of the implant, rendering this approach as rather non-intuitive in an interactive setting.

To provide the surgeon with a direct visualization of the distances between bone and implant on the slices, we first have to define a reasonable distance measure at the points in the volume between implant and bone. Considering a point *x*, a reasonable choice would be the minimum length of all line segments passing through *x* and connecting a point on the implant and bone surface, respectively (see Figure 5). On the bone and implant surfaces, this definition reduces to the distance to the nearest point on the respective other surface. In case of parallel or concentric surfaces (as is the case in the current application), this measure can be well approximated by $d_{Bone}(x) + d_{Implant}(x)$, where d_{Bone} and $d_{Implant}$ denote the distance fields of the bone and implant, respectively. These distance fields have to be computed only



Fig. 7. Color coding of distances on the slices: Left: No color coding of distances. Middle left: Distance to the bone. Middle right: Sum of the distance to the bone and the distance to the implant. Right: Infinitely thin slices without contours.

once when the respective mesh is changed. The result is illustrated in Figure 7 (middle right).

To achieve a high-quality visualization, we render slices of a certain thickness (1mm) instead of infinitely thin slices (as is illustrated in Figure 7 (right) for comparison). If the view direction is falling into the slice plane, infinitely thin slices would become invisible. Furthermore, we render black contours along the intersection lines of the slices with the bone and the implant. In this way, partially overlapping projections of slices can be clearly distinguished.

To color-code the distances, we employ a rainbow color scale which is obtained from the HSV color space, using a hue from 0° = red to 270° = violet. The red end of the scale (highlighting color) is used to encode the distance value 0, the violet end to encode a certain maximum value d_{max} . Analogous to the glyph-based approach, to clearly indicate when the implant has left the interior of the bone, i.e., a geometrically invalid positioning of the implant, the parts of the implant which have left the interior are rendered in a distinct color. Here, we use white color, which is in high contrast to the red color used to encode the smallest distances. In this way geometrically invalid positions are immediately visible.

For the 3D visualization, we use semi-transparent rendering of the bone surfaces. To hide non-important regions with large distances between bone and implant from the visualization, we use a focus+context approach (see Figure 6) by scaling the opacity of the fragments of the *outer* bone surface with the factor $(saturate(d_{Implant}(x)/r))^2$, where *r* denotes the radius of the focus region, and *x* is the world space position of a fragment. This leads to high and low opacities of the outer bone surface in regions with large and small distances between bone and implant, respectively. In this way, the important regions with small distances between bone and implant, especially the calcar region and the lateral femoral wall are clearly visible, and at the same time, the perception of the outer shape of the bone and thus of the spatial relationships is improved.

Additionally, we provide the possibility to render the implant as a semi-transparent surface. This enables the surgeon to monitor the distances between bone and implant also on the back side of the implant, which are otherwise hidden by an opaque implant (see Figure 11, right, and Figure 12, right). Since the user can rotate the entire scene consisting of bone and implant, a clear view on the relevant parts can even be obtained when a color-coded implant surface is used.

4 GPU-Based IMPLEMENTATION

To achieve interactive update rates, we have realized our visualization methods on the GPU. Our implementation is based on the stencilrouted k-buffer [9], which enables to render semi-transparent surfaces in correct visibility order, similar to the implementation described in [4]. Considering the complex spatial situation with multiple nested meshes, semi-transparent rendering is essential to avoid occlusions. In contrast to z-buffer-based rendering of opaque geometry, where of the set of fragments incoming to the same pixel only the fragment with the smallest depth is captured, the k-buffer allows to capture up to 8 fragments incoming to a pixel. To capture up to 32 fragments per pixel, we employ 4 k-buffer slices, which are filled by rendering the transparent geometry 4 times. For each fragment, we store 2×32 bits to encode an object ID, the fragment's depth in camera space, as well as the fragment's RGBA color value.

Using the k-buffer, rendering of the scene is performed in three steps: In the first step, all opaque geometry is rendered into the standard frame buffer. In the second step, all semi-transparent geometry is rendered into the k-buffer. The semi-transparent geometry consists of the bone surface meshes, the implant surface mesh, as well as the glyphs for our glyph-based distance visualization method. In the third step, a full screen ray-casting pass is performed. First, for each pixel the fragments stored in the k-buffer are fetched. These fragments are then sorted according to ascending camera space depth and blended using front-to-back blending, up to the camera space depth of the opaque geometry, which is obtained by accessing the frame buffer and projecting the clip space depth back into camera space. The accumulated color value is finally blended with the color value of the opaque geometry stored in the frame buffer.

For our glyph-based visualization approach, the glyphs have to be updated whenever the implant position is changed. This update is performed on the CPU. For each glyph seed point, we determine the intersection point of the ray emanating from the seed point in the direction of the outward implant surface normal with the inner bone surface. If no intersection is found, the seed point is lying outside of the inner bone surface, and the seed point is skipped. The distance value is then obtained from the distance between the seed point and the intersection point. To achieve interactive update rates, a kd-tree acceleration structure is used to determine ray/mesh intersections. The glyphs' seed points, normals, and heights are then uploaded into a buffer on the GPU, and by using a single generic cylinder triangle mesh, the glyphs are rendered as semi-transparent geometry using instanced rendering.

For the slice-based distance visualization, the slices are rendered on-the-fly during the ray-casting process by analytically computing the intersections between the rays and the slicing planes. To generate the distance fields of the bone and implant used by this approach, we first create a voxelization for each of the two meshes, using the GPU-based algorithm described in [3]. We then compute a distance transform of each voxel volume, using the GPU-based algorithm described in [13]. By utilizing the GPU, these computations can be performed in less than 2 seconds for a voxel resolution of 0.25 mm, allowing the user to almost instantaneously switch between different implant shapes and sizes. The distance fields are stored in GPU memory as 3D textures and are sampled using trilinear interpolation.

During the ray traversal of each pixel's stack of fragments, we maintain three flags, specifying whether the current position of the ray is lying inside the outer bone mesh, inside the inner bone mesh, and inside the implant mesh, respectively. These flags are toggled whenever a fragment of the respective mesh (identified by the fragment's object



Fig. 8. Comparison of different glyph types and different strategies to adapt glyphs to changing distances: Left: Squeezed cylindrical glyphs, uniform seeding density. Middle: Squeezed arrow glyphs, uniform seeding density. Right: Scaled arrow glyphs, adaptive seeding density.

ID) is encountered. The flags are used to clip the rendering of the slices at the bone and implant surfaces, to distinctly colorize the parts of the implant lying outside of the inner bone mesh, and to achieve a high-quality rendering of the inner and outer bone mesh: Since the inner and the outer bone mesh are generated in separated processes, it is possible that the inner mesh slightly sticks out of the outer mesh. This is fixed during rendering by clipping the inner mesh at the outer mesh, and by creating additional inner bone mesh fragments on-the-fly during ray-casting to close the resulting holes.

The rendering of contours along the intersection lines of the slices with the bone and the implant is performed in screen space. Whenever a slice fragment is created, we test in a 5×5 neighborhood of adjacent pixels whether the (potential) slice fragment corresponding to the respective neighboring pixel is lying outside of the outer bone surface or within the implant. If such a pixel is found, the center fragment is a border fragment and is colored accordingly, taking into account the number of neighboring 'outside' fragments to obtain antialiased contours.

5 RESULTS AND EVALUATION

All images in this manuscript were rendered in less than 30 ms on a standard desktop PC, equipped with an Intel Core 2 Quad Q9450 2.66 GHz processor, 8 GB of RAM, and an NVIDIA GeForce GTX 480 graphics card with 1536 MB of local video memory. The view port size was 1920×1200 .

To validate the effectiveness of the developed distance visualization methods, we have performed a user study. In this study, the images were created using the following visualization parameters: General parameters: $\alpha_{\text{Outer Bone}} = 0.2$ for focus+context disabled, $\alpha_{\text{Outer Bone}} = 0.9$ for focus+context enabled, $\alpha_{\text{Inner Bone}} = 0.2$, $\alpha_{\text{Implant}} = 1.0$ or 0.35, focus radius r = 30 mm. Parameters for glyphbased visualization: $r \cdot d = 5 \text{ mm}^2$, $r_{\text{max}} = 2.5$ mm, $d_{\text{max}} = 10$ mm (i.e., red corresponds to a distance of 0 mm, green to 10 mm). Parameters for slice-based visualization: Resolution of distance fields 0.25 mm, slice spacing 15 mm, slice thickness 1 mm, $d_{\text{max}} = 16$ mm (i.e., red corresponds to a distance of 0 mm, violet to 16 mm).

5.1 User Study

For the user study we recruited 30 participants, comprised of 21 computer science and 6 medical students as well as 3 experienced orthopedic surgeons. The participants were selected to be right-handed, and to have no color vision deficiency. All participants were daily users of computers. The students were exposed to the application for the first time. The three surgeons knew the underlying application including the stress visualization, but also used the distance visualization for the first time. The study was performed using the desktop PC described above with a standard 24 inch monitor. The position of the implant was alternatively controlled using the mouse or a Sensable Phantom Omni haptic device [15] (force feedback was not implemented). Simultaneous rotations and translations of the bone and implant model were controlled by mouse or keyboard input, respectively. For each user, an individual session of about 50-60 minutes was performed. Each session consisted of a demonstration and training phase, and three experiments.

Demonstration and Training Phase. The goal of the demonstration and training phase was to acquaint the participants with the visualization methods and the interaction mechanisms for controlling the implant position using the mouse and the Phantom device. To accelerate the learning process, in this phase we used a simplified geometrical setup consisting of two cylinders of different size, with the larger one (resembling the bone) being fixed. Initially, the smaller cylinder was located outside of the larger cylinder, and its main axis was rotated against the main axis of this cylinder. The task was to move the smaller cylinder (resembling the implant) such that it is concentrically centered inside the larger cylinder.

Firstly, the task was demonstrated to each participant individually using all of the three distance visualization methods, i.e., color coding on the implant surface, the glyph-based approach, and the slice-based approach, as well as both the mouse and the Phantom device. During the demonstration, the meaning of the graphical representations employed in our distance visualization methods, as well as the specific mapping of mouse and Phantom inputs onto movements of the implant, was verbally explained to the participant. After the demonstration, which took about 5 minutes, the participants were asked to perform the task. For each combination of visualization method and input device, about 2 minutes of training time were granted. The entire demonstration and training phase took about 20 minutes.

Experiment 1. In the first experiment, we quantitatively and qualitatively evaluated the three distance visualization methods within the interactive 3D environment. In contrast to the training phase, this experiment was performed using the real bone and implant geometry. The participants were asked to place the implant in a certain position within the bone. This target position was explained to the participants by means of a sketch showing a 2D cross section of the bone and the implant in the target position. In particular, the target position was precisely specified by the following criteria: a) the implant must be located inside the cortical bone shell without penetrating this shell, b) the distances between implant and bone in the calcar region and at the lateral wall of the femur must be minimized, and c) the rotation center of the joint must be preserved (i.e., the ball of the implant must be centered within the femoral head).

The participants were asked to perform this task a total of 12 times,



Fig. 9. Time (in seconds) needed by the individual users for interactive implant positioning (Experiment 1) dependent on the input device and the distance visualization method, as well as on the implant shape.

| | Mouse | | | Phantom | | |
|---------|-------|--------|--------|---------|--------|--------|
| Implant | Color | Glyphs | Slices | Color | Glyphs | Slices |
| CUT | 109 | 52 | 61 | 96 | 48 | 56 |
| G2 | 99 | 57 | 53 | 88 | 53 | 48 |

Table 1. Average time (in seconds) needed by the users for interactive implant positioning (Experiment 1) dependent on the input device and the distance visualization method (columns), as well as on the implant shape (rows).

| Implant | Color | Glyphs | Slices |
|---------|--------|---------|---------|
| CUT | 0/0/30 | 19/11/0 | 11/19/0 |
| G2 | 0/0/30 | 5/25/0 | 25/5/0 |

Table 2. Rankings of the distance visualization methods for interactive implant positioning (Experiment 1). The table shows the numbers of users that ranked the respective method first/second/third.

using the three distance visualization methods, two different implant shapes (CUT and G2), and the two input devices. The implant was always placed at the same starting position outside of the bone, with a different orientation than in the target position, and always the same initial view point was selected. To avoid biasing due to learning effects, with each participant we permuted the order in which the individual visualization methods, implants, and input devices were tested. During the experiment, we measured the time which was needed by the participants to navigate the implant into the target position, allowing a tolerance of 3 mm wrt translation and 5° wrt rotation. If the participant was not able to reach the target position within a 3 minute time interval, we proceeded with the next configuration. The times measured for each individual participant are shown in Figure 9, the average times are given in Table 1.

In addition, for each of the two implant shapes we asked the participants for a subjective ranking (first, second, third) of the three visualization methods according to their capability to communicate spatial relationships and distances. The rankings are shown in Table 2.

The results clearly demonstrate the effectiveness of our novel glyph- and slice-based distance visualization methods. Compared to pure color coding of distances, these methods allow for a significantly faster navigation of the implant into the target position. Notably, by using color only the task could not be accomplished by 6 of the users within the 3 minute time limit.

For the positioning of the CUT implant (see Figure 1), the majority of the users preferred the glyph-based approach over the slice-based approach. This is mainly due to the continuous movement of the glyphs, generating an additional visual attraction that can be processed instantly by the visual system. Especially the use of 'marshmallow'like cylindrical glyphs, which simulate quite plausibly the physical deformations of elastic glyphs under pressure, was received very positively by the users. The good matching of this approach with the users' expectation and experience in reality was given the main reason for this positive response. Considering the slice-based approach, the users criticized the fixation of the slicing planes to a few positions in the 3D domain. Due to this, a non-uniform distance distribution across the implant surface cannot easily be perceived during interactive placement of the implant. In addition, as the distance values within a slice are

| Implant | Color | Glyphs | Slices |
|---------|--------|---------|---------|
| CUT | 0/0/90 | 27/63/0 | 63/27/0 |
| G2 | 0/0/90 | 6/84/0 | 84/6/0 |

Table 3. Rankings of the distance visualization methods in still images (Experiment 2). The table shows the numbers of times each method was ranked first/second/third.

referring to 3D points that are not in the slicing plane, with increasing distances no immediate spatial correspondence is given between what is seen in the slice and the objects in the scene.

In contrast, for the positioning of the significantly larger G2 implant (see Figures 11 and 12), the majority of the users indicated a strong preference of the slice-based approach over the glyph-based approach. Due to its large size, this implant has to be navigated in a very narrow surrounding, giving a rather homogeneous distance distribution on the implant. Here, it turns out that in situations were the distances between the implant and the bone are uniformly low, glyph-based approaches fail in clearly depicting these distances. In such cases the very fine differences in the distance values cannot be perceived effectively anymore, and the large number of glyphs everywhere on the implant induces visual clutter.

Experiment 2. In the second experiment, we qualitatively evaluated the three distance visualization methods within 2D still images. We prepared 6 sets of images (see the additional material accompanying this paper), showing the two implants in 3 different (optimal and non-optimal) positions and from different view points. Each set consists of three images, each image showing the same implant in the same position and from the same view point, but each using a different distance visualization method. The three images of a set are simultaneously shown to the user on the monitor. With each user, we permuted the order in which the sets are presented, as well as the order of the images of each set on the screen. For each set, we asked the participant for a subjective ranking of the three visualization methods according to their capability to communicate spatial relationships and distances, analogous to Experiment 1. The rankings are given in Table 3.

Interestingly, a different feedback was given than in Experiment 1. Still, both glyph-based and slice-based approaches are rated superior over the pure color coding approach. However, the results show that for 2D still images the slice-based approach is preferred by the users for both the CUT and the G2 implant. Since the slices, and the coloring on them, can very effectively depict the implant and the bone structure, in case of a 2D still image it seems very easy for our visual system to infer on the spatial relationships between the implant and the bone also in the regions between the slices. Furthermore, since slicing planes restrict the visualization to only a few regions in the domain, visual clutter is significantly reduced, giving a more undisturbed view on the relevant structures.

Experiment 3. In the third experiment, we quantitatively and qualitatively evaluated the effect of differently shaped glyphs on the perception of spatial relationships and distances. Besides of the 'marshmallow'-like cylindrical glyphs, we provided arrow glyphs where—analogous to the cylinders—the length of the arrow represents the distance, and the axis represents the directions along which the distance is measured. The glyphs were scaled either in the same way as for the cylindrical glyphs (a smaller distance results in a larger



Fig. 10. Left: Rendering artifacts resulting from the glyphs partially sticking out of the bone surface. Middle: Clipping the glyphs at the bone surface does not resolve these artifacts. Right: By reordering fragments along the view rays the artifacts are removed.

| | Cylinders | Arrows | Arrows |
|-------------|-----------|----------|---------|
| | Squeezed | Squeezed | Scaled |
| Interactive | 22/8/0 | 0/0/30 | 8/22/0 |
| Still Image | 28/62/0 | 0/0/90 | 62/28/0 |

Table 4. Rankings of different glyph shapes as well as different strategies to adapt glyphs to changing distances for interactive implant positioning and in still images (Experiment 3). The table shows the numbers of times each method was ranked first/second/third.

| | Cylinders | Arrows | Arrows |
|----------|-----------|----------|--------|
| | Squeezed | Squeezed | Scaled |
| Time | 6.4 | 7.2 | 2.3 |
| Accuracy | 15% | 27% | 19% |

Table 5. Average time (in seconds) needed by the users to estimated the distance represented by a glyph, dependent on the glyph shape and the strategy to adapt glyphs to changing distances (Experiment 3). The second row shows the achieved estimation accuracy.

arrow width) or were scaled isotropically (a smaller distance results in a smaller arrow width). In the latter case, the seeding density was increased with decreasing distance to avoid that small distances become invisible. In Figure 8, the different options are shown next to each other. The idea behind using arrows instead of cylinders was to improve the perception of the spatial orientation of the glyphs, since an arrow head (cone) projects onto two clearly distinguishable shapes when seen from the side (triangle) or from the top (circle).

Using an adaptive seeding density of the glyphs is implemented as follows: We first generate a set of uniformly distributed seed points with a minimum spacing of $\delta = 0.5$ mm, using the strategy described in Section 3.1. In contrast to the case of using a uniform seeding density, we now create glyphs only at a subset of the seed points. In particular, we iterate over the set of seed points in a fix order, and create a glyph at the currently considered seed point if that glyph does not overlap with any previously created glyph. More precisely, a glyph with radius r_0 and seed point x_0 the condition $||x-x_0||_2 \ge r+r_0$ is satisfied. The radius r of the glyph (determined by the arrow head) is computed as r = 0.2d, where d denotes the height of the glyph (length of the arrow), i.e., the represented distance.

Firstly, we asked the users to move the CUT implant in the interactive 3D environment and to monitor the change of distances between bone and implant, using the three different glyph options. We then asked the users for a ranking of these options according to their capability to communicate spatial relationships and distances and for comments. The rankings are given in Table 4.

For interactive positioning there was a preference of all users for squeezed cylindrical glyphs over squeezed arrow glyphs. As the key reasons for this judgment it was reported that the squeezing of arrows was observed a rather unrealistic and thus non-intuitive effect, and especially when arrows become extremely flat or thin, their usual and expected potential to indicate spatial orientation was lost, making the rendering of the arrow head unnecessary.

Comparing squeezed cylindrical glyphs and isotropically scaled arrow glyphs, cylindrical glyphs were favored by over 70% of the users. Considering the arrow glyphs, many users were distracted by glyphs popping up or disappearing (this is a consequence of the adaptation of the seeding density, when larger glyphs are replaced by smaller glyphs, and vice versa), and particularly appreciated the visual coherence of the 'marshmallow'-like cylindrical glyphs. In addition, a small number of users criticized the visual clutter introduced by the increasing number of arrow glyphs when distances become small.

We then asked the users to rank the different glyph options in still images. This was performed similarly as in Experiment 2 by using 3 sets of images showing the CUT implant. The rankings are shown in Table 4. The results show a clear preference of the users for the isotropically scaled arrow glyphs over the 'marshmallow'-like cylindrical glyphs. In contrast to the dynamic view, where the arrow glyphs suffer from a limited visual coherence, in the static view the strength of arrows to better depict the glyphs' orientation was observed a clear advantage over cylinders.

Finally, we analyzed the different glyph options with respect to their capability to communicate absolute distances. We prepared three images, showing the CUT implant in different positions and from different view points. For each image, a different glyph option was used to depict the distances. In each image, 3 glyphs were labeled with the respective distances represented by these glyphs. The color scale was hidden from the participants. For each image, we then asked the participants to estimate the distances represented by another selected 5 glyphs, and we measured the time which was needed by the participants. The average times (per glyph) and estimation accuracies can be found in Table 5. Using isotropically scaled arrow glyphs, over 85% of the users could give the same quantitative estimate of absolute distance in one third of the time that was needed when cylindrical glyphs were used.

5.2 Technical Improvements

After the study was finished, the users were asked for modifications or extensions they could imagine for improving the proposed distance visualization. A small number of users observed the change in saturation on the cap of the cylindrical glyphs when partially penetrating the white, semi-transparent bone surface mesh. Even though none of



Fig. 11. G2 implant, front view: Distance visualization using color coding on the implant surface (left), oriented distance glyphs (middle left), and color coding on the implant surface and on a set of axial slices (middle right). Additionally, we provide the possibility to render the implant semi-transparently (right).



Fig. 12. G2 implant from Figure 11, back view.

the users considered this effect to have any negative influence on the distance perception, we integrated a rendering option into our tool to reduce this effect.

This rendering option is based on reordering the fragments of the semi-transparent geometry along each view ray, which is implemented using the k-buffer as described in Section 4. In our initial implementation, the fragments along each ray are sorted according to ascending camera space depth, and then blended using front-to-back blending. The reordering is now performed on the sorted stack of fragments, before front-to-back blending is applied.

Conceptually, if a ray enters a glyph before entering the bone, or leaves a glyph after leaving the bone (in both cases the glyph penetrates the bone surface), we move the corresponding bone surface fragments in front or behind of the glyph surface fragment, respectively. Entering and leaving of closed surfaces is determined by maintaining flags which are toggled whenever a fragment of the respective surface is encountered. Technically, the reordering of fragments is implemented as follows: If the ray encounters an 'entering' glyph fragment (i.e., a glyph fragment at the transition from the outside to the inside of a glyph), starting from the current position, we search for 'entering' bone fragments (from both the inner and outer bone surface mesh) along the ray within a certain distance from this position. These fragments are moved in front of the glyph fragment, without changing the relative order of the 'entering' bone fragments. Analogous, if the ray encounters a 'leaving' glyph fragment, starting from the current position, we search for 'leaving' bone fragments backwards along the ray within a certain distance from this position. These fragments are moved behind of the glyph fragment. In our implementation the search distance is set to 5 mm.

The front-to-back blending is then performed on the reordered stack of fragments. Even though it is clear that this approach cannot always guarantee that all transitions are fixed and depends on the specified search distance, as demonstrated in Figure 10 it can very effectively eliminate virtually all of the transitions.

6 CONCLUSION AND FUTURE WORK

We have presented two different approaches for visualizing distances between two objects in 3D space. These approaches are specifically tailored to support the user in the positioning of objects in interactive environments. We have introduced novel GPU-based visualization techniques using dynamic glyphs to reveal distance, depth, and directional information, and slicing planes on which distance values are instantly updated during navigation. These approaches have been developed to support 3D implant positioning in a medical planning system, and their strengths and potential limitations in this particular application have been analyzed in a number of user experiments.

In the future, we will analyze the capabilities of our approaches in other interactive applications like computational steering or virtual part assembly. The investigation of possibilities to automatically integrate distance visualization techniques into medical and technical illustrations will be another focus. From a perceptual point of view, we are very interested in the analysis of the perceptual effects that are caused by the combination of stereo rendering and the proposed distance visualization techniques. A more elaborate study of alternative or supplementary techniques for revealing distance, such as density volumes or textual annotations, in interactive environments is mandatory.

REFERENCES

- D. Borland and R. M. Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007.
- [2] Ç. Demiralp, G. E. Marai, S. Andrews, D. H. Laidlaw, J. J. Crisco, and C. Grimm. Modeling and visualization of inter-bone distances in joints. In *Proc. IEEE Visualization, Work in Progress*, pages 24–25, 2001.
- [3] C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Computational steering for patient-specific implant planning in orthopedics. In *Proc. Eurographics Workshop on Visual Computing for Biomedicine*, pages 83– 92, 2008.

- [4] C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE TVCG*, 15(6):1399–1406, 2009.
- [5] A. Guéziec. "Meshsweeper": Dynamic point-to-polygonal-mesh distance and applications. *IEEE TVCG*, 7(1):47–61, 2001.
- [6] C. G. Healey. Fundamental issues of visual perception for effective image generation. ACM SIGGRAPH Course, 1999.
- [7] M. W. Jones, J. A. Bærentzen, and M. Sramek. 3D distance fields: A survey of techniques and applications. *IEEE TVCG*, 12(4):581–599, 2006.
- [8] G. E. Marai, D. Laidlaw, Ç. Demiralp, S. Andrews, C. M. Grimm, and J. J. Crisco. Estimating joint contact areas and ligament lengths from bone kinematics and surfaces. *IEEE Transactions on Biomedical Engineering*, 51(5):790–799, 2004.
- [9] K. Myers and L. Bavoil. Stencil routed A-buffer. In Proc. ACM SIG-GRAPH Technical Sketch Program, page 21, 2007.
- [10] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997.
- [11] B. Preim, C. Tietjen, W. Spindler, and H.-O. Peitgen. Integration of measurement tools in medical 3d visualizations. In *Proc. IEEE Visualization*, pages 21–28, 2002.
- [12] I. Rössling, C. Cyrus, L. Dornheim, A. Boehm, and B. Preim. Fast and flexible distance measures for treatment planning. *International Journal* of Computer Assisted Radiology and Surgery, 5(6):633–646, 2010.
- [13] J. Schneider, M. Kraus, and R. Westermann. GPU-based real-time discrete Euclidean distance transforms with precise error bounds. In *Proc. VISAPP*, pages 435–442, 2009.
- [14] S. H. Schwartz. Visual Perception: A Clinical Orientation. McGraw-Hill, 2004.
- [15] Sensable. http://www.sensable.com/.
- [16] S. B. Steinman, B. A. Steinman, and R. P. Garzia. Foundations of Binocular Vision: A Clinical Perspective. McGraw-Hill, 2000.
- [17] K. Suder and F. Wörgötter. The control of low-level information flow in the visual system. *Reviews in the Neurosciences*, 11(2-3):127–146, 2000.
- [18] J. Süßmuth, W.-D. Protogerakis, A. Piazza, F. Enders, R. Naraghi, G. Greiner, and P. Hastreiter. Color-encoded distance visualization of cranial nerve-vessel contacts. *International Journal of Computer Assisted Radiology and Surgery*, 5(6):647–654, 2010.
- [19] R. B. Welch and D. H. Warren. Immediate perceptual response to intersensory discrepancy. *Psychological Bulletin*, 88(3):638–667, 1980.