

Adaptively Merging Large-Scale Range Data with Reflectance Properties

Ryusuke Sagawa, *Member, IEEE*, Ko Nishino, *Member, IEEE*, and Katsushi Ikeuchi, *Fellow, IEEE*

Abstract—In this paper, we tackle the problem of geometric and photometric modeling of large intricately shaped objects. Typical target objects we consider are cultural heritage objects. When constructing models of such objects, we are faced with several important issues that have not been addressed in the past—issues that mainly arise due to the large amount of data that has to be handled. We propose two novel approaches to efficiently handle such large amounts of data: A highly adaptive algorithm for merging range images and an adaptive nearest-neighbor search to be used with the algorithm. We construct an integrated mesh model of the target object in adaptive resolution, taking into account the geometric and/or photometric attributes associated with the range images. We use surface curvature for the geometric attributes and (laser) reflectance values for the photometric attributes. This adaptive merging framework leads to a significant reduction in the necessary amount of computational resources. Furthermore, the resulting adaptive mesh models can be of great use for applications such as texture mapping, as we will briefly demonstrate. Additionally, we propose an additional test for the k-d tree nearest-neighbor search algorithm. Our approach successfully omits back-tracking, which is controlled adaptively depending on the distance to the nearest neighbor. Since the main consumption of computational cost lies in the nearest-neighbor search, the proposed algorithm leads to a significant speed-up of the whole merging process. In this paper, we present the theories and algorithms of our approaches with pseudo code and apply them to several real objects, including large-scale cultural assets.

Index Terms—Adaptive integration of range images, laser reflectance strength, nearest-neighbor search.

1 INTRODUCTION

MODELING the shape and appearance of objects in the real world are important issues in computer vision. Cultural heritage objects are one of the worthiest candidates for modeling of their shape and appearance. There are several advantages to modeling these objects, for example, presentation, preservation, and restoration. Many cultural assets are large in scale and, at the same time, their shapes consist of delicate and intricately curved surfaces. In this paper, our target objects were mainly intricately shaped objects, such as statues of the Great Buddha and ancient temple buildings. High resolution and high precision are required for modeling these objects, just as when modeling small objects.

To acquire the 3D coordinates of the surface points of objects, we use range sensing systems. As most range sensing systems, e.g., stereo, structured light, and laser range finders, return range images obtained from particular viewing points, each output range image covers only a small portion of the target object surface. To ensure that the entire surface of the target is captured, multiple range images of the same object have to be acquired while changing the viewpoint. Thus, the main issue of modeling real objects is creating the entire model of an object from multiple range images. The

“integration” of multiple observations into a unified model is the main issue tackled in this paper.

1.1 Previous Work

1.1.1 Geometric Modeling

So far, due to the recent development of range finders, several researchers [1], [2], [3], [4], [5] have studied the modeling of cultural heritage objects using such powerful sensors. Fig. 1 shows the modeling steps using a range finder. The 3D modeling of the shape of the object is accomplished by performing the following three steps:

1. Acquiring the range images (scanning).
2. Aligning of those acquired range images from different viewpoints (aligning).
3. Reconstructing the unified 3D mesh model (merging).

In the first step, a target object is observed from various viewpoints. If it is a small object, it is mounted on a turntable or a robot arm.

In the second step, multiple range images are aligned into a common coordinate system. If an object is mounted on a turntable or a robot arm, the aligning step is accomplished by recording each local coordinate system a priori. Otherwise, range images are aligned by using registration algorithms which establish point correspondences and minimize the total distance between those points, e.g., feature-based methods [6], [7], ICP-based methods [8], [9], etc. Besl and McKay [8] proposed a point-based matching method, while Chen and Medioni’s method [10] is based on the distance evaluation between the point and the polygons. Wheeler and Ikeuchi [11] introduced M-estimator to the ICP scheme for discarding outliers as wrong correspondences. Neugebauer [12] proposed the idea of “simultaneous registration” that aligns range images simultaneously to avoid the error accumulation of the pairwise alignment methods. Several

• R. Sagawa is with the Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki-shi, Osaka, 567-0047, Japan.
E-mail: sagawa@am.sanken.osaka-u.ac.jp.

• K. Nishino is with the Department of Computer Science, Columbia University, 1214 Amsterdam Ave., MC 0401, New York, NY 10027.
E-mail: kon@cs.columbia.edu.

• K. Ikeuchi is with the Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan.
E-mail: ki@cvt.iis.u-tokyo.ac.jp.

Manuscript received 20 Oct. 2003; revised 12 July 2004; accepted 23 Aug. 2004; published online 14 Jan. 2005.

Recommended for acceptance by S. Seitz.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0325-1003.

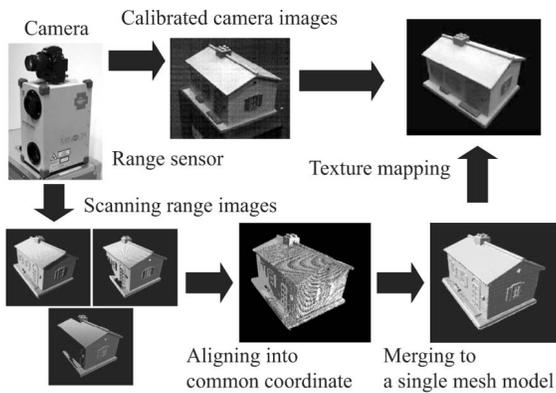


Fig. 1. Steps of geometric and photometric modeling of a small object.

other variants of simultaneous alignment have been developed [13], [14], [15], [16]. Huber and Hebert [17] proposed a method of automatic aligning range images without any knowledge about initial positions of range images, while other methods require a rough estimation of their positions.

For merging multiple prealigned range images, the third step of the pipeline in Fig. 1, several approaches have been proposed. Turk and Levoy [18] proposed a method to “zipper” two range images at a time, by first removing overlapping portions of the meshes, clipping one mesh against another, and then retriangulating the mesh on the boundary. Although integrating two range images is an intuitive process, pairwise merging does not remove errors well when merging multiple range images and is very sensitive to noise in the range images. Soucy and Laurendeau [19] also proposed a merging algorithm based on mesh representation, which is also sensitive to noise of mesh boundary. Given a number of range images overlapping each other, a merging procedure which extracts the isosurface is suitable, e.g., a merging method that makes use of volumetric, implicit-surface representation and then extracts the mesh surface by using the marching-cubes algorithm [20] (We will abbreviate that algorithm as MC throughout the rest of this paper). Hoppe et al. [21] constructed 3D surface models by applying MC to a discrete, implicit-surface function generated from a set of range images. After inferring local surface approximations from clouds of points based on tangent plane estimations, a local search was accomplished to compute the signed distance from each voxel to the surface of the point set. Curless and Levoy [22] enhanced Hoppe’s algorithm in a few significant points by developing a method to compute signed distances from multiple range images. Their method efficiently traverses the volume by resampling range images along scanlines of voxels; since it finds corresponding points on the screen space by projecting both voxels and a range image, and, in effect, updates only a narrow band of voxels on either side of the zero level, it does not go through all voxels. However, none of these methods, including [23], compensate for outliers of point data; it is assumed that the data is part of the object and the noise can be removed by averaging. Each of these methods suffers from inaccuracy, e.g., integrating unrelated observations, and these accuracy problems will affect the result even when the data is noise-free. Whitaker [24] proposed a level-set approach for integrating range images; this approach introduced a smoothness constraint using a Bayesian formulation for averaging observations. This method removes outliers of range images by smoothing. Level-set methods [24], [25], [26] use the narrow-band

method to reduce the computational cost, which updates the finite band of voxels on either side of zero level. Wheeler et al. [27], [28] addressed these important problems by designing a consensus surface algorithm. The consensus surface algorithm attempts to justify the selection of observations used to produce the average by finding a quorum or consensus of locally coherent observations. This process successfully eliminates many troublesome effects of noise and extraneous surface observations, and also provides desirable results with noise-free data. We developed a new method based on this method to merge large amounts of data. The methods proposed in [27], [29] use an octree as the data structure to reduce the computational cost of converting range images to a volumetric representation.

In our merging algorithm, we search the nearest-neighbor points of range images. The nearest-neighbor problem in multidimensional space itself is a major issue in many applications. Many methods have been developed to search for the nearest neighbor of a query. A simple exhaustive search computes the distance from a query to every point. Its computational cost is $O(n)$. This approach is clearly inefficient. Hashing and indexing [30], [31] finish a search in constant time; however, they require a large space in which to store the index table. For accessing multidimensional data, some hierarchical structures have been proposed, e.g., k-d tree [32], quadtree [33], R-tree [34], and octree spline [35]. These trees differ in structure, but their search algorithms are similar. The k-d tree [32] is one of the most widely used structures for searching for nearest neighbors. It is a variant of binary tree that partitions space using hyperplanes that are perpendicular to the coordinate axes. If a k-d tree consists of n records, the k-d tree requires $O(n \log_2 n)$ operations to construct and $O(\log_2 n)$ to search. Zhang [36] proposed a method which prunes branches of a k-d tree when their records are farther than a threshold. This method does not find any candidate if the nearest neighbor is farther than the threshold. Greenspan and Yurick [37] proposed an Ak-d tree for searching the nearest-neighbor points approximately to speed up aligning range images by omitting back-tracking. It does not guarantee to find the correct nearest neighbor. This method is similar to our idea [38]. However, the nearest neighbor may not be accurate if it is farther than the bin size. In this paper, we introduce a new thresholding method to the k-d tree search. This method efficiently reduces the search cost of merging large data sets of range images. The difference between ours and Zhang’s method is that our method always finds a candidate of the nearest neighbor, since we need a rough estimation even if it is far from a query for hole filling [39].

1.1.2 Photometric Modeling

Modeling appearance is known as photometric modeling and typically involves registration of color images with a geometric model so that the images can be texture mapped onto the geometric model (Fig. 1). Several methods of aligning color images with range images [27], [40], [41], [42] have been proposed. Neugebauer and Klein [42] proposed simultaneous registration of multiple texture images. Wheeler [27] uses occluding edges extracted from the 3D model for aligning with a 2D color image. If a range image is obtained by a laser range finder, laser reflectance

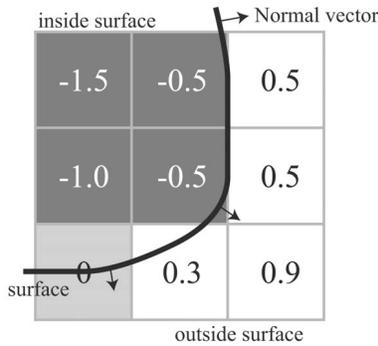


Fig. 2. An example of implicit surface function computed from an explicit surface.

strength (LRS) image can be obtained. Reflectance edges are more robust than occluding edges for the change of viewpoint. Kurazume et al. [40] thus extended a technique for aligning a 2D color image and a range image of an object by comparing the edges of the color image and the edges of LRS values attached to the range image. To align a color image with a merged mesh model by this method, we propose a new method to merge range images with LRS values.

1.2 Overview of This Paper

In order to model large-scale and intricately shaped objects, we propose the following techniques in this paper:

1. Adaptive merging of range images according to geometric characteristics.
2. Adaptive merging of photometric attributes of range images.
3. Adaptive searching of the nearest-neighbor points in a huge amount of range images.

We first describe our merging algorithm, which is based on Wheeler's method [28] in Section 2. Then, we propose two approaches to handle a huge amount of range images in merging range images. Section 3 describes a new method to merge range images in adaptive resolution. Then, we propose a new method to merge LRS values of range images in our merging framework in Section 4. Section 5 explains an adaptive neighbor search in finding the closest point of a range image.

2 CONSENSUS SURFACE ALGORITHM

2.1 Signed Distance Field

Using all range images, this method first constructs a volumetric representation, which is called a signed distance field (SDF). Those range images are assumed to be already aligned into a common coordinate frame. In this volumetric representation, 3D space is partitioned into three-dimensional voxels. A voxel has a signed distance $f(\mathbf{x})$ from its center \mathbf{x} to the nearest surface. The sign of $f(\mathbf{x})$ is positive if the center \mathbf{x} is outside the object; it is negative if the center \mathbf{x} is inside the object. Because the surface of the object is represented implicitly by $f(\mathbf{x}) = 0$, $f(\mathbf{x})$ is called the implicit surface function. Fig. 2 shows a 2D slice view of an example of SDF, which is composed of nine voxels. If the surface is converted to SDF, the $f(\mathbf{x})$ of each voxel is computed as shown. The voxels inside the object are dark gray and the ones outside the object are white.

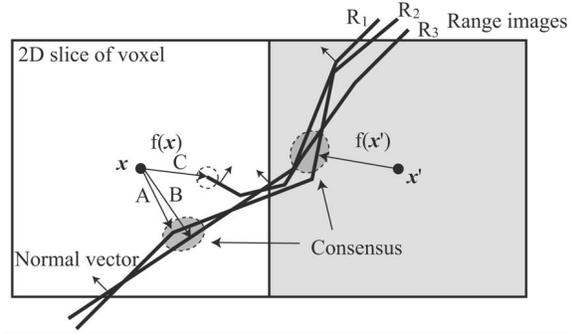


Fig. 3. Consensus surface algorithm: The signed distance is chosen from the consensus surfaces inside the gray circle.

2.2 Marching-Cubes Algorithm

Though a volumetric representation such as SDF can be visualized by volume rendering [43], a mesh model is suitable for our goal, which is geometric modeling and the analysis of objects. Lorensen and Cline [20] proposed the marching-cubes algorithm, which converts the volumetric representation to a mesh model. MC constructs a surface mesh by "marching" around the cubes which contain the zero level of the implicit surface $f(\mathbf{x}) = 0$. MC generates surface triangles to intersect voxels which have positive signed distances and voxels which have negative signed distances. Since the original algorithm has ambiguity in the algorithm of generating triangles, Nielson and Hamann [44] proposed a method to resolve ambiguous cases.

2.3 Taking a Consensus of Range Images

To compute signed distances from multiple range images, our approach is based on the consensus surface algorithm proposed by Wheeler et al. [28]. It computes the implicit surface function $f(\mathbf{x})$ of each voxel using multiple range images. Fig. 3 shows an example in which there are three range images which are intersecting between two neighboring voxels. The centers of the two voxels are \mathbf{x} and \mathbf{x}' . After this, the definition of a range image is a mesh model which consists of 3D vertices and triangles that connect the neighboring vertices. If there is a large discontinuity between vertices, we do not connect them in the same manner with [18], [23]. The normal vectors of the range images in Fig. 3 are facing outwards of the object. The normal vector of each vertex is computed by averaging the normal vectors of triangles which share the vertex.

To compute the signed distance value $f(\mathbf{x})$, the algorithm finds the nearest point to the center of the voxel in each range image. Since the nearest point is not always on a vertex of the triangles, the algorithm finds the nearest vertex of a range image and computes the nearest point in the triangles which include the nearest vertex. We assume that the true nearest point is in the neighborhood of the vertex. Although it is a heuristic that can fail, it works well in practice. In this example, there are three nearest points A , B , and C .

If the positions of the nearest points are close and their normal vectors are similar directions, we regard those points as having consensus. The consensus surface algorithm computes a reliable point by taking an average of the points which have consensus. If there are some reliable points, the algorithm chooses the reliable nearest points from them. In this case, because C is isolated, the method discards C and

takes an average of A and B , and computes the magnitude of $f(\mathbf{x})$ by the distance between \mathbf{x} and the averaged point. Since the inner product $(\mathbf{x} - \mathbf{p}) \cdot \mathbf{n} > 0$, where the averaged nearest point is \mathbf{p} and its normal vector is \mathbf{n} , \mathbf{x} is outside and the sign is determined as $f(\mathbf{x}) > 0$. Similarly, \mathbf{x}' is inside and $f(\mathbf{x}') < 0$. Because the algorithm discards outliers, it is not simply averaging the distance together as [22].

Though the original algorithm [28] uses a weighting scheme for computing consensus, we simply count the number of overlapping range images. If the number is more than a consensus threshold, the overlapping range images are valid and we take average of them. Otherwise, they are discarded from averaging. The consensus threshold depends on the accuracy of the range finder. In our experiment, the value is 2; thus, we can discard outliers if three range images are acquired at a point and one of them is an outlier as Fig. 3. On this assumption, we do not have to handle the boundary of range images in a special manner.

2.4 Subdividing Voxels Based on Octree

To determine where the implicit surface is, we have to compute the signed distances of all voxels around the zero level of the implicit function. It is costly to compute the signed distances of all voxels since the computational cost is $O(n^3)$ if the volume of interest is uniformly divided into $n \times n \times n$ voxels along each axis.

Wheeler [28] proposed the strategy of computing signed distances by subdividing the volume of interest recursively in an octree manner. It starts with the entire volume being a single voxel for computing the signed distance; it subdivides the voxel if the signed distance satisfies the following inequality,

$$|f(\mathbf{x})| < \frac{3\sqrt{3}}{2}w, \quad (1)$$

where w is the width of the voxel of interest. If (1) is satisfied, the implicit surface can exist inside the voxel or the neighbor voxels. It stops subdividing if the voxel becomes the user-defined finest resolution. Since the width of voxels that contain the implicit surface is the same, MC [20] is applied to the voxels of the same size which are subdivided in an octree manner. Subdividing voxels in an octree manner practically reduces the computational cost to $O(n^2)$ because the finest resolution voxels exist only near the surface.

3 ADAPTIVE MERGING ALGORITHM

Wheeler's algorithm produces a mesh model of the finest resolution everywhere; however, the dense sampling is not necessary where the shape of the object is nearly planar. Thus, we propose an algorithm to construct the 3D model in an efficient representation. By taking the surface curvature into account when splitting the voxels recursively in an octree manner, the resulting 3D surface will be subdivided more in high curvature areas and less in surface areas that are nearly planar. Therefore, the resulting geometric model will require fewer triangular patches to represent the object surface.

This is similar to research on mesh model simplification algorithms based on surfaces [45], [46], [47]. On the other hand, we reconstruct a simplified 3D model through a range image merging process based on implicit surface representation. Our approach is more reasonable than generating a dense mesh model of constant resolution and simplifying it.

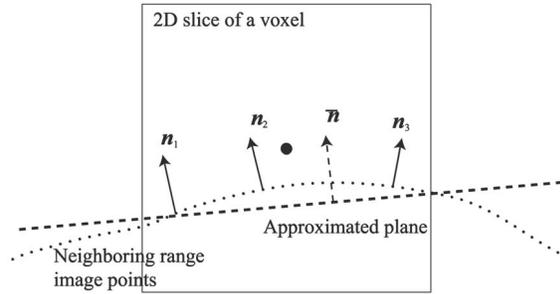


Fig. 4. Comparison of the normal vector of each vertex and the approximate normal $\bar{\mathbf{n}}$ by PCA.

The adaptive mesh model created by our method can be used for the input of simplification algorithms for further mesh optimization. The simplification is done when splitting voxels recursively, enabling better preservation of the topology and mass of the object compared with the results of other volume-based simplification methods [48], [49]. Frisken et al. [50] proposed adaptive sampling of the signed distance field. They generate surface meshes based on the surface nets approach [51]. For converting the volumetric representation of the 3D model to a triangle-based mesh model, we propose an extended version of the marching-cube algorithm; this version handles voxels at different resolutions. However, the aim of their paper is not merging range images. Thus, we propose a method for adaptively merging range images.

3.1 Subdividing Voxels Based on the Geometric Attributes of Range Images

We determine the sampling interval of the signed distance, depending on the variation of geometric attributes to efficiently represent the final mesh model. Depending on the change in surface curvature, the proposed method coarsely samples in planar areas, consequently reducing the amount of data and computation, while creating a finer model of an intricately shaped object by efficiently utilizing computation power.

Our method determines the variation of surface curvature by comparing surface normals of range images. We compare the normal \mathbf{n}_i of each 3D point of all range images inside the voxel in interest and the normal $\bar{\mathbf{n}}$ of the approximated plane (see Fig. 4), which can be estimated by applying principal component analysis (PCA) to all point data in the voxel. If the angle between the data point normals \mathbf{n}_i and approximate normal $\bar{\mathbf{n}}$ satisfies

$$\max_i(\arccos(\mathbf{n}_i \cdot \bar{\mathbf{n}})) < \delta_n, \quad (2)$$

where δ_n is the threshold of the angle, the sampling interval is fine enough, and no further voxel splitting is required.

To avoid erroneous subdivisions of voxels by the influence of noise included in each range image, our method takes a consensus between range images on the decision of voxel subdivision. Now, N_n is the number of range images which satisfies (2) and N_{all} is that of consensus range images. Our method does not subdivide the voxel if

$$\frac{N_n}{N_{all}} > T_n, \quad (3)$$

where T_n is the threshold of consensus for normal vectors.

If a range image is not smooth, the computation of normal vectors becomes unstable, especially in the case that it contains zigzag noise, which has high spatial frequency. It often occurs with a laser range finder when range images are acquired under inappropriate conditions. In such a case, the algorithm of computing consensus does not work well, and neither does the subdivision based on geometric attributes because their criteria are based on normal vectors. Thus, we proposed another method for taking consensus of range images [52], which does not depend on normal vectors. If range images contain zigzag noise, we refine those range images by [52] before merging them. Since the refined range images have reliable normal vectors, we can avoid erroneous subdivisions of voxels by the influence of noise included in each range image. We therefore assume that the normal vector is reliable for subdividing voxels.

The algorithm of traversing an octree with adaptive voxel subdivision is represented as Algorithm 1. In this algorithm, **ConsensusSurface** ($\mathbf{x}, R_{\text{set}}$) computes the nearest point \mathbf{p} and its normal vector \mathbf{n} from the point \mathbf{x} . The changes from the original algorithm are indicated by gray boxes. To determine whether to subdivide the current voxel N , we consider the curvature of range images inside the voxel by **LocalCurvature** (N, R_{set}). **LocalCurvature** returns the percentage of range images which satisfies (2). Moreover, since we subdivide the voxels adaptively, the voxels attain sufficient resolution even if the threshold value of the magnitude of a signed distance is reduced to $\frac{\sqrt{3}}{2}w$. If voxels are at a fixed resolution, a voxel should be subdivided if one of the neighboring voxels contains vertices of range images. However, in the case of adaptive resolution, it is enough to subdivide voxels which contains vertices in order to attain the sufficient resolution of a merged mesh model.

Algorithm 1 AdaptiveTraverseOctree($N, d_{\text{max}}, R_{\text{set}}$)

Input: Current Node of Octree: N
Input: Maximum Depth of Octree: d_{max}
Input: Set of Range Images: R_{set}
Local: Center of N : \mathbf{x}
Local: Octree Depth of N : d
Local: Width of N : w
Local: Tuple of Point, Normal: $\langle \mathbf{p}, \mathbf{n} \rangle$
Output: Signed Distance of N : v

```

 $\langle \mathbf{p}, \mathbf{n} \rangle \leftarrow \text{ConsensusSurface}(\mathbf{x}, R_{\text{set}})$ 
if  $(\mathbf{x} - \mathbf{p}) \cdot \mathbf{n} > 0$  then
   $v \leftarrow \|\mathbf{x} - \mathbf{p}\|$ 
else
   $v \leftarrow -\|\mathbf{x} - \mathbf{p}\|$ 
end if
if  $|v| < \frac{\sqrt{3}}{2}w \wedge d < d_{\text{max}}$ 
   $\wedge \text{LocalCurvature}(N, R_{\text{set}}) > T_n$  then
    for all children  $N_i (i = 0, \dots, 7)$  of  $N$  do
      AdaptiveTraverseOctree( $N_i, d_{\text{max}}, R_{\text{set}}$ )
    end for
  end if

```

3.2 Marching Cubes for Adaptive Octree

The original marching-cubes algorithm can be applied only to voxels that have the same resolution (size of voxels). We extend the algorithm to triangulate voxels at different resolutions as generated in our method.

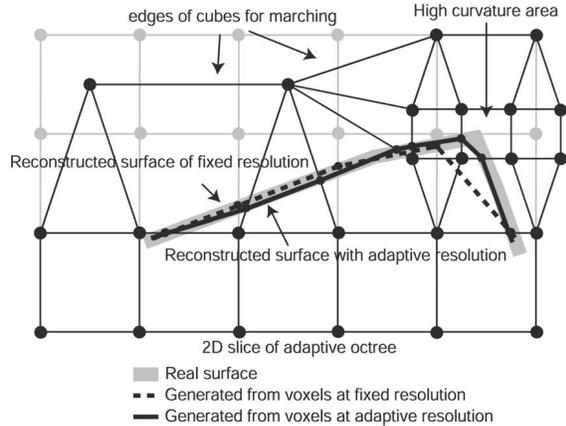


Fig. 5. Edges connecting adjacent voxels in an adaptive octree and the generated mesh model by MC.

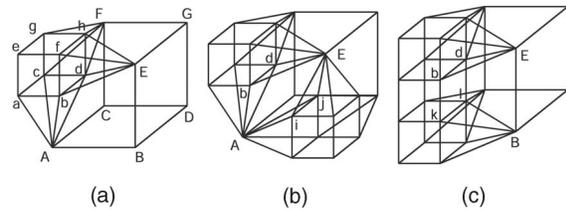


Fig. 6. Partially subdivided cubes.

For voxels that are surrounded by voxels with the same resolution, the vertices of a cube to march are the centers of eight adjacent voxels. In a similar manner, voxels surrounded by different size voxels will have a set of adjacent voxels, which are no longer cube-shaped as shown in Fig. 5. When we use voxels of fixed resolution (grids of gray lines), a mesh model of the dotted line is generated, and its vertices are on the edges of cubes. When we use adaptively subdivided voxels up to one level higher resolution (grids of black lines), the mesh model of a solid line is generated, and its vertices are on the edges of transformed cubes. If we subdivide the high curvature area into small voxels, the generated mesh model gets closer to the real surface (gray thick lines) without increasing unnecessary vertices in planar areas. Since a transformed cube becomes a skewed rectangle or a triangle in a 2D slice of the volume, as shown in Fig. 5, the vertices of the mesh model generated by MC are on those edges.

Fig. 6 shows three partially subdivided cubes, whose vertices are the centers of voxels. One of eight voxels which compose a cube is subdivided in Fig. 6a. Similarly, two voxels are subdivided in Figs. 6b and 6c. After subdivision, a cube is partitioned into several forms: For example, in Fig. 6a, the number of forms is seven and their vertices are $\{ABCDdEFG\}$, $\{Aabcd\}$, $\{ABbdE\}$, $\{Ebdfh\}$, $\{dEhFG\}$, $\{Fcdgh\}$, and $\{abc-defgh\}$. The form composed by $\{abc-defgh\}$ is a cube and $\{Aabcd\}$ is a quadratic pyramid, while $\{ABbdE\}$ is not a polyhedron. Since MC interpolates points of zero level on the edges, the form to which MC is applied is not necessarily a cube, nor even a polyhedron.

Fig. 7 shows examples of a transformed cube. Fig. 7b is a pyramid, such as $\{Aabcd\}$ in Fig. 6a. We can regard that Fig. 7b is equal to Fig. 7a whose upper four vertices have the same signed distance with the top vertex of Fig. 7b and they gather to the position of top vertex of Fig. 7b. Thus, we can generate the isosurface of Fig. 7b by applying MC to the transformed

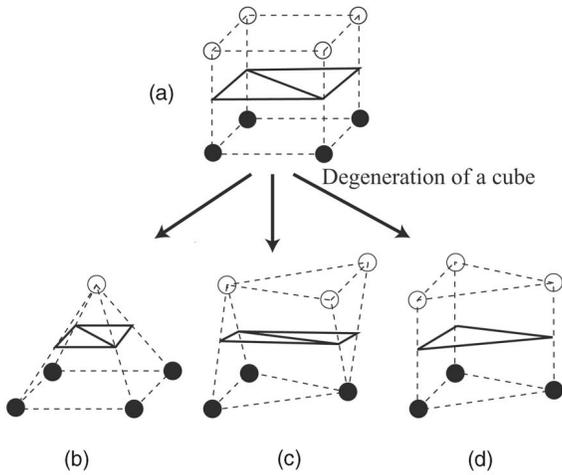


Fig. 7. Examples of degenerated cubes and the surfaces generated by MC.

cube from Fig. 7a. By regarding the irregular forms as degenerated and transformed cubes, MC can be applied to them without creating new tables of mesh generation for the irregular forms.

Fig. 7c is {AijbdE} in Figs. 6b and 7d is {BklbdE} in Fig. 6c. In the case of Figs. 7b and 7c, two triangles are generated. However, the number of triangles is reduced in the case of Fig. 7d because the number of edges of the transformed cube is reduced. We therefore removed the redundant vertices of the mesh model after generation by MC.

4 ADAPTIVE MERGING WITH REFLECTANCE PROPERTIES

With regard to applications that utilize geometric models, for instance, 3D object recognition and localization tasks, it is desirable to construct 3D models with additional attributes such as color and intensity. With the additional information provided by photometric attributes, higher accuracy and robustness can be expected from those applications. Thus, it is necessary to efficiently create a model with photometric attributes. In this section, we consider an adaptive merging method which subdivides voxels based on photometric attributes.

When we acquire a range image using a laser range finder, we can obtain a LRS value of the surface for each vertex of the range image. Thus, our proposed method takes a consensus of the reflectance parameters of the target object from multiple range images. It reconstructs the 3D model with reflectance parameters attached per vertex, discarding

outliers due to noise and specular reflection produced in the image-capturing process.

4.1 Laser Reflectance Strength Attached to Range Images

Laser range finders measure distance by shooting a laser and receiving its reflection from the target object. The distance to a particular point on the target object is computed by measuring the time duration between the time laser was shot and the time it was received back in time-of-flight range finders by measuring the phase difference in phase-transition based range finders, or by optical triangulation of the illuminant, surface, and optical sensors. In either case, an LRS value, which is the ratio of the discharged laser strength and the reflected laser strength, can be obtained per each 3D point. If we assume the dichromatic reflection model, as the laser can be considered to be light with a very narrow wavelength distribution, almost a single value, the behavior of the reflected laser on the target surface can be considered to be the same as the general light reflection. Namely, almost isotropic reflection analogous to diffuse reflection and sharp reflection distributed around the perfect mirror direction analogous to specular reflection occurs. Since the specular reflection is observed only if the laser is almost parallel to the normal direction of the object surface, the observed laser is usually caused by the diffusive reflection. Thus, it is exceptional to observe the specular reflection, which can be regarded as an outlier. Fig. 8 depicts four images using the LRS values attached to each 3D point as pixel values, rendered from the view point of the laser range finder Cyrax2400 [53].

LRS values are considered to depend on the characteristics of the surface, the incident angle of laser light, and the distance from the sensor. The LRS value which we obtain by a laser range finder is the ratio of the discharged laser strength and the reflected laser strength. If we assume that the LRS value depends only on the diffuse reflection, the relationship of the LRS value and the other parameters are represented by the following equation:

$$I_1 = I_0 e^{-\alpha x}, \quad (4)$$

$$I_2 = r I_1 e^{-\alpha x} \cos \theta, \quad (5)$$

where I_0 is the discharged laser strength, I_1 is the incident laser strength on the surface, and I_2 is the reflected laser strength. As for the other parameters, x is the distance from the laser range finder, α is the absorption coefficient of the laser in the air, r is the reflectance parameter of the surface, and θ is the incident angle of the laser (see Fig. 9). Since I_0 is a given value and I_2 is measured by the sensor, while I_1 is unknown, (4) and (5) become



Fig. 8. Range images of the Great Buddha of Kamakura using LRS values as pixel values.

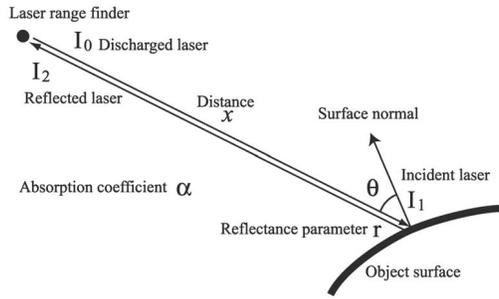


Fig. 9. Reflection model of a laser light.

$$\frac{I_2}{I_0} = r e^{-2\alpha x} \cos \theta. \quad (6)$$

Since the reflectance parameter r is a characteristic value to the surface, we want to obtain r by using several observations from various viewpoints.

Since we can obtain I_2/I_0 , x and θ for each vertex of range images, the unknown variables are r and α . The logarithm of (6) becomes

$$\log \frac{I_2}{I_0} = \log r - 2\alpha x + \log \cos \theta. \quad (7)$$

Thus, the system becomes a linear equation with two unknowns. Since we find corresponding points of the range images in taking a consensus of range images, as shown in Fig. 3, we can solve the system if more than two corresponding points are found. If we have more than three equations, we can solve the system by the least square method.

Another method to estimate the reflectance parameter r is calibrating the absorption coefficient α before scanning a target. Since the absorption coefficient α depends on the atmosphere around the environment of the target, α can be assumed to be constant for all points in the range images which are acquired at roughly the same time. If we measure the same point from a fixed direction with varying distances, we can estimate α by fitting α to the following equation:

$$y = -2\alpha x + c, \quad (8)$$

where $y = \log I_2/I_0$ and $c = \log r + \log \cos \theta$. Once α is determined, the reflectance parameter r can be computed by (6); however, the reflectance parameters of the corresponding points, which are found in the merging process, vary because of the specular reflection. Thus, in the merging process, we take a consensus of r of the corresponding points of the range images.

The reflectance variation of the corresponding points should have a DC component because of the invariant diffuse reflection with a sharp peak caused by specular reflection added to it, which can be observed from a narrow viewing direction. Thus, if the point is observed from a sufficient number of viewing directions, the histogram of the reflectance parameters should have a sharp peak at the diffuse reflection value, with some distribution around it due to specular reflection. Fig. 10 depicts an example of the LRS values of the corresponding points for a voxel. Based on this consideration, we take the median value of the corresponding points as a consensus value of the reflectance parameter.

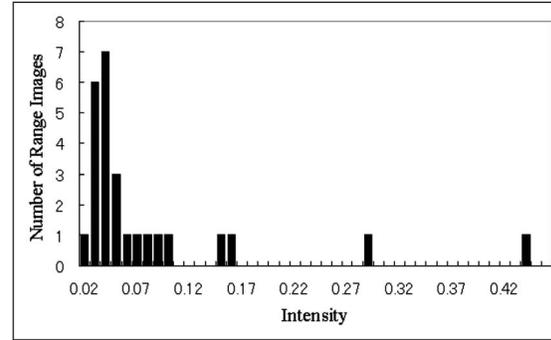


Fig. 10. An example of the histogram of the reflectance parameter of corresponding points. Some outliers due to specular reflection are observed. In this case, the median value is 0.04.

4.2 Subdividing Voxels Based on the Reflectance of Range Images

We have introduced a new criterion of voxel subdivision based on the geometric attributes of the surface for the adaptive merging method in Section 3.1. As the second criterion of voxel subdivision, we propose the voxel subdivision based on the variation of the reflectance parameters. Photometric attributes are used for the criterion of mesh simplification in [45], [47]. We estimate reflectance parameters in addition to geometric attributes. For further applications, such as texture mapping, we subdivide voxels, which are not subdivided by geometric attributes, from the viewpoint of reflectance parameters. It can be accomplished in a similar manner as with geometric attributes.

If we subdivide voxels around the drastic variation of reflectance parameters, each triangular patch contains almost the same reflectance parameters. Since the LRS image and color/intensity image of an object are highly correlated, those 3D models tessellated with regard to the reflectance variation of the models are useful to accomplish further texture analysis and synthesis. For instance, the registration of a 2D image and a 3D model of an object can be considered. Kurazume et al. [40] used the edges of LRS values attached to a range image. If we apply this method to our merged 3D model, the subdivision based on the reflectance parameters is desirable to extract fine edges of reflectance parameters, and we can directly extract 3D reflectance edges from range images. Moreover, when a texture image is mapped on the adaptive model subdivided based on the reflectance parameters, view-dependent texture mapping like [54] can achieve higher compression since global texture compression stacking triangular patches with a similar texture can be applied.

In a similar manner to subdividing by the curvature of the surface, our method computes the variation of reflectance parameters of 3D points inside the voxel of interest. Now, r_i, r_j are the reflectance parameters of neighbor points included in a range image. If the maximum difference satisfies

$$\max_{i,j} (\text{Distance}(r_i, r_j)) < \delta_r, \quad (9)$$

where δ_r is the threshold and $\text{Distance}(r_i, r_j)$ is the function which computes the difference of two reflectance parameters, the sampling interval is fine enough for the range image.

Our method also takes a consensus while considering the reflectance parameters. Similar to (3), our method does not subdivide the voxel if

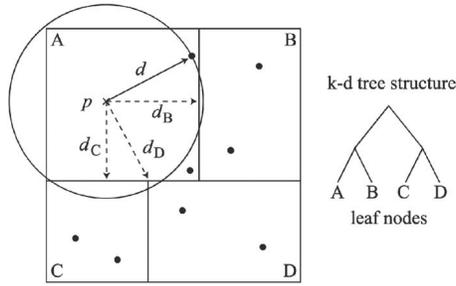


Fig. 11. A 2D example of a k-d tree.

$$\frac{N_r}{N_{all}} > T_r, \quad (10)$$

where N_r is the number of range images which satisfy (9) and T_r is the threshold of consensus for the reflectance parameters.

5 ADAPTIVE NEAREST-NEIGHBOR SEARCH

In the previous section, we described algorithms for constructing 3D models that efficiently represent the object by adaptively merging a large amount of range images. When the number of range images and the number of points in those range images is very large, it is also crucial to speed up the merging process. The speed of the whole process depends on how efficiently one can search the nearest-neighbor points. In many cases of merging a lot of range images simultaneously, most of the vertices of the range images can be discarded from searching the nearest-neighbor points since the portion of range images which are overlapped at a position is quite small compared with the total range images.

We introduce an additional test that takes place when traversing the k-d tree. This test compares the distance from a query to the nearest neighbor with a threshold defined by the user. Since this method improves the locality of reference, we can reduce not only the computational cost for searching the nearest neighbor but also the required memory to traverse a k-d tree. At the same time as reducing the search cost, this method roughly estimates the nearest neighbor even if it is far from a query. The signed distance is used when we fill holes of a model [39]. Since the hole filling works well even if we do not find the true nearest neighbor, our adaptive nearest-neighbor search is effective.

5.1 Basic Search Algorithm Using k-d Tree

First, we explain the basic algorithm by which the k-d tree searches for the nearest neighbor. Fig. 11 shows a 2D example of a k-d tree that consists of four leaf nodes labeled A, B, C, and D. We do not describe how to construct a k-d tree in this paper; for details, please refer to [32].

Now, we describe how to find the nearest-neighbor point from a query point \mathbf{p} . In the search algorithm, we start at the root node and traverse down to the leaf node that contains the query point. In Fig. 12, the leaf node A contains \mathbf{p} and we compute the distances from \mathbf{p} to the records of A.

To avoid examining all leaf nodes, the algorithm prunes branches by the Bounds-Overlap-Ball (BOB) test [32]. After node A is examined, the distance from \mathbf{p} to the nearest neighbor is d . We examine B if d satisfies the following BOB test:

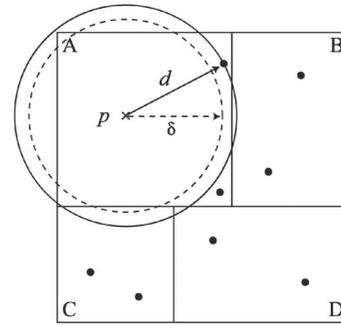


Fig. 12. The Bounds-Overlap-Threshold (BOT) test.

$$d > d_B, \quad (11)$$

where d_B is the distance from the query point \mathbf{p} to the boundary of A and B. Similarly, we compare d with d_C and d_D to decide whether or not we will examine C and D. In this case, d satisfies (11) for B, C, and D. Thus, we have to examine all nodes. If the hypersphere of radius d is completely inside of a node after examining the node, the algorithm finishes the search. (This is called the Ball-Within-Bounds (BWB) test.)

5.2 Bounds-Overlap-Threshold Test

In this section, we introduce the Bounds-Overlap-Threshold (BOT) test to the search algorithm. BOT test prunes branches which are farther than a threshold δ in the similar manner to BOB test. In Fig. 12, the node B and D are pruned. Though this method is same as the thresholding technique proposed by Zhang [36], his method discards all records farther than the threshold from the result. In this situation, since the records even in node A are discarded, it finds no records. On the other hand, our method chooses the nearest one from all the records which are examined while traversing a tree. Thus, it finds at least a record even if all records are far away from the query. In Fig. 12, the nearest neighbor is the record in node A, to which the distance from the query is d . The pseudo code is shown in the Appendix.

When we apply the BOT test to the consensus surface algorithm, if the distance from a voxel to the range images is larger than $\frac{\sqrt{3}}{2}w$, where w is the interval of voxels, there is no surface around the voxel. Thus, it is enough for us to find that no point in the k-d tree is closer than $\frac{\sqrt{3}}{2}w$, and we set $\delta = \frac{\sqrt{3}}{2}w$. Our merging method reduces the computation of the SDF in an octree manner; therefore, the voxel width w varies according to the depth of octree subdivision to which the current voxel belongs; we adaptively change the threshold δ as well as the voxel width w .

6 EXPERIMENTS

For evaluation of our method, we have built a PC cluster that consists of eight PCs, each equipped with dual PentiumIII 800MHz processors with 1GB memory, connected by 100BASE-TX Ethernet. Since consensus surfaces can be computed independently requiring only adjacent voxels, we have proposed a parallel merging algorithm [55], [56] by splitting the whole volume into pieces and parallel searching the nearest neighbors. With this parallel implementation, we are able to handle a huge amount of range

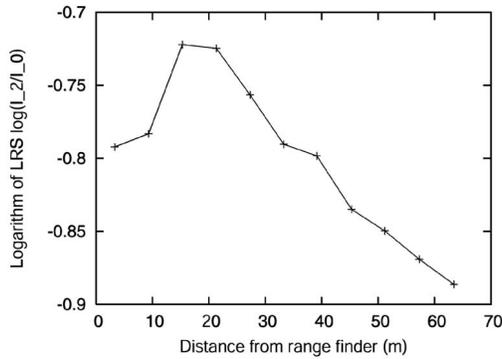


Fig. 13. Ratio of discharged and reflected laser.

image data. In our experiments, we use the Cyrax 2400 and 2500 [53] to measure distances.

6.1 Preliminary Experiment of Estimation of Reflectance Parameter

First, we verify the reflection model of (7). We measure the same point several times from different distances. Since the incident angles are constant in this experiment, the reflectance parameters are considered to satisfy (8). Fig. 13 shows the logarithm of the LRS value ($\log(I_2/I_0)$) at each distance. At far distance ($> 20\text{m}$), the logarithm of the LRS values becomes almost linear. Thus, it indicates that our model is appropriate and α is estimated to be 1.7×10^{-3} .

However, at near distance ($< 20\text{m}$), the logarithm of LRS values becomes nonlinear. One of the reasons is the focus of the laser beam. Since laser range finders use lenses to detect the light, the lasers are focused in the expected range. If the distance of the object is in the unfocused range, a part of the reflected laser does not land on the receiver. Thus, the reflected laser is clipped by the receiver and the power of light becomes less than expected. Though this result occurs

in the case of Cyrax, similar effects are expected to occur with other laser range finders.

Nevertheless, our model works well in the focused range of a laser range finder; however, we have to take the focus/clipping effects into account when the object is in the unfocused range. In the following experiments, we measured objects in the focused range, and the reflectance parameters were computed by (7) with the estimated α . If we use range images acquired in the unfocused range, a look-up table, which is created from the result shown in Fig. 13, is utilized to estimate the reflectance parameters.

6.2 Adaptive Merging of Range Images

We first apply our method to a standard model from Stanford University [57]. Fig. 14 shows the merged results of the bunny from 10 range images. Fig. 14a is the result without adaptive subdivision and Fig. 14b is the result with adaptive subdivision based on the geometric attributes. When we merge range images without adaptive integration, the volume is divided to $128 \times 128 \times 128$ voxels in the finest resolution. We used $\delta_n = 37^\circ$ and $T_n = 0.5$ for (2) and (3) for generating Fig. 14b, which is chosen manually. Fig. 14a contains 34,667 vertices and 69,463 triangles, while Fig. 14b contains 23,671 vertices and 47,338 triangles. The computational times are 10 minutes and 4.8 minutes, respectively. We computed the difference of Figs. 14a and 14b using Metro [58]. The mean/RMS/max differences are 0.096 percent/0.23 percent/2.7 percent of the longest edge of bounding box. Therefore, our method effectively reduced the amount of data and the computational time.

6.3 Adaptive Merging of Range-Reflectance Images

Next, we applied our algorithm to the Great Buddha of Kamakura, whose height is about 11.3m. We acquired 16 range images with LRS values attached to each 3D point; about 0.3 million vertices and 0.6 million triangles were

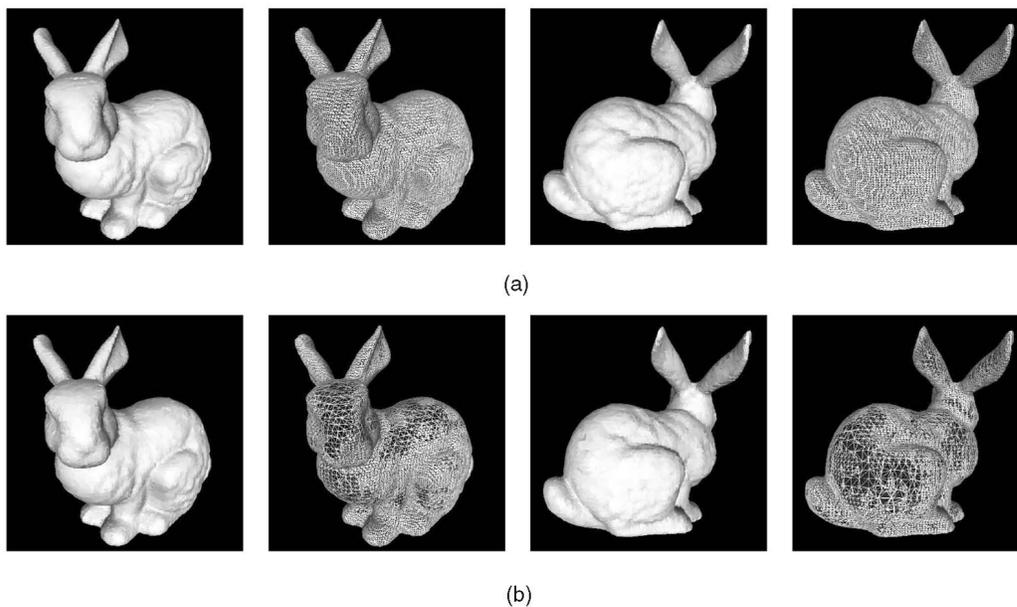


Fig. 14. Merged models of Stanford bunny. (a) Without adaptive subdivision. (b) With adaptive subdivision based on geometric attributes.

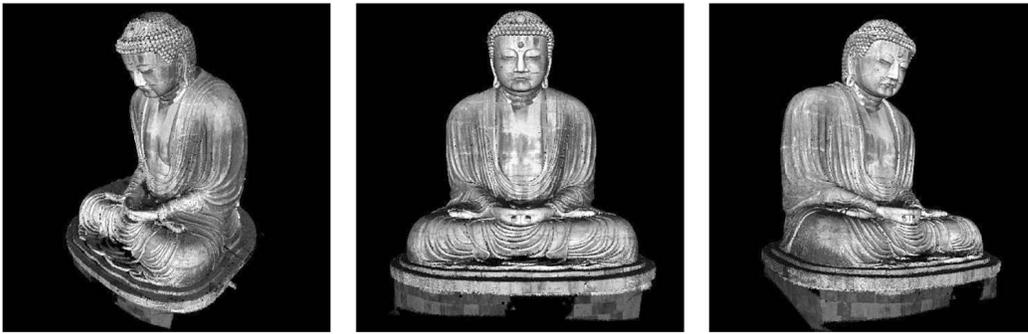


Fig. 15. The merging result of the Great Buddha of Kamakura with reflectance parameters.

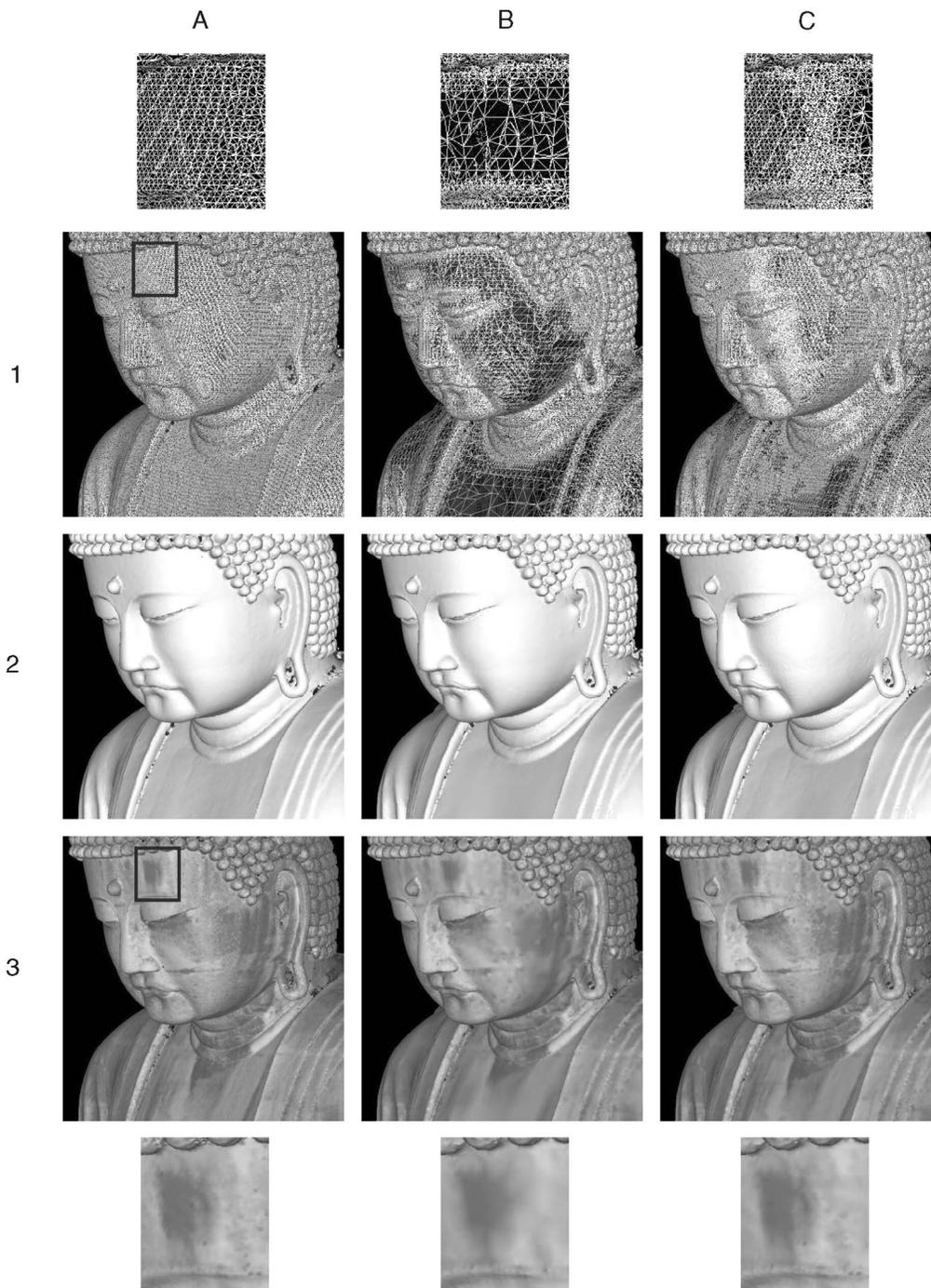


Fig. 16. Adaptive merging results of the Kamakura Buddha with reflectance parameter.

TABLE 1
Statistics of Models of the Buddha

	Number of points	Time for integration	Mean/RMS/Max difference
(A)	3.0 million	61 min.	N/A
(B)	1.2 million	25 min.	0.99/3.5/86 mm
(C)	1.4 million	30 min.	0.44/1.2/66 mm

contained in each range image. Fig. 8 shows four of the range images with reflectance parameters, and Fig. 15 shows the merging result with reflectance parameters. Fig. 16 shows three different results of our method. Column A contains the models created without adaptive integration, Column B contains those created by adaptive subdivision only based on the curvature of the surface, and Column C contains those with adaptive subdivision by the estimation of curvature and reflectance. Row 1 contains wire-frame representations and Row 2 has polygonal representations of these models. Row 3 shows the images rendered with reflectance. The far upper and far lower rows are zoom-ups of the forehead of the Buddha. We used $\delta_n = 18^\circ$, $\delta_r = 0.1$, and $T_n = T_r = 0.5$.

When we merge range images without adaptive integration, the volume is divided to $1,024 \times 1,024 \times 1,024 (= 2^{10})$ voxels in the finest resolution, and the width of the finest voxel is about 1.4cm. The merged model consists of 3.0 million vertices and 6.0 million triangles. The mean difference between the merged model and a range image is 2.7mm. It is appropriate compared with the maximum error of the Cyrax2400, which is about 7-8mm.

The figures in Row 2 are rendered using triangular faces. The result of the adaptive merging (B-2) seems completely the same as the result of the fixed resolution (A-2). However, if they are rendered by a wire frame, as shown in Row 1, we can see that our adaptive merging algorithm generates larger triangles in more or less planar areas. Thus, the size of the result of the adaptive merging is reduced to less than 50 percent of the result of the fixed resolution. Consequently, the time for merging is also reduced to less than 50 percent.

Fig. 16 Column A, Row 3 is the result of reflectance merging without adaptive integration. The texture of reflectance of Fig. 16 Column B, Row 3 is smoothed out compared with Fig. 16 Column A, Row 3. However, by considering the reflectance as a criterion of voxel subdivision, the sharp edges due to the variation in reflectance are well preserved (see Fig. 16 Column C, Row 3).

The statistics of the merging process are described in Table 1. The adaptive merging algorithm reduces the amount of data and computation time required using the original merging method. We compared the difference between the models of Columns A and B, the models of Columns A and C using Metro. The mean difference (0.99mm) between Columns A and B was quite small compared with the height of the Buddha. Our method successfully reduces the amount of data and computation time. However, the mean errors are quite small compared with the Buddha size. Also, adaptive

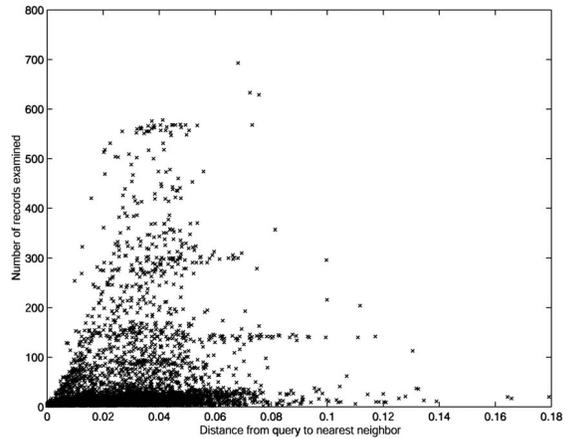


Fig. 17. Relationship between distance from a query to the nearest neighbor and the number of records examined using the basic search algorithm for merging range images.

merging based on the photometric attributes successfully reduces the amount of data and the computational time, while it preserves the edges of the reflectance well.

6.4 Evaluation of BOT Test

Figs. 17 and 18 show an example of the distribution of the number of records examined during the search for a nearest-neighbor point in merging range images. When we search for the nearest-neighbor points using the BOT test, the number of records examined gets closer to 1 at any distance from the query. This is because we adjust threshold δ according to the criterion described in Section 5.2. In this example, the total numbers of records examined are 11,844,253 without the BOT test and 2,716,475 with the BOT test. Specifically, the computational cost of searching the nearest-neighbor points is reduced to 22.9 percent of that of the basic search algorithm.

The performance of the BOT test depends on the distribution of distances from queries to nearest-neighbor points. Our method works best when the portion of the number of nearest-neighbor points that are farther than δ

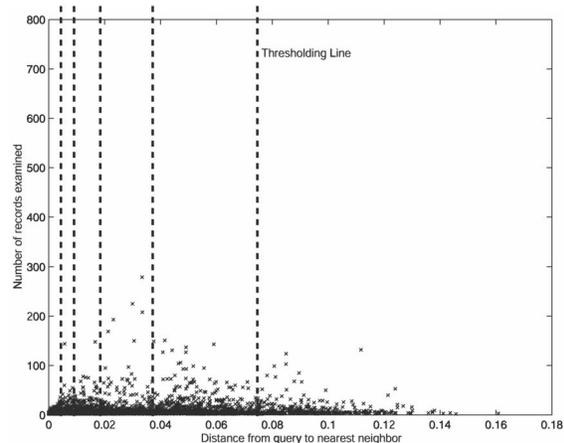


Fig. 18. Relationship between distance from a query to the nearest neighbor and the number of records examined with the BOT test for merging range images.

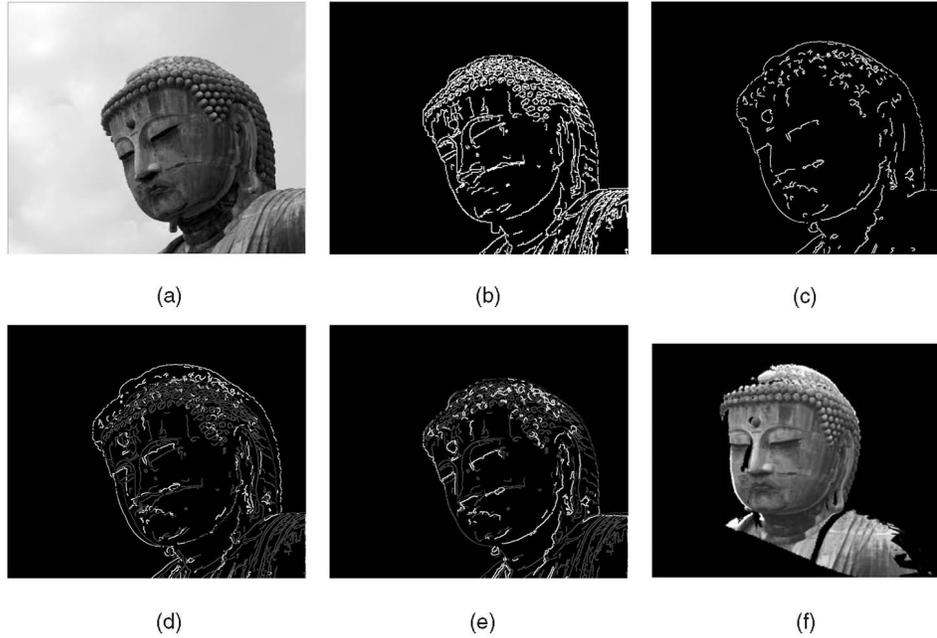


Fig. 19. Aligning a 2D image with a 3D model of the Kamakura Buddha using the photometric attributes of the 3D model.

becomes larger. The BOT test can be applied with the variable threshold δ without recreating the structure of a k-d tree. Thus, the BOT test works efficiently since we subdivide voxels in an octree manner.

6.5 Application: Aligning a Merged Model with a 2D Image

An example of applications utilizing a merged model with reflectance is a 2D-3D registration [40]. Fig. 19 shows an example of aligning a 2D image and a 3D model of the Kamakura Buddha with reflectance. Fig. 19b is the edges of the color values extracted using a Canny filter [59] from the camera image Fig. 19a. Fig. 19c shows the occluding edges and reflectance edges extracted from the 3D model. In Figs. 19d and 19e, the method estimates the posture of the camera by taking matching edges of 2D image (gray lines) and 3D model (white lines). Fig. 19d is the initial posture of camera before iterative computation and the posture converges to Fig. 19e. Finally, texture mapping is accomplished using estimated camera parameter (Fig. 19f).

7 CONCLUSION

In this paper, we have tackled the problem of geometric and photometric modeling large-scale and intricately shaped objects. In modeling such objects, the following new issues occurred: Creating a detailed model from a huge amount of data and merging of reflectance parameters of range images.

For merging a huge amount of range images, we proposed two approaches: the adaptive algorithm of merging range images and a new algorithm for searching for the nearest neighbor using the k-d tree. First, we developed an algorithm for constructing a 3D model in an efficient resolution. Taking into account the surface curvature and the photometric attributes, we constructed

3D models that have higher detail in surface areas that contain high curvature and variety of reflectance parameters. If the nearest-neighbor point is far from a query, the nearest neighbor is not used in extracting a merged mesh model. Thus, we developed the Bounds-Overlap-Threshold test, which approximately searches by pruning branches if the nearest-neighbor point is beyond a threshold. This technique drastically reduces the computational cost if the nearest neighbor is far from a query.

We extended our merging framework to merge reflectance parameters which are attached with range images acquired by a laser range finder. By taking a consensus of the appearance changes of the target object from multiple range images, we reconstructed a 3D model with an appearance which discards outliers due to noise. Also, we were able to provide a model with Lambertian reflected light values by discarding specular reflections as outliers. The reflectance parameters of the model can be used for aligning 2D images with the 3D model surface.

We have been able to successfully construct detailed models using these proposed methods; these models have millions of vertices and triangles. Thus, we can make full use of the power of range finders and can model large-scale and intricately shaped objects using a huge amount of range images.

APPENDIX

ALGORITHM OF BOT TEST

Algorithm 2 shows the algorithm of BOT test, which is written in a recursive manner. N is the node of interest. \mathbf{p} is the query point. d is the distance of the current nearest neighbor. $\text{rightson}(N)$ and $\text{leftson}(N)$ mean the children of node N . $d_{\text{rightson}}(N)$ and $d_{\text{leftson}}(N)$ are the distance from the

query to the boundary of the right/left child of N . The difference from the basic algorithm is illustrated in the gray boxes.

Algorithm 2 SearchNearestNeighborBOT(N, p)

```

Input: Node  $N$ 
Input: Query Point  $p$ 
if  $N$  is leaf node then
  Examine records of  $N$  and compute the smallest  $d$ 
else
  if  $p$  is inside leftson( $N$ ) then
    SearchNearestNeighborBOT(leftson( $N, p$ ))
    if  $d > d_{\text{rightson}(N)} \wedge \delta > d_{\text{rightson}(N)}$  then
      SearchNearestNeighborBOT(rightson( $N, p$ ))
    end if
  else
    SearchNearestNeighborBOT(rightson( $N, p$ ))
    if  $d > d_{\text{leftson}(N)} \wedge \delta > d_{\text{leftson}(N)}$  then
      SearchNearestNeighborBOT(leftson( $N, p$ ))
    end if
  end if
end if
if ball(center  $p$ , radius  $d$ ) is within  $N$  then
  Finish search
end if

```

REFERENCES

- [1] K. Ikeuchi, Y. Sato, K. Nishino, R. Sagawa, T. Nishikawa, T. Oishi, I. Sato, J. Takamatsu, and D. Miyazaki, "Modeling Cultural Heritage through Observation," *Proc. IEEE First Pacific-Rim Conf. Multimedia*, Dec. 2000.
- [2] D. Miyazaki, T. Oishi, T. Nishikawa, R. Sagawa, K. Nishino, T. Tomomatsu, Y. Takase, and K. Ikeuchi, "The Great Buddha Project: Modelling Cultural Heritage through Observation," *Proc. Sixth Int'l Conf. Virtual Systems and MultiMedia*, pp. 138-145, 2000.
- [3] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin, "Building a Digital Model of Michelangelo's Florentine Pietà" *IEEE Computer Graphics & Applications*, vol. 22, no. 1, pp. 59-67, Jan./Feb. 2002.
- [4] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. SIGGRAPH 2000*, pp. 131-144, 2000.
- [5] J.-A. Beraldin, M. Picard, S. El-Hakim, G. Godin, V. Valzano, A. Bandiera, and C. Latouche, "Virtualizing a Byzantine Crypt by Combining High-Resolution Textures with Laser Scanner 3D Data," *Proc. Eighth Int'l Conf. Virtual Systems and MultiMedia*, Sept. 2002.
- [6] F. Stein and G. Medioni, "Structural Indexing: Efficient 3-D Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 125-145, Feb. 1992.
- [7] A. Johnson and M. Hebert, "Surface Registration by Matching Oriented Points," *Proc. Int. Conf. Recent Advances in 3-D Digital Imaging and Modeling*, May pp. 121-128, 1997.
- [8] P. Besl and N. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.
- [9] C. Dorai, G. Wang, A. Jain, and C. Mercer, "From Images to Models: Automatic 3D Object Model Construction from Multiple Views," *Proc. 13th IAPR Int'l Conf. Pattern Recognition*, pp. 770-774, 1996.
- [10] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145-155, Apr. 1992.
- [11] M.D. Wheeler and K. Ikeuchi, "Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 252-265, Mar. 1995.
- [12] P. Neugebauer, "Geometrical Cloning of 3D objects via Simultaneous Registration of Multiple Range Images," *Proc. Int'l Conf. Shape Modeling and Application*, pp. 130-139, Mar. 1997.
- [13] K. Nishino and K. Ikeuchi, "Robust Simultaneous Registration of Multiple Range Images," *Proc. Fifth Asian Conf. Computer Vision ACCV '02*, pp. 454-461, Jan. 2002.
- [14] K. Pulli, "Multiview Registration for Large Data Sets," *Proc. Second Int'l Conf. 3D Digital Imaging and Modeling*, pp. 160-168, Oct. 1999.
- [15] D. Eggert, A. Fitzgibbon, and R. Fisher, "Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering," Technical Report 804, Dept. of Artificial Intelligence, Univ. of Edinburgh, 1996.
- [16] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Toward a General Multiview Registration Technique," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 540-547, May 1996.
- [17] D. Huber and M. Hebert, "Fully Automatic Registration of Multiple 3D Data Sets," *Image and Vision Computing*, vol. 21, no. 7, pp. 637-650, July 2003.
- [18] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," *Proc. SIGGRAPH '94*, pp. 311-318, July 1994.
- [19] M. Soucy and D. Laurendeau, "A General Surface Approach to the Integration of a Set of Range Views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 344-358, Apr. 1995.
- [20] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction algorithm," *Proc. SIGGRAPH '87*, pp. 163-170, 1987.
- [21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Proc. SIGGRAPH '92*, pp. 71-78, 1992.
- [22] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. SIGGRAPH '96*, pp. 303-312 1996.
- [23] A. Hilton, A. Stoddart, J. Illingworth, and T. Winder, "Reliable Surface Reconstruction from Multiple Range Images," *Proc. European Conf. Computer Vision*, pp. 117-126, 1996.
- [24] R. Whitaker, "A Level-Set Approach to 3D Reconstruction from Range Data," *Int'l J. Computer Vision*, vol. 29, no. 3, pp. 203-231, Oct. 1998.
- [25] H.-K. Zhao, S. Osher, and R. Fedkiw, "Fast Surface Reconstruction Using the Level Set Method," *Proc. First IEEE Workshop Variational and Level Set Methods, in conjunction with Proc. ICCV '01*, pp. 194-202, 2001.
- [26] J. Sethian, *Level Set Methods*. Cambridge Univ. Press, 1996.
- [27] M.D. Wheeler, "Automatic Modeling and Localization for Object Recognition," PhD dissertation, School of Computer Science, Carnegie Mellon Univ., 1996.
- [28] M. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus Surfaces for Modeling 3D Objects from Multiple Range Images," *Proc. Int'l Conf. Computer Vision*, Jan. 1998.
- [29] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle, "Robust Meshes from Range Maps," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling*, pp. 205-211, May 1997.
- [30] H.J. Wolfson and I. Rigoutsos, "Geometric Hashing: An Overview," *IEEE Computational Science & Eng.*, vol. 4, no. 4, pp. 10-21, 1997.
- [31] A. Califano and R. Mohan, "Multidimensional Indexing for Recognizing Visual Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 373-392, 1994.
- [32] J. Friedman, J. Bentley, and R. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. Math. Software*, vol. 3, no. 3, pp. 209-226, 1977.
- [33] H. Samet, "The Quadtree and Related Hierarchical Data Structure," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187-260, 1984.
- [34] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 47-54, 1984.
- [35] S. Lavalley and R. Szeliski, "Recovering the Position and Orientation of Free-Form Objects from Image Contours Using 3D Distance Maps," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 378-390, Apr. 1995.
- [36] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *Int'l J. Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.
- [37] M. Greenspan and M. Yurick, "Approximate k-d Tree Search for Efficient ICP," *Proc. 3DIM 2003*, pp. 442-448, 2003.

- [38] R. Sagawa, T. Masuda, and K. Ikeuchi, "Effective Nearest Neighbor Search for Aligning and Merging Range Images," *Proc. 3DIM 2003*, pp. 79-86, 2003.
- [39] R. Sagawa and K. Ikeuchi, "Taking Consensus of Signed Distance Field for Complementing Unobservable Surface," *Proc. 3DIM 2003*, pp. 410-417, 2003.
- [40] R. Kurazume, K. Nishino, Z. Zhang, and K. Ikeuchi, "Simultaneous 2D Images and 3D Geometric Model Registration for Texture Mapping Utilizing Reflectance Attribute," *Proc. Fifth Asian Conf. Computer Vision*, vol. 1, pp. 99-106, Jan. 2002.
- [41] I. Stamos and P. Allen, "Registration of 3D with 2D Imagery in Urban Environments," *Proc. Eighth Int'l Conf. Computer Vision*, 2001.
- [42] P. Neugebauer and K. Klein, "Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views," *Proc. Eurographics '99*, pp. 245-256, 1999.
- [43] J. Foley, A. van Dam, S. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice in C*, second ed. Addison Wesley Professional, 1995.
- [44] G. Nielson and B. Hamann, "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes," *Proc. Visualization '91*, pp. 83-91, 1991.
- [45] M. Garland and P. Heckbert, "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics," *Proc. IEEE Visualization*, 1998.
- [46] H. Hoppe, "Progressive Meshes," *Computer Graphics (Proc. SIGGRAPH 1996)*, pp. 99-108, 1996.
- [47] H. Hoppe, "New Quadric Metric for Simplifying Meshes with Appearance Attributes," *Proc. IEEE Visualization '99*, pp. 59-66, 1999.
- [48] R. Shekhar, E. Fayyad, R. Yagel, and J. Cornhill, "Octree-Based Decimation of Marching Cubes Surfaces," *Proc. Visualization '96*, pp. 335-342, 1996.
- [49] R. Shu, Z. Chen, and M. Kankanhalli, "Adaptive Marching Cubes," *The Visual Computer*, vol. 11, pp. 202-217, 1995.
- [50] S. Frisken, R. Perry, A. Rockwood, and T. Jones, "Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics," *Proc. SIGGRAPH 2000*, pp. 249-254, July 2000.
- [51] S.F.F. Gibson, "Using Distance Maps for Accurate Surface Representation in Sampled Volumes," *Proc. IEEE Symp. Volume Visualization*, pp. 23-30, 1998.
- [52] R. Sagawa, T. Oishi, A. Nakazawa, R. Kurazume, and K. Ikeuchi, "Iterative Refinement of Range Images with Anisotropic Error Distribution," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, pp. 79-85, Oct. 2002.
- [53] Cyra Technologies, Inc. <http://www.cyra.com>. 2004.
- [54] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-Texture Method: Appearance Compression Based on 3D Model," *Proc. Computer Vision and Pattern Recognition '99*, vol. 1, pp. 618-624, Jun. 1999.
- [55] R. Sagawa, K. Nishino, M. Wheeler, and K. Ikeuchi, "Parallel Processing of Range Data Merging," *Proc. 2001 IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, vol. 1, pp. 577-583, Oct. 2001.
- [56] R. Sagawa, "Geometric and Photometric Merging for Large-Scale Objects," PhD dissertation, Graduate School of Eng., The Univ. of Tokyo, 2003.
- [57] The Stanford 3D Scanning Repository, <http://www-graphics.stanford.edu/data/3Dscanrep/>. 2004.
- [58] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167-174, June 1998.
- [59] F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.



Ryusuke Sagawa received the BE degree in information science from Kyoto University, Kyoto, Japan, in 1998. He received the ME degree in information engineering in 2000, and the PhD degree in information and communication engineering from the University of Tokyo, Tokyo, Japan in 2003. He is a research associate at the Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan. His primary research interests are computer vision, computer graphics and robotics (mainly geometrical modeling and visualization). He is a member of the IEEE and the IEEE Computer Society.



Ko Nishino received the BE and ME degrees from The University of Tokyo in 1997 and 1999, respectively. He received the PhD degree in information science from The University of Tokyo in 2002. Since 2002, he has been a postdoctoral research scientist at Columbia University. His research interests span computer vision and computer graphics. He has published papers on photometric and geometric problems in scene and object modeling that include physics-based vision, image-based modeling, and rendering. He is a member of the IEEE and ACM.



Katsushi Ikeuchi received the BE degree from Kyoto University in 1973 and the PhD degree from the University of Tokyo in 1978. After working at the MIT AI Laboratory for three years, ETL for five years, and CMU Robotics Institute for 10 years, he joined the University of Tokyo in 1996, and is currently a full professor. His research interest spans computer vision, robotics, and computer graphics. In these research fields, he has received several awards, including the David Marr Prize in computational vision for the paper "Shape from Interreflection," and IEEE R&A K.-S. Fu memorial best transaction paper award for the paper "Toward Automatic Robot Instruction from Perception—Mapping Human Grasps to Manipulator Grasps." In addition, in 1992, his paper, "Numerical Shape from Shading and Occluding Boundaries," was selected as one of the most influential papers to have appeared in the *Artificial Intelligence Journal* within the past 10 years. His IEEE activities include general chair, IROS '95, ITSC '00, IV '01; program chair, CVPR '96, ICCV '03; associate editor, *IEEE Transactions on Robotics and Automation*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*; distinguished lecture SPS (2000-2002), RAS (2004-2006). Dr. Ikeuchi was elected as an IEEE fellow in 1998. He is the editor-in-chief of the *International Journal of Computer Vision*.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.