ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

Reconstruction with 3D Geometric Bilateral Filter

A. Miropolsky and A. Fischer

Laboratory for Computer Graphics and CAD, Technion -Israel Institute of Technology, Haifa, Israel

Abstract
In recent years, reverse engineering (RE) techniques have been developed for surface reconstruction from 3D scanned data. Typical sampling data, however, usually is large scale and contains unorganized points, thus leading to some anomalies in the reconstructed object. To improve performance and reduce processing time, Hierarchical Space Decomposition (HSD) methods can be applied. These methods are based on reducing the sampled data by replacing a set of original points in each voxel by a representative point, which is later connected in a mesh structure. This operation is analogous to smoothing with a simple low- pass filter (LPF). Unfortunately, this principle also smoothes sharp geometrical features, an effect that is not desired. The high performance results of bilateral filtering for removing noise from 2D images while preserving details motivated us to extend this filtering and apply it to 3D scan points. This paper introduces anisotropic 3D scan point filtering, which we have defined as 3D Geometric Bilateral Filtering (GBF). The proposed GBF method smoothes low curvature regions while preserving sharp geometric features, and it is robust, simple and fast.

Categories: Hierarchical Space Decomposition, surface reconstruction, bilateral filtering.

1. Introduction

1.1. Motivation

Typical scan data is very large scale and noisy, making the reconstruction process time consuming and inaccurate. In order to overcome these problems, the data must be reduced and denoised. The denoising process should be fast and should preserve sharp features. Methods described in the literature capable of denoising the sampled data have been applied only to the reconstructed mesh. Removing the noise from the sampled points directly and reducing the data while preserving the shape of the sampled surface, particularly its fine details, still remains a challenge. Explicit information regarding sharp features is usually missing from the cloud of points, thus presenting an obstacle. Scanning technologies that provide additional information regarding the surface normal associated with every sampled point can be used in order to overcome the above problem. This paper describes a new method that utilizes normal and position information regarding the sampled points for reducing and denoising the data, while preserving features. A detailed description of the proposed method is given in the approach section.

1.2. Problem statement

This work addresses the problem of surface reconstruction from large-scale noisy sampled data. In order to overcome this problem, this paper proposes data and sampling noise reducing by applying *edge-preserving filtering* on scanned points. The desired result is an approximation of the surface points, while preserving critical features such as sharp edges and corners common in mechanical parts. The properties required from the *edge-preserving filter* are as follows:

- It should be fast.
- It has to preserve high curvature features, such as sharp edges and corners.
- It has to remove noise, providing a good estimation of underlying sampled surface.

2. Overview

Following is an overview of the state-of-the-art approaches for surface reconstruction and noise filtering methods.

2.1. Surface reconstruction

Computational geometry studies have focused on piecewise-linear interpolation of unorganized points. Most of the algorithms in this field [AB98][ACK01][Boi84][DGH01] define a reconstructed surface as a carefully chosen sub-set of Delaunay triangulation in R^3. The data point interpolation approach provides high performance for noiseless data. If the noise is significant, however, approximating the surface is preferable to interpolating it through the data points.

In the fields of *computer graphics and computer aided geometric design*, methods constrain the surface to traverse between the points, while the maximal error is bounded by the user. A well-known reconstruction algorithm of Hoppe [HDD*92] works effectively on real test cases and objects with arbitrary topology. However, the method uses the Marching Cubes algorithm [LC87], and the produced meshes suffer from nearly singular triangles and deficient approximation of sharp features. Curless [CL96] proposes a volumetric approach which exploits the fact that the cloud of points is a collection of laser range images. Unfortunately, this assumption restricts the applicability of the system to objects scanned by laser range devices. Azernikov [AF03] proposes a volumetric method that approximates the surface by filtering the set of original points with a *low-pass* filter. Thus, the process becomes more robust and stable with respect to sample noise. But this approach also leads to suppression of fine details and sharp features.

A number of works use normal direction information to significantly improve surface approximation. Bernardini [BMR*99] proposed a Delaunay triangulation-based method called the *ball-pivoting* algorithm. This method interpolates the data points and this becomes a drawback when the data contains a significant amount of noise. Azernikov [AMF03] extended the above mentioned volumetric approach with respect to normal information. This extension enables the reconstruction of object

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

families, such as thin flat parts. However, this approach increases the noise sensitivity of the method with respect to noise in the normal domain.

The above discussion shows that appropriate sample noise reduction plays an important role in surface reconstruction. Noise filtering techniques are summarized in the following section.

2.2. Noise filtering

Noise filtering is part of ongoing research in image processing and computer vision. The state-of-the-art approaches in this field have focused on methods for *edge-preserving* noise removal. Tomasi [TM98] introduced a non-iterative scheme for *edge-preserving* smoothing, called *bilateral filtering*. The idea underlying this method is to refer to similarity in the range domain of an image in the way that traditional filters refer to closeness in the spatial domain.

Denoising methods for 3D scan data are naturally based on image denoising approaches. Most efforts have been toward reducing noise after the mesh has been created. Taubin [Tau00] introduced *signal processing* on surfaces by defining the Laplacian operator on meshes. Fleishman [FDC03] proposed the *bilateral-mesh-denoising* method. The problem of these denoising algorithms is that they still require preprocessing such as mesh generation. Moreover they are strictly dependent on connectivity information provided by the mesh, where the mesh generation process itself suffers from noise.

In order to overcome this problem we have developed a new 3D method based on *bilateral filtering*, which is applied directly on the scanned points. The next section gives a detailed description of the method.

3. The approach

This paper conforms to the surface reconstruction method proposed by Azernikov [AMF03]. The approach utilizes the *Hierarchical Space Decomposition Model (HSDM)*, based on Octree data structure, where the set of scan points in each voxel is replaced by approximated representative surface point, which is then connected in a mesh structure. Operations applied to a set of points in each voxel are analogous to smoothing operations with *low-pass* filter (*LPF*). The assumption is that in low curvature regions surface point coordinate values change slowly, and neighbor points are likely to have similar coordinates. Therefore, it is appropriate to simply average them together. The noise points are mutually less correlated than are the signal ones, so noise is extracted while the signal is preserved. However, the assumption of slow spatial variations fails at the edges, which are consequently blurred by low-pass filtering. Thus, 3D *edge-preserving filtering* is required in order to prevent averaging across high curvature regions or edges, while still smoothing the areas with low curvature.

This paper introduces a *3D Geometric Bilateral Filter (GBF)* method for *edge-preserving* data reduction and denoising. The method is based on normals information. It calculates a voxel value as a weighted average of the scanned points encapsulated in the voxel, while the weights are defined as a function of both spatial location and normal value of the points. Figure 1 illustrates the complete *GBF* reconstruction process.

Unlike previously described denoising methods applied on the reconstructed mesh, the proposed method enables noise reduction directly from sampled points during surface reconstruction, thus facilitating the process. Moreover, *GBF* enables proper data reduction, thus significantly improving the execution time of the entire reconstruction algorithm. The advantages of the proposed method can be summarized as follows:

- Reduces sampling noise.
- Preserves sharp features.
- Reduces data, leading to fast reconstruction.
- Is simple to implement.

In the following discussion, Sections 4.1-4.2 describe the technique for implementing the *3D Geometric Bilateral Filter*. In Section 5, the feasibility of the method is demonstrated on complex objects, and Section 6 discusses the properties of the proposed filtering method. Section 7 provides a summary and conclusions.

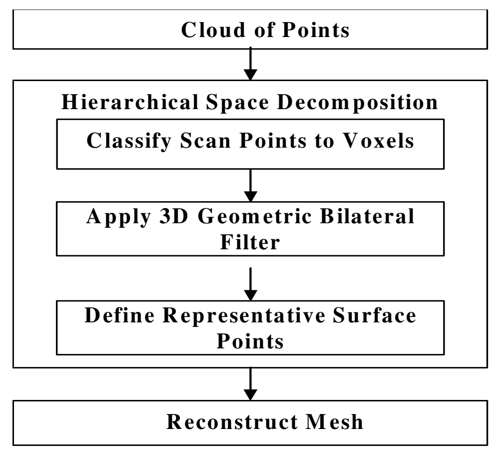


Figure 1: *Reconstruction process with GBF method.*

4. Implementation

4.1. 3D Geometric Bilateral filter

The *GBF* method follows the 2D *bilateral filtering* technique in *image processing*. However the problem is different in the case of a 3D image. The desired value we seek is the spatial location of surface point **s** and not the surface normal at this point, which is analogous to the intensity value in image processing.

In order to initialize the process, the approximated position of point **s** is calculated by defining the centroid of the sampled points encapsulated in the voxel. For each sampled point in the voxel, its relative distance from the centroid is calculated. These distances are used to define the *position closeness* weights of the points. The *centroid normal* $\mathbf{N_s}$ is analogous to the intensity value $\mathbf{I_s}$ in the 2D case. For each sampled point, the difference of its normal relative to $\mathbf{N_s}$ is calculated, to be used later for defining the *normal similarity* weight of the point. The normal spectrum of the scan points is obtained by mapping all point normals to a Gauss sphere, and the *centroid normal* is then calculated as a simple average of this spectrum. The spatial stop function **f( )** determines the point weight according to its closeness to the centroid in the spatial domain, and the edge-stopping function **g( )** refers to the similarity between the point normal and the centroid normal in the vector domain. Thus, the output of the *Geometric Bilateral Filter (GBF)* for a voxel **V** can be defined as follows (equation (1)):

$$I_s = \frac{1}{k(s)} \sum_{p \in V} f[d(p,s)] \times g[\delta(N_p, N_s)] \times I_p \qquad (1)$$

where: p – index of scanned point, encapsulated in the voxel; s – index of the centroid; V – 3D voxel space; I_s – spatial location of the representative point in the voxel; I_p – spatial location of sampled point p, encapsulated in the voxel; f( ) – spatial stop function; g( ) – edge-stopping function; N_p – surface normal at point p; N_s – centroid normal; d(p,s) - the Euclidian distance between point p and centroid s; $\delta(p,s) = \langle N_p, N_s \rangle$-inner product of normal vectors describing the difference between normals of point p and centroid s; k(s)—normalization factor given by equation (2):

$$k(s) = \sum_{p \in V} f[d(p,s)] \times g[\delta(N_p, N_s)] \tag{2}$$

Consider now the case of a scanned sharp edge, as shown in Figure 2. This case is very similar to the *uncertainty problem* described by Durand [DD02] as corresponding to pixels where there is not enough information in the neighborhood to decouple large-scale and small-scale intensity changes. In this case, we are unable to separate the points with large-scale and small-scale variations of normals. In Figure 2(a) all point normals are equally different from the centroid normal. As a result, all points get the same *normal similarity* weight, and the output of the *GBF* will be the same points centroid, which is analogous to the *LPF*. In this case, the previously proposed bilateral filtering technique described by equation (1) fails, and the sharp edge is finally smoothed.

Our strategy for solving the *uncertainty problem* is as follows. First we propose to map the original points to a sphere with its center at the centroid. This mapping is accomplished by simple projection of the points to the sphere along the vector from a point to the centroid, as shown in Figure 2(b). Such mapping can be seen as an approximation of sharp edge/corner to smooth surface, where the *Geometric Bilateral Filter* should work well. The normals of the map points are defined as unit vectors from centroid to original points. Since they are located on the sphere, all the map points are equidistant from the centroid as the center of the sphere. Thus, all the map points receive the same spatial weight, which means that the spatial stop function no longer plays any important role and therefore can be ignored. The edge-stopping function g( ), however, is still essential, but it now refers to mapped normals instead of original normals.

The geometric bilateral filter is therefore reduced to the following equation:

$$I_s = \frac{1}{k(s)} \sum_{p \in V} g[\delta(n_p, N_s)] \times I_p \tag{3}$$

$$k(s) = \sum_{p \in \Omega} g[\delta(n_p, N_s)] \tag{4}$$

where $\mathbf{n_p}$ – normal at map point corresponding to original point p. Thus, the algorithm of the proposed method is based on the following steps:

(1) Construct *HSDM* from a cloud of points.
(2) Calculate the centroid **s** in each voxel as a simple average of the scanned points encapsulated in it.

For each voxel:

(3) Calculate the centroid normal **N**.
(4) Calculate the mapped normal **Nm** for each scanned point.
(5) Calculate the weight $\mathbf{w_i}$ for each scanned point according to dissimilarity **δ** of its mapped normal with respect to centroid normal.
(6) Calculate the representative surface point **RepPoint** as a weighted average of scanned points.

The following is the pseudo-code for applying a *GBF* to a single voxel:

```
GBF (scannedPoints{qi}, centroid s)
    N = centroid->Normal;
    RepPoint = new Point3D;
    for i = 1 to sizeOf( {qi} )
        mN = new Vector3D(s,      qi);
        normalize(mN);
        δ = ScalarProduct(mN, N);
        wi  = g(δ);
        RepPoint += wi * qi;
        K += wi;
    end;
    return RepPoint/K;
```

Although only the *normal similarity* function **g( )** exists, the *GBF* filter remains bilateral due to new mapped normals that refer implicitly to the spatial location of original points. The new normals $\mathbf{n_p}$ are compared to the centroid normal $\mathbf{N_s}$, so that the normal information is respected as well.

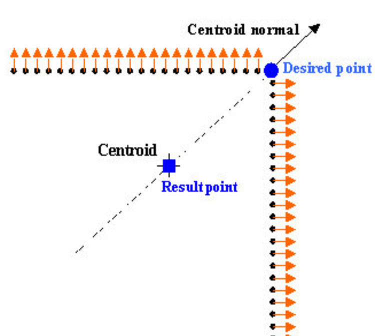
4.2. 3D Geometric bilateral filter as a robust statistical estimator

Durand [DD02] showed that the *bilateral filter* belongs to the group of robust statistical estimators. Thus, in our case the proposed *3D Geometric Bilateral Filter (GBF)* can be seen as a robust statistical 1-step estimator that correctly estimates the underlying sampled surface. Based upon the discussion on desirable properties of *edge-stopping* function in [DD02], we can assume that the most robust results for *GBF* will yield Gaussian (equation (5)) and Tukey's biweight (equation (6)) functions.

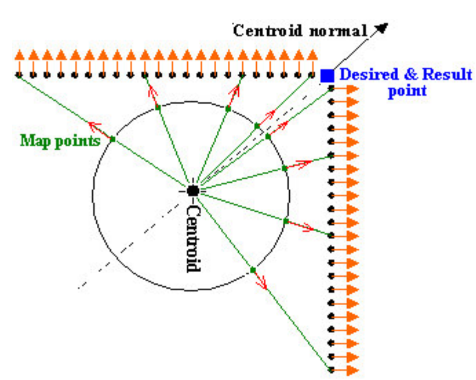
$$g(x) = e^{-\frac{x^2}{2\sigma_r^2}} \tag{5}$$

$$g(x) = \begin{cases} \frac{1}{2}\left[1 - (\frac{x}{\sigma_r})^2\right]^2 & |x| \le \sigma_r \\ 0, & otherwise \end{cases} \tag{6}$$

We used the Gaussian *edge-stopping* function for all our experiments.



(a)



(b)

Figure 2: *(a) The uncertainty problem; (b) The solution.*

5. Examples

We have demonstrated the feasibility of the proposed *GBF* method on several complex models. Figure 3 shows the synthetic model with added noise, Figure 4 refers to a scanned toy airplane an, Figure 5 and Figure 6 present respectively the Turbine Blade and Skeleton Hand models, which were downloaded from the Large Geometric Models Archive of Georgia Institute of Technology [LGMA]. The Hausdorff distance between the data points and the reconstructed mesh, normalized by a bounding cuboid size, was chosen as an error estimator. The error can be controlled by the voxel size requirement.

Table 1 presents the performance of the proposed method on an Intel Pentium 4, 1.6GHz, 523Mb RAM.

Object	n	d	Max e [%]	t [sec]
Mechanical part	164425	6	1.5	14.3
Airplane toy	117152	7	0.7	21.8
Skeleton Hand	327323	7	0.7	23.3
Turbine Blade	882954	7	0.7	172.8

Table 1: *Performances of the proposed method: n – number of sampled points, d – Octree depth, e – error, t – execution time*

6. Discussion

In this section, some of the properties of the proposed *3D Geometric Bilateral Filtering (GBF)* method are discussed.

(1) The only parameter that has to be provided for *GBF* is the normal spread threshold σ_r. This threshold is a scale parameter in the normal domain that defines the smoothing level. This parameter defines the robustness of the filter since it determines which points are considered outliers, thus filtering them away. The distribution of scanned points is quite different for each voxel. Therefore it is impossible to set a global value for the σ_r parameter. Thus, it cannot be controlled by the user and should be adapted for every voxel. It can initially be defined arbitrarily and then be self-tuned according to the normalization factor **k**. We chose the normalization factor **k** to be close to one, and thus the appropriate threshold σ_r is adjusted for each voxel to achieve the desired result. This reference value **k = 1** performed consistently well for all our experiments. Although *bilateral filtering* is considered a non-iterative method, the proposed *GBF* filter may require two iterations for every voxel due to the adaptive nature of the σ_r threshold.

(2) It was mentioned above that the proposed technique is based on additional information regarding surface normals, thus restricting the method to a special kind of scan technology. However, this information is used only for calculation of the centroid normal. Once this normal can be approximated by other means, the proposed method can be applied without dependence upon the scanning technology.

(3) It should be emphasized that the performance of the proposed filtering method is strictly dependent on the density of the scanning data. In general, a geometric feature can be reliably recovered if its size is much bigger than the scanning interval. This is a well-known requirement and fine quality features are usually scanned with high density. However, high density scanning causes redundant data in low curvature regions. Thus the proposed filtering method reduces redundant data while preserving the essential reconstruction information.

7. Summary and Conclusion

This paper has proposed a new fast *3D Geometric Filtering* method for data reduction and noise removal that can be applied directly on scanned points during mesh reconstruction. The proposed filtering method utilizes surface normal information for preserving fine details of the sampled surface. The developed method uses the HSDM model, providing data reduction and a robust statistical estimation of the underlying sampled surface. The proposed method is simple, fast and accurate. The feasibility of the method is demonstrated on complex objects.

References:

[AB98] AMENTA N., BERN M.W.: Surface reconstruction by Voronoi filtering. *Symposium on Computational Geometry* (1998), 39-48.

[ACK01] AMENTA N., CHOI S., KOLLURI R.: The power crust. *6th ACM Symposium on Solid Modeling and Applications* (2001), 249-260.

[AF03] AZERNIKOV S., FISCHER A.: Surface reconstruction of freeform objects based on hierarchical space decomposition. *4th Israel-Korea Conference on Geometric Modeling and Computer Graphics* (2003), 111-116.

[AMF03] AZERNIKOV S., MIROPOLSKY A., FISCHER A.: Surface Reconstruction of Freeform Objects Based on Multiresolution Volumetric Method. *8th ACM Symposium on Solid Modeling and Applications* (2003), 115-126.

[BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 5* (1999), 4, 349–359.

[Boi84] BOISSONAT J. D.: Representing 2D and 3D shapes with the Delaunay triangulation. *7th International Conference on Pattern Recognition* (Montreal, Canada, July 30-Aug. 2, 1984), IEEE, 1984, 745-748.

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. *Computer Graphics 30* (Annual Conference Series-1996), 303–312.

[DGH01] DEY T. K., GIESEN J., HUDSON J.: Delaunay based shape reconstruction from large data. *2001 Symp. on Parallel and Large-Data Visualization and Graphics* (2001), 19–27, 146.

[DD02] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics 21* (2002), 3, 257-266.

[FDC03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *Proceedings of SIGGRAPH* (2003), 950-953.

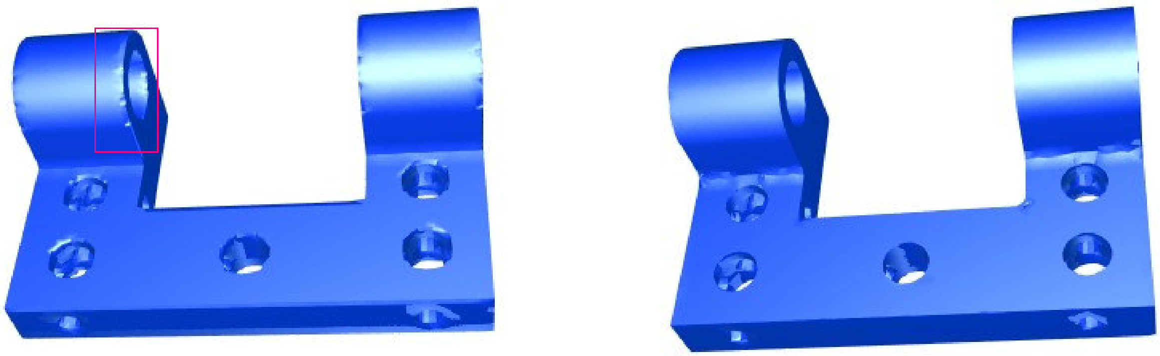
[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *Computer Graphics 26* (1992), 2, 71–78.

[LGMA] Large Geometric Models Archive: http://www.cc.gatech.edu/projects/large_models/.

[LC87] LORENSEN W., CLINE H.: Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics 21* (1987), 4, 163–169.

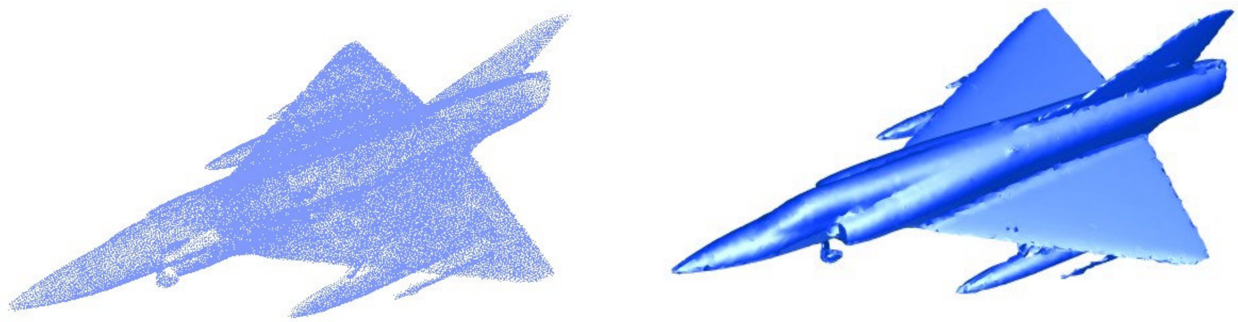
[Tau00] TAUBIN G.: Geometric signal processing on polygonal meshes. *Eurographics* State of the Art Report, 2000.

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. *Proc. IEEE Int. Conf. on Computer Vision* (1998), 836-846.



(a) (b)

Figure 3: *Mechanical detail: (a) Reconstructed mesh with LPF method; (b) Reconstructed mesh with GBF method.*

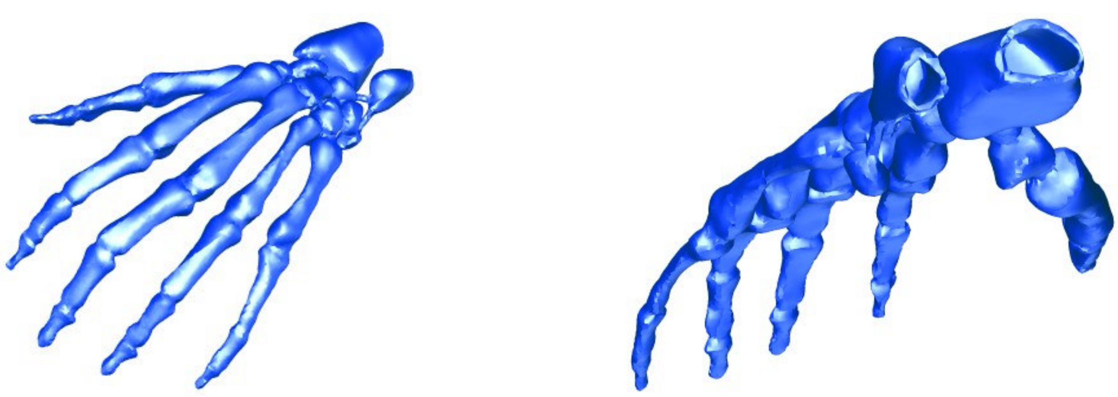


(a) (b)

Figure 4: *Toy airplane: (a) Cloud of scanned points; (b) Reconstructed mesh with GBF method.*

(a) (b)

Figure 5: *Turbine Blade: Reconstructed mesh with GBF method (a) Back view; (b) Front view.*



(a) (b)

Figure 6: *Skeleton Hand: Reconstructed mesh with GBF method (a) Front view; (b) Back view.*