I N F O R M A T I K

Sparse Meshing of Uncertain and Noisy Surface Scattered Data

Oliver Schall Alexander G. Belyaev
Hans-Peter Seidel

MPI–I–2005–4–002 February 2005

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Authors' Addresses

Oliver Schall
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
`schall@mpi-sb.mpg.de`

Alexander G. Belyaev
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
`belyaev@mpi-sb.mpg.de`

Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
`hpseidel@mpi-sb.mpg.de`

Abstract

In this paper, we develop a method for generating a high-quality approximation of a noisy set of points sampled from a smooth surface by a sparse triangle mesh. The main idea of the method consists of defining an appropriate set of approximation centers and use them as the vertices of a mesh approximating given scattered data. To choose the approximation centers, a clustering procedure is used. With every point of the input data we associate a local uncertainty measure which is used to estimate the importance of the point contribution to the reconstructed surface. Then a global uncertainty measure is constructed from local ones. The approximation centers are chosen as the points where the global uncertainty measure attains its local minima. It allows us to achieve a high-quality approximation of uncertain and noisy point data by a sparse mesh. An interesting feature of our approach is that the uncertainty measures take into account the normal directions estimated at the scattered points. In particular it results in accurate reconstruction of high-curvature regions.

Keywords

surface reconstruction, scattered point cloud data, uncertainty

1 Introduction

Surface reconstruction from scattered point data is currently a subject of intensive research. The problem was probably first considered in [8] and became increasingly popular after [21]. These papers established two main surface reconstruction approaches: Delaunay-based methods (see, for example, [1–3, 5, 7, 13, 17, 19] and a recent review in [16]) where the reconstructed surface is usually approximated by a subcomplex of the Delaunay complex of scattered data [1] and volumetric methods (for instance, [6, 9, 25, 30]) which reconstruct the surface as the zero-level set of a 3D function. Other methods include zippering meshes obtained from range images [31] and statistical learning [22, 34].

Recent research trends in surface reconstruction from scattered data include dealing with scattered polygonal data [23, 29], processing noisy point data [10, 18, 26], surface reconstruction from incomplete data [15, 23, 28, 33], and processing uncertain data [4, 27].

Once a dense polygonal mesh is generated from scattered point data, usually the next task of the surface reconstruction pipeline consists of accurately approximating the mesh by a coarse high-quality mesh [12, 32].

In this work, we propose a method which combines two surface reconstruction steps together. Given real world scattered point data (usually noisy and uncertain) digitized from a 3D model, the method generates a sparse triangle mesh which delivers an accurate approximation of the surface of the model.

Given a set of points $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ scattered over an unknown surface in 3D and provided with confidence measures $\mathcal{W} = \{w_1, \dots, w_N\}$ ($w_i \in [0, 1]$, if w_i is not given we set $w_i = 1$), we associate with each $\mathbf{p}_i$ a local uncertainty function. We compose a global uncertainty function from local ones and determine its argminima: $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_M\}$. These argminima form a sparse set of 3D points, $M \ll N$, and we use them as the vertices of a sparse mesh $\mathcal{M}$ approximating $\mathcal{P}$. Once the set of approximation centers $\mathcal{Q}$ is generated we use a standard meshing technique to build $\mathcal{M}$ from $\mathcal{Q}$. Finally a simple cleaning procedure is applied to $\mathcal{M}$ for removing few spikes and small disconnected components.

Our approach is inspired by the kernel density estimation method (Parzen-window estimation method) and we use a procedure similar to the Mean Shift technique [11, 14, 20] for detecting the approximation centers.

For the sake of simplicity, we assume that points $\mathcal{P}$ are equipped with unit normal vectors $\mathcal{N} = \{\mathbf{n}_1, \dots, \mathbf{n}_N\}$ which determine normal directions at the scattered points. Often such set of point normals is included in the standard output of 3D digitizing devices. We do not require a consistent orientation of $\mathcal{N}$. If point normals are not initially given, they can be easily

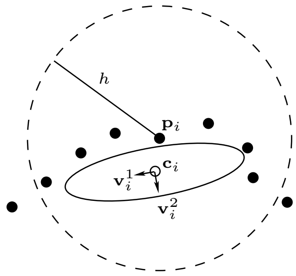


Figure 1: Illustration of the ellipsoidal kernel computation for the scattered data point $\mathbf{p}_i$.

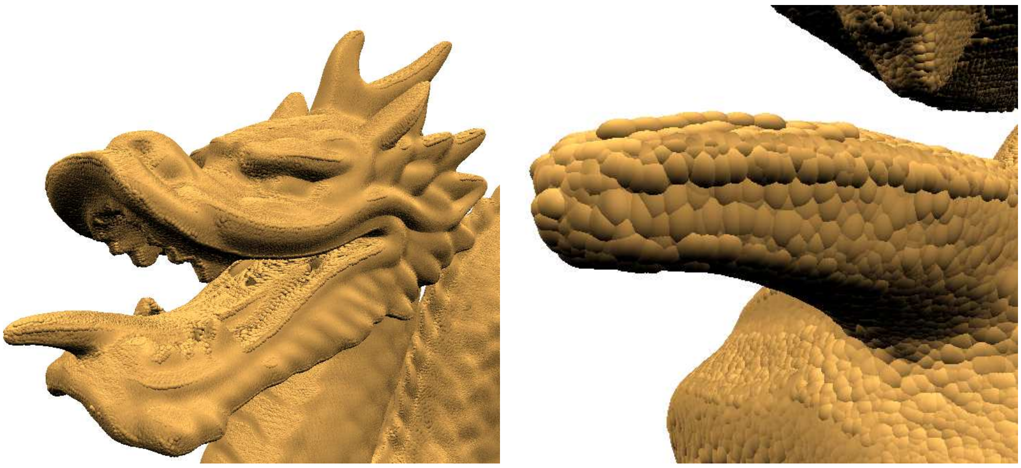


Figure 2: Left: The Stanford Dragon head visualized using its ellipsoidal kernels. Right: Zoomed tongue region of the left image. It is apparent that the ellipsoids elongate according to the distribution of the input points.

generated by Principal Component Analysis of $\mathcal{P}$.

2 Determining Approximation Centers

With each point $\mathbf{p}_i \in \mathcal{P}$ we associate a covariance matrix

$$\mathbf{C}_i = \sum_{j=1}^{N} (\mathbf{p}_j - \mathbf{c}_i)(\mathbf{p}_j - \mathbf{c}_i)^T \chi\left(\frac{||\mathbf{p}_j - \mathbf{p}_i||}{h}\right) \tag{1}$$

where $\chi(r)$ is a cut-off function equal to one if $r \leq 1$ and vanishing if $r > 1$, h is a user-specified parameter. Thus to compute $\mathbf{C}_i$ we clamp all input points outside a sphere of radius h, centered at $\mathbf{p}_i$ (see Figure 1 for illustration). Let $\mathbf{c}_i$ be the center of gravity of all input points inside the sphere. Since $\mathbf{C}_i$ is

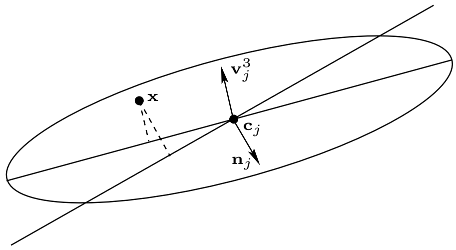


Figure 3: Uncertainty is measured using the distances to the tangent planes defined by the smallest semi-axis of the ellipsoid and the point normal.

symmetric and positive semi-definite, its eigenvalues λ_i^l, $l = 1, 2, 3$, are real-valued and non-negative: $0 \leq \lambda_i^3 \leq \lambda_i^2 \leq \lambda_i^1$. Furthermore, the corresponding eigenvectors $\mathbf{v}_i^l$ form an orthonormal basis. Thus the covariance matrix (1) defines the ellipsoid

$$E_i(\mathbf{x}) : (\mathbf{x} - \mathbf{c}_i)^T \mathbf{C}_i^{-1} (\mathbf{x} - \mathbf{c}_i) \leq 1, \tag{2}$$

elongated along $\mathbf{v}_i^1$ direction. Figure 2 shows the ellipsoidal kernels for the Stanford dragon head.

Once the ellipsoidal kernels are computed, we define the global uncertainty function by

$$F(\mathbf{x}) = \sum_{\{j\}:E_j(\mathbf{x})\leq 1} w_j \Phi\left(\frac{(\mathbf{x}-\mathbf{c}_j)\cdot\mathbf{v}_j^3}{\sqrt{\lambda_j^3}}\right) \cdot \left[(\mathbf{x} - \mathbf{c}_j) \cdot \mathbf{n}_j\right]^2, \tag{3}$$

where w_i is the confidence measure associated with $\mathbf{p}_i$ and $\Phi(\cdot)$ is a Gaussian-like function whose tails are splined to zero

$$\Phi(x) = \begin{cases} e^{-8x^2} & \text{if } 0 \leq x \leq 0.5 \\ 16/e^2(1-x)^4 & \text{if } 0.5 < x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Notice how (3) takes into account the normal directions $\{\mathbf{n}_i\}$ estimated at the scattered points. See also Figure 3 illustrating geometric ideas behind (3).

To find the local minima of $F(\mathbf{x})$ we use a procedure similar to a gradient-descent minimization. We freeze the weighting function Φ since it changes slowly and approximate $\nabla F(\mathbf{x})$ by

$$2 \sum_{\{j\}:E_j(\mathbf{x})\leq 1} w_j \Phi\left(\frac{(\mathbf{x}-\mathbf{c}_j)\cdot\mathbf{v}_j^3}{\sqrt{\lambda_j^3}}\right) \cdot \left[(\mathbf{x} - \mathbf{c}_j) \cdot \mathbf{n}_j\right] \cdot \mathbf{n}_j. \tag{5}$$

(In the following we often abbreviate the subindex to keep the formulas more readable.)

Choosing the adaptive step-size

$$\tau = \frac{1}{2\sum_{\{j\}} w_j \Phi\left(\frac{(\mathbf{p}_i^k - \mathbf{c}_j)\cdot \mathbf{v}_j^3}{\sqrt{\lambda_j^3}}\right)} \tag{6}$$

we arrive at the following iterative procedure

$$\mathbf{p}_i^0 = \mathbf{p}_i \quad , \quad \mathbf{p}_i^{k+1} = \mathbf{p}_i^k - \mathbf{m}_i^k \tag{7}$$

$$\mathbf{m}_i^k = \frac{\sum_{\{j\}} w_j \Phi\left(\frac{(\mathbf{p}_i^k - \mathbf{c}_j)\cdot \mathbf{v}_j^3}{\sqrt{\lambda_j^3}}\right) \cdot \left[(\mathbf{p}_i^k - \mathbf{c}_j)\cdot \mathbf{n}_j\right] \cdot \mathbf{n}_j}{\sum_{\{j\}} w_j \Phi\left(\frac{(\mathbf{p}_i^k - \mathbf{c}_j)\cdot \mathbf{v}_j^3}{\sqrt{\lambda_j^3}}\right)} \tag{8}$$

which resembles the Mean Shift technique [11, 14, 20].

We choose the input points $\mathcal{P}$ or a random subset of $\mathcal{P}$ as start points for the gradient descent. This choice is sensible as the noisy input points lie near to the smooth surface we want to approximate. We stop the iterative process if

$$||\mathbf{p}_i^{k+1} - \mathbf{p}_i^k|| < 10^{-4}h, \tag{9}$$

where h is the kernel size used for the ellipsoidal kernel computation. In our experiments we choose h in the order of ten times the average sampling density of the given scattered data.

Figure 4 shows that the start points converge to a smooth surface. Furthermore, the image illustrates the clustering property of our algorithm. Several start points converge to the same local minimum of the defined global uncertainty function. Since it is not necessary that several points represent one minimum we calculate one approximation center for them using standard Mean Shift (see Figure 5). This is done by centering a spherical kernel with size t at a random converged point. Then the kernel is moved by applying Mean Shift iterations until it convergences at the centroid of the cluster. After this all points in the kernel are clustered to the kernel center. This process is iterated until all points are clustered. We found out that it is a good choice to select the kernel size t adapted to the average sampling density of the noisy input data. Thus, we are able to represent a smooth surface approximating the noisy scattered input data using much fewer approximation centers than given input points.

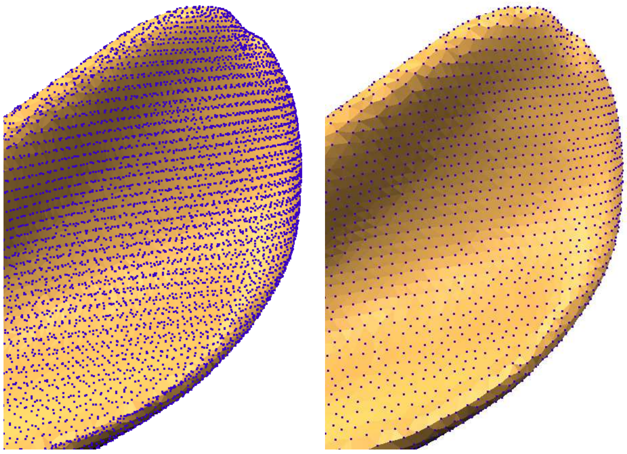


Figure 4: Left: Zoom of the Stanford bunny ear showing that the input points converge to a smooth surface. Right: The reconstructed bunny ear after the clustering procedure. The noise is removed and high-curvature regions are accurately reconstructed.

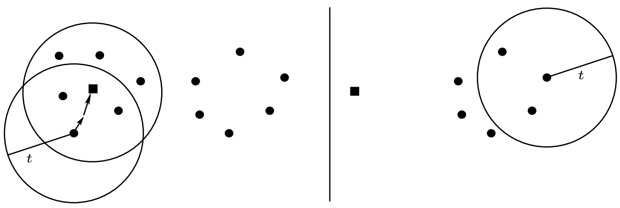


Figure 5: Left: A circular kernel is placed around a randomly chosen point. Using standard Mean Shift the kernel center converges to the centroid of the cluster. Right: All points in the kernel are clustered to the circle center and a new point is selected.

3 Meshing

Once we determined the approximation centers using our method we can render the approximated smooth surface using different visualization techniques like surface splatting or point cloud ray tracing. On the other hand we can use the approximation centers to compute a mesh accurately representing the approximated smooth surface. In this paper, we focus on mesh reconstruction as the mesh representation allows not only for visualization but several other applications like for instance surface modeling and finite element simulations. We use two leading surface reconstruction methods from computational geometry to compute the mesh which are Power Crust of Amenta et al. [3] and Tight Cocone developed by Dey et al. [17].

Although there have been recent advances in improving the robustness of Delaunay-based surface reconstruction methods by Dey et al. [18] and Kolluri et al. [24] most of the algorithms are still sensitive to noise. Figure 6 shows that Tight Cocone and Power Crust fail to reconstruct the dragon head directly from the range scans. If the approximation centers are used as input data for both algorithms the reconstruction result is clearly improved indicating the large reduction of noise that is obtained by our method. The best result is achieved by Tight Cocone as the algorithm uses exactly the approximation centers as vertices of the reconstructed mesh (see bottom image of Figure 6). Therefore, we use this algorithm to calculate a mesh directly from the approximation centers.

4 Mesh Cleaning

In order to improve the mesh reconstructed from the approximation centers using Tight Cocone we apply a simple mesh cleaning. The cleaning robustly removes small disconnected components and single spikes on the surface which can arise due to a large amount of noise in the input data. The cleaning result is illustrated in Figure 7.

First, we eliminate topological noise from the mesh by removing small disconnected components. This is accomplished by a region growing approach. We first choose a seed vertex of the mesh randomly and mark its adjacent triangles. Then we start to grow a region over the surface by selecting triangles adjacent to already marked ones until the whole component is covered. In order to find the next part of the mesh we choose a new seed vertex that has not yet been visited. This process is iterated until all components of the mesh are determined. Finally, we clean the mesh by removing all components that have fewer vertices than 10% of the number of vertices of the

Figure 6: Top: Power Crust (left) and Tight Cocone (right) are not able to reconstruct the dragon head from the range scans. Middle: Power Crust reconstruction of the dragon head using the approximation centers as input data. Bottom: Smooth and detailed Tight Cocone reconstruction using exactly the approximation centers as mesh vertices.

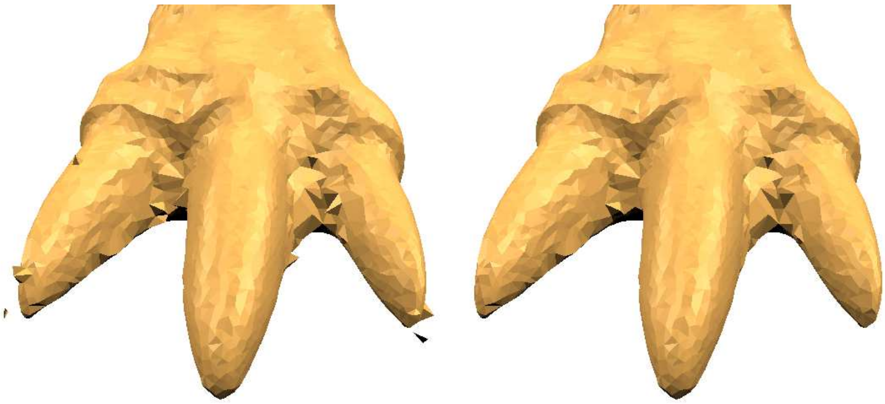


Figure 7: Left: The reconstruction of the left foot of the Armadillo model shows small disconnected components and single spikes on the surface. Right: Spikes and disconnected components are robustly removed after mesh cleaning.

largest component. According to our experiments this criterion works well as the unwanted disconnected components are very small compared to the reconstructed surface.

After the topological cleaning we remove the spikes on the surface using a simple criterion. In order to apply it to a particular vertex $\mathbf{x}_i$ of the mesh we assume that we are given the one-ring neighborhood $\mathcal{N}(i)$ of $\mathbf{x}_i$ consisting of n vertices. We relocate $\mathbf{x}_i$ to the center of gravity of its one-ring if

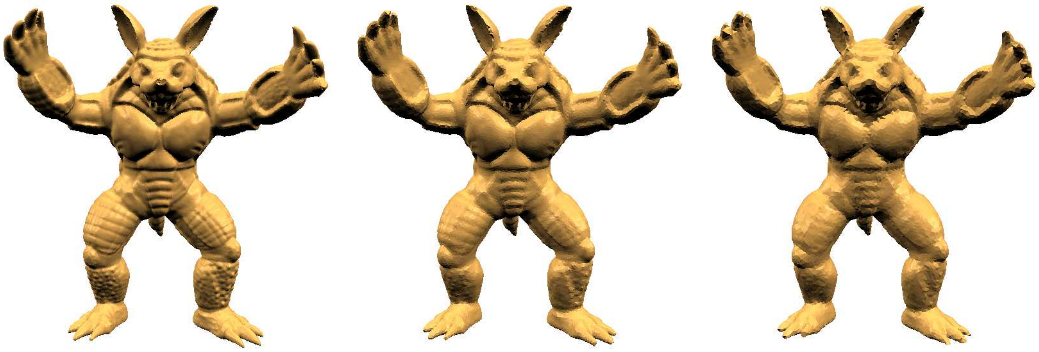
$$\frac{||\mathbf{x}_i - \frac{1}{n}\sum_{\{j\}\in\mathcal{N}(i)} \mathbf{x}_j||}{\frac{1}{n}\sum_{\{j\}\in\mathcal{N}(i)} ||\mathbf{x}_i - \mathbf{x}_j||} > \tau. \tag{10}$$

The formula measures the elevation of the spike using the relation between the distance of $\mathbf{x}_i$ to the center of gravity of $\mathcal{N}(i)$ and the average distance of $\mathbf{x}_i$ to its one-ring vertices $\mathbf{x}_j$. The values of the formula are in the interval $[0, 1)$. If the one-ring neighborhood is flat the quotient is close to zero. When the spike becomes more acute the quotient increases to one. In our implementation we choose the bound $\tau = 0.7$ considering a regular case. If the one-ring vertices form a regular planar n-gon and all edge lengths $||\mathbf{x}_i - \mathbf{x}_j||$ are identical then the value $\tau = 0.7$ corresponds to an angle of approximately 45 degrees between the faces and the base of the spike.

5 Results and Discussion

Robustness
Surface reconstruction methods from computational geometry for meshing scattered data usually require only the specification of a few user-defined parameters and have a strong theoretical background. Therefore, these methods show optimal reconstruction results for various clean datasets. However, in the presence of noise the preconditions of the rigorous mathematical fundament are usually not fulfilled such that these methods fail to reconstruct real-world noisy data as shown in the top images of Figure 6. For the evaluation we choose Power Crust [3] and Tight Cocone [17] as both surface reconstruction algorithms are leading methods and are publicly available. Using our method we compute approximation centers of the input point cloud which are aligned on a smooth surface and are therefore the optimal input data for the computational geometry surface reconstruction algorithms. Figure 6 shows the reconstruction results of Power Crust (middle) and Tight Cocone (bottom) using the approximation centers as input data. The best reconstruction is computed by the Tight Cocone algorithm using exactly the approximation centers as vertices of the reconstructed mesh. The reconstruction is smooth and preserves fine details. Furthermore, the mesh contains no topological noise (see Table 1).

Reconstruction accuracy
Our method computes high-quality sparse meshes from noisy real-world scan data. This is accomplished by computing approximation centers from the scattered data and using them as vertices of a mesh determined by the Tight Cocone algorithm. Figure 9 shows a comparison between the Stanford bunny reconstruction and a reconstruction created using our method having the same number of vertices. The figure illustrates the more accurate reconstruction of our algorithm in high-curvature regions. Furthermore, Table 1 shows the capability of our method to generate meshes with low genus from noisy scattered data. Another interesting feature of our algorithm is the ability to control the reconstruction accuracy by the clustering parameter t. While choosing t adapted to the average sampling density of the noisy scattered data generates an accurate surface reconstruction, increasing the clustering parameter t leads to coarser approximations of the input data (see Figure 8). The Dolphin scan shown in Figure 10 illustrates the robustness of our method against sampling irregularities caused by overlapping range scans. Although the sampling of the point cloud is irregular the reconstruction of the Dolphin model is smooth and accurately preserves small details like the eye and crevices on the fins. Furthermore, our algorithm is easy to



$t = 3.0 \cdot 10^{-4}$ 240K vertices | $t = 1.0 \cdot 10^{-3}$ 29K vertices | $t = 1.5 \cdot 10^{-3}$ 13K vertices

Figure 8: The clustering parameter t controls the reconstruction accuracy by the number of approximation centers used for the Tight Cocone reconstruction. Choosing t adapted to the average sampling density of the noisy input point cloud leads to an accurate surface reconstruction (left). By increasing the clustering parameter the user can obtain coarser reconstructions of the Armadillo model (middle and right).

handle since the kernel size parameter h does not require careful parameter tuning as changes of the kernel size near its optimal value have no significant effect on the reconstruction result (see Figure 11).

Time measurements

The computation times for the results presented in this paper are measured with a standard 2.66 GHz Pentium 4 PC with 1.5 GB RAM. The timings for the different stages of our algorithm are shown in seconds in Table 1. In addition to the computation time of the iterative minimization process we also specify the number of used start points which are selected as a random subset of $\mathcal{P}$. Our experiments with the bunny scans show that the runtime of the iterative minimization process is proportional to the number of start points. This observation is not obvious as we do not perform a fixed number of iterations for every start point but terminate the convergence based on the step size. This allows the user to significantly reduce the runtime of the algorithm by selecting fewer start points if a larger clustering radius is specified to compute a coarse surface approximation of the scattered data.

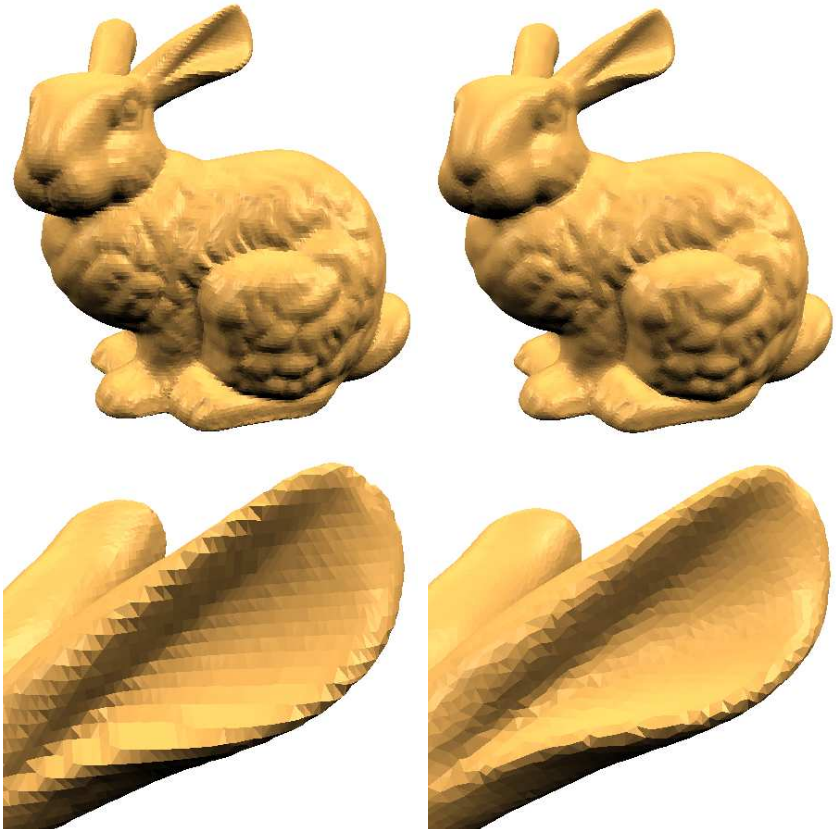


Figure 9: Left: The Stanford reconstruction of the bunny model (top) and a zoom of the left ear (bottom). Right: Reconstruction of the bunny using our method from the original range scans. For comparison both meshes have the same amount of vertices. Our method shows a more accurate reconstruction in high-curvature regions.

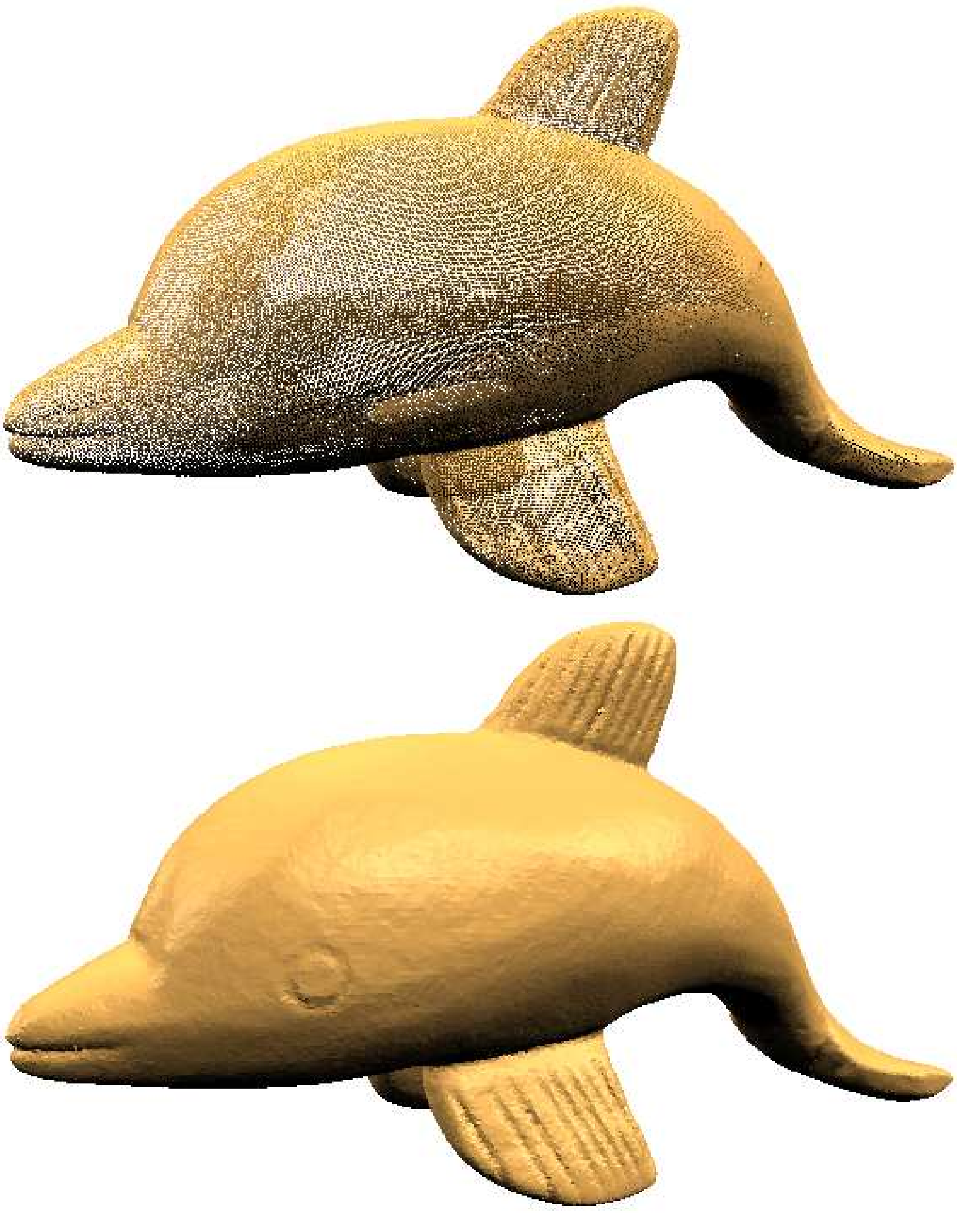
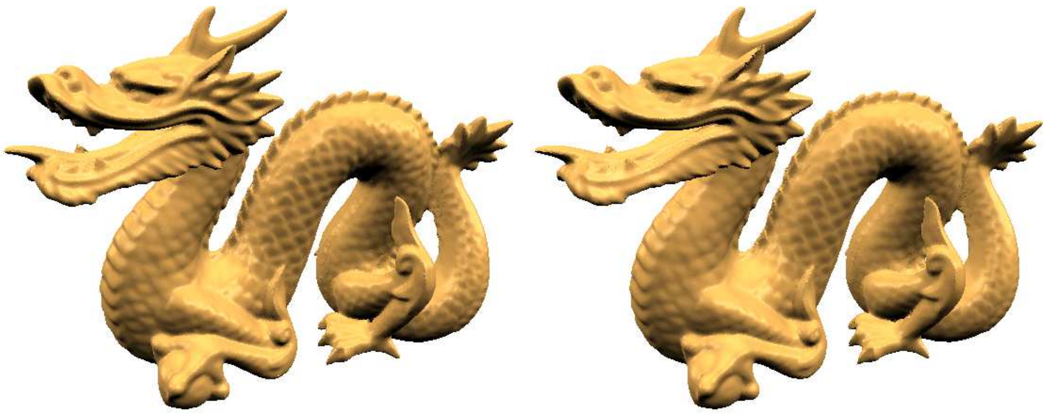


Figure 10: Top: The dolphin point cloud consists of 12 range scans with overlapping regions causing irregular sampling. Bottom: The reconstruction of the Dolphin is smooth and contains fine structures like the eye and ravines on the fins.

Scattered data	# of points	# of approx. centers	ellipsoid kernel comp.	minimization time (# of start points)	mesh cleaning	# of holes	# of handles	kernel size h	clustering radius t
Bunny scans	362K	36K	4.28 sec	76.66 sec (362K)	0.10 sec	0	0		
	362K	34K	4.31 sec	43.84 sec (181K)	0.11 sec	0	0	$1.5 \cdot 10^{-3}$	$8.3 \cdot 10^{-4}$
	362K	29K	4.24 sec	22.05 sec (91K)	0.09 sec	0	0		
Armadillo scans	2.4M	240K			2.20 sec	0	0		$3.0 \cdot 10^{-4}$
	2.4M	29K	183 sec	6183 sec (2.4M)	0.19 sec	0	0	$1.6 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
	2.4M	13K			0.08 sec	1	0		$1.5 \cdot 10^{-3}$
Dragon scans	2.1M	433K	104 sec	2909 sec (2.1M)	4.45 sec	0	6	$1.5 \cdot 10^{-3}$	$2.0 \cdot 10^{-4}$
	2.1M	428K	167 sec	5542 sec (2.1M)	2.13 sec	3	3	$2.0 \cdot 10^{-3}$	$2.3 \cdot 10^{-4}$
Dragon head scans	485K	118K	23.32 sec	709 sec (485K)	0.77 sec	0	0	$1.5 \cdot 10^{-3}$	$2.0 \cdot 10^{-4}$
Dolphin scans	241K	154K	2.29 sec	23.68 sec (241K)	0.72 sec	0	0	$5.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-2}$

Table 1: Computation time measurements, parameter settings and number of computed approximation centers for results presented in this paper. Furthermore, the mesh cleaning times and topological properties of Tight Cocone reconstructions using the approximation centers are shown.



$h = 1.5 \cdot 10^{-3}$ $h = 2.0 \cdot 10^{-3}$

Figure 11: Changes of the kernel size h near the optimal value have no significant effect on the reconstruction result. Therefore, the selection of the kernel size h does not require careful parameter tuning.

6 Future Work

For future work we plan to augment our clustering method to out-of-core handling of very large data sets. As every start point of the gradient-descent procedure is converging independently parallelization of the process is self-evident. Furthermore, we tend to explore direct visualization techniques like surface splatting or point cloud ray tracing to render the surface represented by the approximation centers.

Acknowledgments

We would like to thank Tamal Dey and Nina Amenta for making their surface reconstruction software available. The Armadillo, Bunny and Dragon datasets are courtesy of the Stanford 3D scanning repository. The research of the authors was supported in part by the European FP6 NoE grant 506766 (AIM@SHAPE).

References

[1] U. Adamy, J. Giesen, and M. John. Surface reconstruction using umbrella filters. *Computational Geometry*, 21(1-2):63–86, 2002.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH 1998*, pages 415–421, 1998.

[3] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[4] N. Amenta and Y. J. Kil. Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, August 2004. Proceedings of SIGGRAPH 2004.

[5] M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3):457–465, 2000. Proceedings of EUROGRAPHICS 2000.

[6] C. L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *Proceedings of ACM SIGGRAPH 95*, pages 109–118, August 1995.

[7] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[8] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, October 1984.

[9] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of ACM SIGGRAPH 2001*, pages 67–76, August 2001.

[10] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of ACM GRAPHITE 2003*, pages 119–126, Melbourne, Australia, February 2003.

[11] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

[12] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, August 2004. Proceedings of SIGGRAPH 2004.

[13] D. Cohen-Steiner and F. Da. A greedy Delaunay-based surface reconstruction algorithm. Technical report, INRIA, September 2002. Rapport de recherche 4564.

[14] D. Comaniciu and P. Meer. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[15] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First International Symposium on 3D Data Processing, Visualization, and Transmission*, Padua, Italy, June 2002.

[16] T. K. Dey. *Curve and surface reconstruction*. CRC Press, 2nd edition edition, 2004.

[17] T. K. Dey and S. Goswami. Tight cocone: A water-tight surface reconstructor. In *Proc. 8th ACM Sympos. Solid Modeling Applications*, pages 127–134, 2003.

[18] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. In *Proc. 20th ACM Sympos. Comput. Geom.*, 2004.

[19] H. Edelsbrunner and E. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994.

[20] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40, 1975.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of ACM SIGGRAPH 1992*, pages 71–78, 1992.

[22] I. Ivrissimtzis, W.-K. Jeong, and H.-P. Seidel. Using growing cell structures for surface reconstruction. In *Shape Modeling International 2003*, pages 78–86, 2003.

[23] T. Ju. Robust repair of polygonal models. *ACM Transactions on Graphics*, 23(3), August 2004. Proceedings of SIGGRAPH 2004.

[24] R Kolluri, J. R. Shewchuk, and J. F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, pages 11–21. ACM Press, July 2004.

[25] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, July 2003. Proceedings of SIGGRAPH 2003.

[26] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Shape Modeling International 2004*, Genova, Italy, June 2004.

[27] M. Pauly, N. J. Mitra, and L. J. Guibas. Uncertainty and variability in point cloud surface data. In *Eurographics Symposium on Point-Based Graphics*, pages 77–84, Zurich, Switzerland, June 2004.

[28] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics*, 23(3):878–887, August 2004. Proceedings of SIGGRAPH 2004.

[29] C. Shen, G. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Transactions on Graphics*, 23(3), August 2004. Proceedings of SIGGRAPH 2004.

[30] G. Turk, H. Q. Dinh, J. O'Brien, and G. Yngve. Implicit surfaces that interpolate. In *Shape Modelling International 2001*, pages 62–71, Genova, Italy, May 2001.

[31] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of ACM SIGGRAPH 1994*, pages 311–318, July 1994.

[32] S. Valette and J.-M. Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum*, 23(3):381–390, September 2004. Eurographics 2004 issue.

[33] J. Verdera, V. Caselles, M. Bertalmío, and G. Sapiro. Inpainting surface holes. In *IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain, September 2003.

[34] Y. Yu. Surface reconstruction from unorganized points using self-organizing neural networks. In *Proc. of IEEE Visualization '99*, pages 61–64, 1999.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-001	J. Hoffmann, Carla Gomes	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-001	M. Dimitrios	?
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H.d. Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints
MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension

MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-004	A. Podelski, A. Rybalchenko	Software Model Checking of Liveness Properties via Transition Invariants
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in DL $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-018	G. Schaefer	A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem
MPI-I-2003-1-017	G. Schfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems
MPI-I-2003-1-015	A. Kovcs	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012		- not published -
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Polynomial Time Algorithms for Network Information Flow
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability

MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference