Multiresolution Surface Reconstruction For Hierarchical B-splines

David Forsey and David Wong
(forsey@cs.ubc.ca)
The Department of Computer Science
The University of British Columbia

Abstract

This paper presents a method for automatically generating a hierarchical B-spline surface from an initial set of control points. Given an existing mesh of control points $M^{[k+1]}$, a mesh with half the resolution $M^{[k]}$, is constructed by simultaneously approximating the finer mesh while minimizing a smoothness constraint using weighted least squares. Curvature measures of $M^{[k+1]}$ are used to identify features that need only be represented in the finer mesh. The resulting hierarchical surface accurately and economically reproduces the original mesh, is free from excessive undulations in the intermediate levels and produces a multiresolution representation suitable for animation and interactive modelling.

1 Introduction

Many 3D computer graphics modelling and animation systems use a B-spline representation for curves and surfaces because of their geometric properties such as smoothness and controllable C^n parametric continuity between patches. However, the tensor product nature of the underlying parameterization makes it expensive to construct complicated, continuous surfaces where different regions have differing amounts of local detail.

For example, the surface in Figure 1, as a bicubic B-spline, requires 16524 control points to define its shape simply because the non-local property of knot insertion forces the creation of patches over the entire surface. This also increases the overhead when using splines for surface approximation, rendering, or polygonizing the surface for export to other applications such as video games. It is also the major factor limiting the complexity of spline-based facial models.

Furthermore, as the shapes become more complex, it becomes difficult to make a broad-scale change to the surface shape without distorting or deforming the small-scale details of the surface. This greatly increases the time and cost of manipulating free-form shapes for tasks such as facial animation.

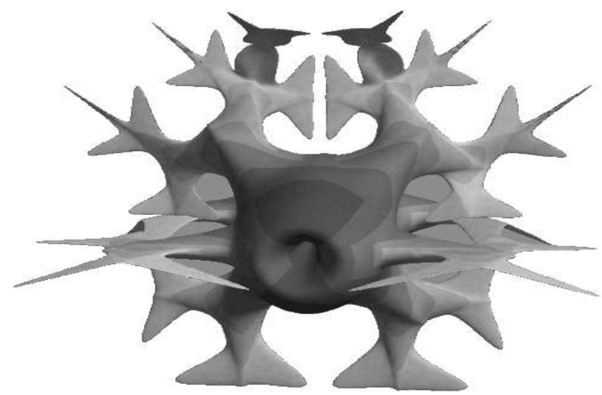

Figure 1 Spiny B-spline surface, 16524 control points

A hierarchical B-spline [1] is a multi-resolution surface representation that provides:

1. Local refinement of tensor-product spline surfaces which localizes detail where it is required.
2. Multi-resolution surface editing that retains surface detail during broad-scale surface manipulation.
3. Economical surface representation. The surface in Figure 1 requires only 178 data points as a hierarchical spline.
4. Multi-resolution animation capabilities that make it easier to animate complicated surfaces [2].

Hierarchical spline surfaces are constructed using a interactive surface modelling system. However numerous applications in medical imaging and computer animation begin with real-world data and it would be extremely useful to construct hierarchical surfaces from such information.

This problem is related to work in multiresolution analysis, particularly curve evolution [3] and wavelets. Wavelets have also been used both for curve and surface representation in an interactive setting [4], and have also been extended to B-splines [5] and subdivisions surfaces [6]. Wavelets have the nice mathematical property that they provide a single unique representation of a function where each level of detail is a least-squares approximation of the next finest level.

In hierarchical surfaces constructed *ab initio*, the coarser level surfaces are typically not "blurry" versions of the final form, but are built up in much the same way as a clay sculpture, starting with a basic shape to which details are added. Thus the hierarchy for a surface with a high-frequency spike would be modelled as flat except at the finest level of detail (Plate 2). A multiresolution representation created by least-squares approximation [7] or wavelet decomposition produces filtered surfaces that can also undulate (Figure 2). This is less useful if these coarser representations are used to interactively control shape.

This paper presents a scheme that, given an initial control mesh for a tensor-product B-spline surface, will construct a multi-resolution hierarchical B-spline that is appropriate for use in modelling and animation. By adding detail only where curvature is high the storage cost of representing that shape either as a spline or when tessellated into polygons can be drastically reduced. We assume that the initial mesh already sufficiently represents the desired surface and we are concerned solely with constructing the multiresolution structure for a hierarchical B-spline.

The remainder of the paper is structured as follows: Section 2 provides the necessary background on hierarchical B-spline surfaces. Section 3 details the method for constructing a coarser approximation for a given mesh. Section 4 describes our particular implementation and numerical methods employed, Section 5 the results of applying the algorithm to a variety of inputs, and conclusions and future work are in Section 6.

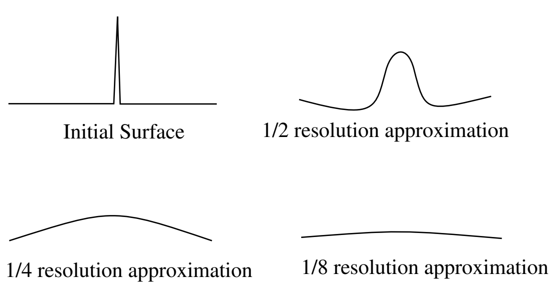

Figure 2 Least squares approximation

2 Hierarchical B-Splines

In 1988 Forsey & Bartels [1] introduced the concept of locally refining a B-spline surface. In this scheme a surface is represented as a series of *levels*, each of which consists of a collection of B-spline surfaces $H^{[k]}$ with twice the resolution of the parent surface at level k-1.

$$H^{[k]}(t, u) = \sum_{i=0}^{m}\sum_{j=0}^{n} V_{i,j}^{[k]} B_{i,\kappa}^{[k]}(t) \; B_{j,\tau}^{[k]}(u) \qquad \text{(EQ 1)}$$

$H^{[k]}(t,u)$ is defined by an $n \times m$ mesh of control vertices $V_{i,j}^{[k]}$, and piecewise polynomial basis functions, $B_{i,\kappa}(t)$ and $B_{(j,\tau)}(t)$, of degree κ and τ respectively. The parameters t and u vary independently from some minimum t_{min}, u_{min} to some maximum t_{max}, u_{max}. Certain values of t and u, called *knots*, correspond to the *joints* between polynomial patches and form a knot sequence in each parameter $\{t_0, \ldots, t_{m+\kappa}\}$, and $\{u_o, \ldots, u_{n+\tau}\}$. The basis functions are assumed to have local support; i.e., each basis function affects only a restricted area of the spline.

To be used in a hierarchical surface representation the bases are required to be *refinable* in the sense that each one can be re-expressed as a linear combination of one or more "smaller" basis functions with additional knots.

Let $M^{[0]}$ be the initial mesh a level 0, defined by control vertices $V^{[k]}$. Points $R^{[k+1]}$ in the refined mesh $M^{[1]}$ are computed by linear combinations of the $V^{[k]}$. Expressed in matrix form:

$$R^{[k+1]} = SV^{[k]} \qquad \text{(EQ 2)}$$

The components of the *subdivision matrix* S depend upon the order and type of basis functions, the number and location of the inserted knots and the topology of the surface. In tensor-product surfaces such as the B-spline, refinement is non-local in that knots are added in the parametric domain resulting in the addition of an entire new row or column of patches across the entire surface.

In a hiearchical B-spline surface representation, each control vertex $V^{[k+1]}$ in $M^{[k+1]}$ is represented in *offset form*:

$$V^{[k+1]} = R^{[k+1]} \oplus \vec{O}^{[k+1]} \qquad \text{(EQ 3)}$$

where $R^{[k+1]}$ is derived by mid-point refinement of $V^{[k]}$ as in Equation 2, $\vec{O}^{[k+1]}$ is the *offset vector* and "$\oplus$" the *offset operator*.

The position of $V_{i,j}^{[k+1]}$ directly after a new level is created is equal to $R_{i,j}^{[k+1]}$ and, by definition, $\vec{O}_{i,j}^{[k+1]} = 0$. Any change to the position of a control node $V_{i,j}^{[k+1]}$, is represented in the offset vector $\vec{O}_{i,j}^{[k+1]}$ as a relative change in position from the reference point $R_{i,j}^{[k+1]}$. Changes to $V_{i,j}^{[k+1]}$ (and thus to $\vec{O}_{i,j}^{[k+1]}$) do not change the position of the reference point. On the other hand, changes in the shape of a parent surface alter $R_{i,j}^{[k+1]}$, resulting in a procedurally defined move of $V_{i,j}^{[k+1]}$.

The reference+offset form allows a hierarchical B-spline surface to be locally refined so that patches can be added to a restricted region of the surface [1], and improves the economy of the representation because the non-zero offsets totally define the shape of the surface.

Offsets correspond to the difference between the shape of the surface at two levels of representation. The offset operator determines how that difference is interpreted. In the simplest case the operator is simply vector addition and is, in operation, similar to wavelet decomposition. However the offset operator can be arbitrarily complex and is used to enhance the behaviour of the surface so that detailed features follow broad scale deformations of the surface [1] or of an underlying skeleton [2].

The hierarchical approach is not restricted to B-splines but is applicable to any representation where the basis functions have local support and are refinable. For example [4] uses a wavelet basis while incorporating the offset form for the wavelet coefficients.

3 Multi-resolution B-spline Approximation

Given an initial control mesh $M^{[k+1]}$ with points $V^{[k+1]}$ defining a B-spline surface, the idea is to generate a "smooth" $M^{[k]}$ mesh for $H^{[k]}$ that, *when refined*, minimizes the magnitude of the offsets in $H^{[k+1]}$. WeNote that we are operating on the control mesh, not on any data points that may have been used to create $M^{[k+1]}$.

3.1 Initial Approximation

Minimizing the offsets implies minimizing

$$\| SV^{[k]} - V^{[k+1]} \|^2 \qquad \text{(EQ 4)}$$

which leads to solving the linear system

$$Ax = b \qquad \text{(EQ 5)}$$

where x is the column vector of the unknown control vertices $V^{[k]}$, A is determined by the subdivision matrix S, and b is the column vector of the $V^{[k+1]}$.

3.2 Smoothness Constraint

To control the solution to Equation 5, we impose a smoothness constraint upon the approximation. Various fairness norms are available and we have chosen to use a simple thin plate model that minimizes

$$\iint \lambda \left(\| V^{[k]}{}_{uu} \|^2 + 2 \| V^{[k]}{}_{uv} \|^2 + \| V^{[k]}{}_{vv} \|^2 \right) du dv \qquad \text{(EQ 6)}$$

Where λ is the regularization parameter that controls the stiffness of the plate. Here the control mesh itself is treated as a grid of sampled points on a control mesh surface.

Discretizing the energy measure for the control mesh yields

$$\sum_{i=0}^{m} \sum_{j=0}^{n} \lambda \left((V^{[k]}_{i,j})^2_{uu} + 2 (V^{[k]}_{i,j})^2_{uv} + (V^{[k]}_{i,j})^2_{vv} \right) \qquad \text{(EQ 7)}$$

Where

λ is the regularization parameter.

$(V^{[k]}_{i,j})^2_{uu}$ is the second order uu-derivative computed by central differencing $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$.

$(V^{[k]}_{i,j})^2_{uv}$ is the mixed second order uv-derivative computed by forward differencing $\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$.

$(V^{[k]}_{i,j})^2_{vv}$ is the second order vv-derivative computed by central differencing $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$.

These components are collected into the smoothness energy matrix P, which is de-coupled into a linear system for each of the x, y, and z coordinates.

Thus to minimize both the offsets and the energy we attempt to minimize

$$\| SV^{[k]} - V^{[k+1]} \|^2 + \lambda \| PV^{[k]} \|^2 \qquad \text{(EQ 8)}$$

which can be written as an over-determined least square problem

$$\begin{pmatrix} S \\ \sqrt{\lambda} P \end{pmatrix} V^{[k]} = \begin{pmatrix} V^{[k+1]} \\ 0 \end{pmatrix} \qquad \text{(EQ 9)}$$

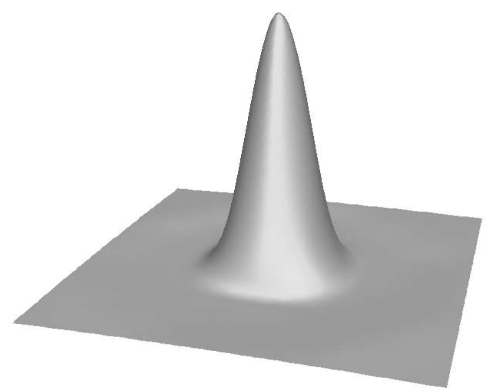

Figure 3 Original Surface. 19x19 control points

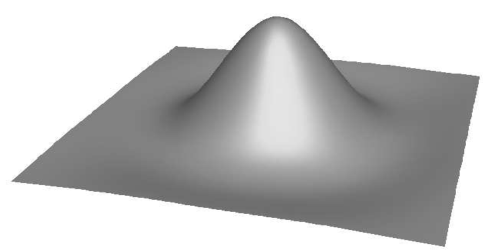

Figure 4 Approximation with $\lambda = 0.1$

3.3 Weighted Approximation

Applied to a mesh that is flat except for a single extreme spike (Figure 3), the method described above produces the surfaces in Figure 4 (with $\lambda = 0.1$). This is a reasonable approximation. It is smooth, and does not exhibit the undulations typical of an unadorned least-squares solution, but it does not capture the notion of a simple surface with added features.

To identify these features, the second order partial deriva-

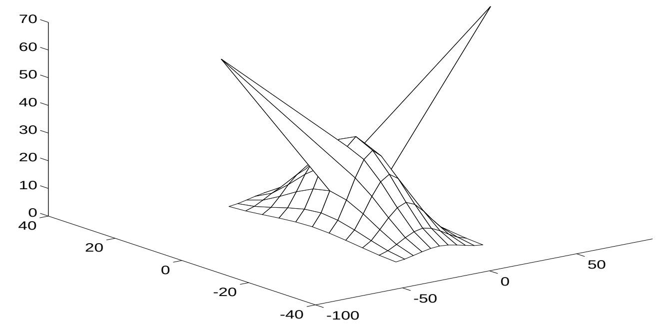

Figure 5 Original 11x11 control mesh

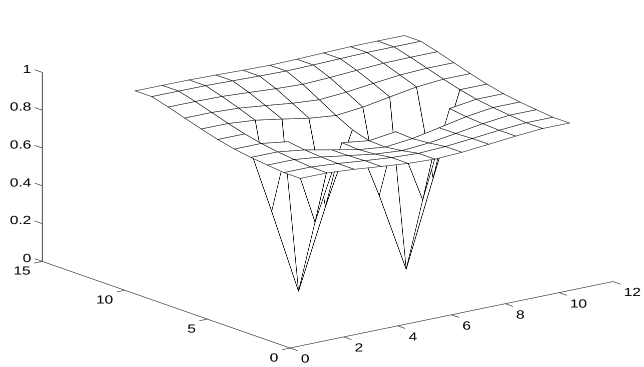

Figure 6 Weights for mesh in Figure 5.

tives uu and vv (which approximate curvature) of each vertex in $V^{[k+1]}$ are calculated (using central differencing). These are normalized and used as a weight w_{ii} for each control vertex as an indication of the importance of that vertex in the approximation. The higher the curvature, the less important that vertex will be in determining the shape of the approximating surface. This alters Equation 9 giving

$$\begin{pmatrix} WS \\ \sqrt{\lambda}P \end{pmatrix} V^{[k]} = \begin{pmatrix} WV^{[k+1]} \\ 0 \end{pmatrix} \qquad \text{(EQ 10)}$$

For example, Figure 5 shows the control mesh for the surface of Plate 3, with the corresponding weights graphed in Figure 6.

Once the unknown $V^{[k+1]}$ are determined, a two-level hierarchical surface is created by using the reference points $R^{[k+1]}$ to determine the offsets required to calculate the $V^{[k]}$ by Equation 3. To reduce the total number of offsets required to represent the surface, offsets that are less than a given tolerance are set to zero. It is this tolerance that determine how close the approximation fits the initial mesh.

4 Implementation

4.1 Basic Algorithm

Section 3 describes how to create a fair approximating control mesh with half the resolution of the initial input mesh. A complete hierarchical structure is constructed by repeatedly applying this procedure until the resolution of the mesh can no longer be reduced (i.e. there would no longer be enough control points to define a valid surface). Once all the meshes have been calculated, the offsets for each level (starting at the coarsest) are determined with thresholding applied to the magnitude of the offsets. The surface is stored by collecting all the non-zero offsets and saving them in a format that can be read by the interactive surface modeler.

4.2 Solving Methods

The heart of the algorithm is to solve the over-determined least squares problem Equation 9 at each coarse level. Because the x, y, and z component of the resulting minimization are de-coupled, the solution can be computed by solving three linear systems. This speeds up and simplifies the computation.

There are numerous sparse matrix techniques available for solving the least square problem [8,9,10,11,12]. We compared several versions of CNE (Corrected Normal Equation) and conjugate gradient (CG) methods as well as several others which there is not space to describe. CNE w/o ordering turned out to be the most efficient and accurate approach to compute the solution.

The conjugate gradient methods turn out to have overhead per iteration in CG in relation to the rate of convergence. Moreover, the normal equation A^TA matrix surprisingly has less non-zero entries than the original A matrix as show in Table 1 below.

Matrix	11x11 mesh	19x19 mesh	35x35 mesh	67x67 mesh
A	1083	3203	10803	39443
A^TA	841	2401	7921	28561

Table 1: Number of non-zeros in matrix

This is due to the tensor product of the B-spline and the second order partial derivative that are within the profile of the B-spline subdivision matrix.

The Cholesky factorization in the CNE method is done only once to yield a triangular system used for each of the three x, y, and z components. In addition, A^TA is banded in the flat topological mesh case, making the CNE very efficient. The instability of the normal equation is reduced by the correction steps in CNE, by the over-determined nature of the problem (about 6 times more rows than columns).

5 Results

Solution of the least squares problem outlined above is O(n^2). Typical executions times are withing 10 seconds for a 67x67 mesh on a MIPS RS4000 processor.

The algorithm was tested on a variety of input meshes. Most were created using an interactive editor for hierarchical B-splines, one created using SoftImage, and the last from a digitized surface data created using a Cyberware laser scanning system. The results are illustrated in Section 8 and summarized in Table 2 and Table 3. All of the resulting surfaces (with exceptions noted below) approximate the initial input mesh with 0.5% of its overall range. The final column in the table lists the number of control points required to represent the surface as a traditional full tensor-product B-spline.

Surface	Original No. of Offsets	No. of Offsets Obtained	No. of Patches	B-spline Equivalent
Plop (Pl.1)	19	19	28	121
Spike (Pl.2)	17	17	34	49
Hump (Pl.3)	19	19	40	143
Eye (Pl.4)	422	253	565	2625
Dog (Pl.5)	257	302	609	1176

Table 2: Example surfaces created using an interactive editor for hierarchical b-splines.

Plates 1 through 3 demonstrate the hierarchical surface produced for some simple geometries. The intermediate levels quite closely resemble those produced with the interactive editing system, have no extra undulations, and add no extra offsets. Note that the non-monochrome images in this section show the distribution of patches of different parametric size. Patches defined at the finest level of refinement are yellow, while patches at coarser levels are reddish.

Surface	Initial No. of Points	No. of Offsets Obtained	No. of Patches	B-spline Equivalent
Reboot (Pl.6)	648	658	600	576
Variable (Pl.7)	576	324	369	576
Spock (Pl.8)	43520	6492	19567	40376

Table 3: Example surfaces from other sources.

The human eye in Plate 4 was one of the first efforts of someone unfamiliar with the interactive editor. The resulting approximation actually reduces the number of offsets used to represent the surface while retaining the final shape. A later effort by the same individual (Plate 5) the algorithm was unable to improve the representation, but is reasonably close.

One goal of the algorithm is to widen the applicability of hierarchical b-splines by creating them from existing data. The head in Plate 6 was created using a commercial modelling and animation package (SoftImage) for **Reboot**, a weekly television series executed entirely using CGI. Because of time constraints in the production schedule, this surface is about at the limit of the allowed complexity. The surface is tedious to manipulate and there are too many patches in places where they are not needed (i.e. the top and back of the head).

The initial mesh of 27x24 control points produced a 3-level hierarchical surface (Plate 6a-c). If a single offset tolerance is used for the entire surface, 658 offsets are generated. By selectively decreasing the tolerance in those regions where detail is not necessary (such as the top and back of the head), the number of offsets can be reduced to 324, and the total number of patches required dropped from 576 to 369 because only the surface in the region of interest is refined (Plate 7). In either case, the hierarchy is suitable for building complicated facial expressions far faster than with the original model (Plate 6d).

Finally, a hierarchical surface was generated from digitized surface information. The original data was 170x256 points representing radii (Plate 8h), evenly distributed around a vertical axis. These were converted to cartesian coordinates in 3 dimensions and used directly as the B-spline control points for the initial mesh. The mesh was approximated using an offset tolerance of 0.5% (the data ranges from 0-500) producing a hierarchical surface (Plate 8a-f) with 5 levels using a total of 6492 offsets. The surface decomposition is show in Table 4.

Level	No. of patches	No. of Offsets
0	40	64
1	160	195
2	640	505
3	2551	1250
4	9178	2650
5	19567	1828

Table 4: Distribution of offsets for the surface in Plate 8

With the surface in hierarchical form, control points at lower levels of detail in the hierarchy are used to make broad scale changes in surface shape. Plate 8i shows the result of moving just 5 control points at level 2 in the hierarchy: three for the eyebrow and two for the corner of the mouth.

6 Conclusions

We have described an efficient method for contructing the levels for a hierarchical B-spline surface from an initial mesh of control points.

On input meshes derived from hierarchical surfaces, the algorithm generates intermediate levels that, in simple cases, is very close to the original and for more complex cases improved on the representation built interactively.

On input meshes derived from external sources, the results are also excellent, producing a representation that is immediately useful for subsequent modelling and animation.

Future work will examine additional methods to improve the identification of a features on a mesh.

7 References

[1] D. R. Forsey and R. H. Bartels. "Hierarchical b-spline refinement." *Computer Graphics*, 22(3):205–212, August 1988.

[2] D. Forsey. "A surface model for skeleton-based character animation." *Proceedings of the the Second Eurographics Workshop on Animation and Simulation*, September 1991.

[3] F. Mokhatarian and A. Mackworth. "A theory of multiscale, curvature-based shape representation for planr curves." *IEEE Transactions of Pattern Analysis and Machine Intelligence*, (14):789–805, 1992.

[4] D. S. Adam Finkelstein. "Multiresolution curves." *Computer Graphics*, July 1994.

[5] T. Lyche and M. Knut. "Spline-wavelets of minimal support." *Preprint 1994-4, Institutt for Informatikk, Universitetet i Oslo, Postboks 1080 Blindern, N-0316*, 1992.

[6] J. W. Tony D. DeRose, Michael Lounsbery. "Multiresolution analysis for surfaces of arbitrary topological type." (93-10-05).

[7] D. Forsey and R. H. Bartels. "Surface fitting and hierarchical splines." *To Appear in Transactions on Graphics*.

[8] P. Matstoms. *Sparse QR Factorization with Applications to Linear Least Squares Problems*. Linköping, 1994.

[9] A. Björck. "Iterative refinement of linear least squares solutions i." In *BIT*, pages 257–278, 1967.

[10] A. Björck. "Least squares methods." Technical report, Department of Mathematics, Linköping University, ?

[11] A. Björck. "SSOR preconditioning methods for sparse least squares problems." In *Computer Science and Statistics 12th Annual Symposium on the Interface*, pages 21–25, 1979.

[12] K. Martin Hanke and L. Per Christian Hansen. *Regularization Methods for Large-Scale Problems*, pages 253–315. Springer-Verlag, 1993.

8 Colour Plates

Plate 1 Plop.
Levels 0-3 (28 Patches, 19 offsets)

Plate 2 Offset Spike.
a) Levels 0-2,
b) Level 3. (28 patches, 17 offsets)

Plate 3 Hump.
Levels 0-3 (40 patches, 19 offsets)

Plate 4 Eye.
Levels 0-6 (565 patches, 253 offsets)

Plate 5 Dog Head.
Levels 0-3 (609 patches, 257 offsets)

Plate 6 Reboot Head. Uniform tolerance.
a)-c) Levels 0-2 (600 patches, 658 offsets),
d) Example expression

Plate 7 Reboot Head Variable Tolerance.
Lighter region indicates regions with larger patches (369 patches, 324 offsets)

Plate 8 Spock Head.
a-f) Levels 0-5,
g) Level 5 rendered to show distribution of coarse patches (orange) among finer patches (yellow),
h) Original Cyberware data, 43520 data points.