Least Squares Subdivision Surfaces
Simon Boyé, Gael Guennebaud, Christophe Schlick

To cite this version:

HAL Id: inria-00524555
https://hal.inria.fr/inria-00524555
Submitted on 23 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Least Squares Subdivision Surfaces

S. Boyé, G. Guennebaud and C. Schlick

INRIA – LaBRI (Bordeaux University - CNRS)

Abstract
The usual approach to design subdivision schemes for curves and surfaces basically consists in combining proper rules for regular configurations, with some specific heuristics to handle extraordinary vertices. In this paper, we introduce an alternative approach, called Least Squares Subdivision Surfaces (LS3), where the key idea is to iteratively project each vertex onto a local approximation of the current polygonal mesh. While the resulting procedure has the same complexity as simpler subdivision schemes, our method offers much higher visual quality, especially in the vicinity of extraordinary vertices. Moreover, we show it can be easily generalized to support boundaries and creases. The fitting procedure allows for a local control of the surface from the normals, making LS3 very well suited for interactive freeform modeling applications. We demonstrate our approach on diadic triangular and quadrangular refinement schemes, though it can be applied to any splitting strategies.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

During the last twenty years, subdivision surfaces have become increasingly popular as a powerful surface representation for interactive modeling applications [ZS00]. A subdivision surface takes as input a relatively coarse polygonal mesh with arbitrary (manifold) topology and produces a smooth surface through an iterative refinement process. This refinement naturally yields a multiresolution structure which is particularly useful to enable adaptive rendering or to efficiently edit 3D objects at different scales [ZSS97, Zor06]. Furthermore, subdivision surfaces offer a relatively high degree of flexibility to artists: arbitrary topology, sharp edges and corners with controllable sharpness [DKT98], etc. For all these reasons, subdivision surfaces are now a well established standard, especially in the game and movie industries.

So far, many different schemes have been proposed [CC78, DS78, Loo87, DLG90, Kob96, PR97, Kob00, CADS09]. In most cases, subdivision rules are designed to reproduce some given (box-) splines. However, such rules can be derived in the regular case only. Consequently, subdivision surfaces are also famous to exhibit severe artifacts around so called extraordinary vertices [SB02] for which no ideal subdivision rule can be derived. In particular, we can distinguish three combined effects: the polar artifact (contraction or dilatation around vertices of low or high valence respectively), poor fairness (ripples), and the lack of curvature continuity.

During the last decade, many work has been focused on improving the behavior of subdivision surfaces around extraordinary vertices. The classical approach consists in tuning extraordinary point rules by analyzing the eigenstructure of the subdivision matrix [DS78, Rei96, ZS00, WW02, BK04]. While most approaches strive to satisfy bounded cur-
nature, it has been shown that such schemes are likely to
decrease the surface fairness [KPR04] and increase the
po
cular artifact [BK04] yielding surfaces of poor visual quality.
Cashman et al. [CADS09] shown higher degree subdivision

Figure 2: Illustration of one LS3 iteration. Starting from the current mesh (a), we first apply the splitting rule to obtain a denser mesh (b). Then the relaxation rule regularizes the mesh (c). The final position of each vertex (d) is obtained by projecting it onto its respective local least-squares approximation patch (shown in green).

procedure developed by Guennebaud et al. [GGG08] which

In this paper we propose a novel approach to design
approximating subdivision surfaces called Least Squares
Subdivision Surfaces (LS3 for short). The key idea behind
LS3 is to explicitly decouple the smoothing rule of classical
subdivision surfaces into a relaxation rule regularizing the
mesh, and a parameterization-free projection operator.
The later offers new possibilities to control the shape and
smootheness of the surface regardless of the other steps of
the subdivision. Central to our framework is the choice of the
projection operator. In this paper, we focus on the use of
local least-square approximations. At each step, each ver-
tex is relaxed by centroid averaging and then it is projected
onto a dynamically fitted surface patch locally approximat-
ing the old neighbor vertices. This is inspired from the Point
Set Surfaces (PSS) literature [ABC+03] where surfaces with
arbitrary smoothness can be defined by mean of local Mov-
ing Least Squares (MLS) approximations [Lev03]. Unlike
our approach, PSS require strong sampling criteria on the
input point clouds as they do not rely on any connectivity
information.

In particular, we adapt the Algebraic Sphere Fitting (ASF)

2. Least Squares Subdivision

Our method follows the same principle than classical subdi-
vision surfaces: an initial polygonal mesh \( M^0 \) with arbitrary
connectivity is iteratively refined by applying a subdivision
operator to the currently refined mesh. This process gener-
ates a sequence of meshes \( M^0, M^1, M^2, \ldots \) with increasing
density which converges to a final limit surface. The subdi-
vision operator is usually a two-step procedure: splitting and
smoothing. In order to gain more flexibility, this last step is
decoupled into a relaxation rule combined with a projection
operator. This lead to the following three-step procedure il-
ustrated in figure 2.

The splitting rule refines the mesh by inserting new ver-
tices and creating new faces (fig. 2-b). To simplify the de-
scription, we only consider primal diadic refinements of tri-
angular meshes, but the process can be easily generalized
to other refinement schemes, such as quadrangular meshes,
dual or \( \sqrt{3} \) subdivisions.

The relaxation rule aims at producing meshes which are
locally uniform by weighted affine combinations of the old
It has been shown that this polar artifact can be easily overcome by relaxing some $C^2$ necessary conditions [BK04]. Such rules are ideal in our case since the final shape of the surface will be given by the projection operator.

More precisely, this relaxation step boils down to classical subdivision. The intermediate position $\mathbf{p}_k$ of each vertex produced by the splitting rules is computed by weighted affine combinations of the old vertex positions $\mathbf{p}_{k-1}$ of the mesh $\mathcal{M}^{k-1}$. The respective masks are shown in figure 3 where the weights $c_0^1$, $c_1^0$, and $c_1^1$ are appropriately set to reduce the polar artifact according to [BK04]-Table 2.

4. Projection operator

The relaxation step alone already produces relatively smooth surfaces most of the time, but with severe smoothness and fairness issues in the vicinity of extraordinary vertices. The role of the projection operator is to alleviate these issues while offering additional controls on the limit surface by mean of the input surface normals.

In a nutshell, our projection operator consists in projecting each vertex of position $\mathbf{q}$ coming from the relaxation rule, onto a simple surface patch that locally approximates the neighbor vertices of the previous mesh $\mathcal{M}^{k-1}$. This yields the final positions $\mathbf{p}_k$ of the new mesh $\mathcal{M}^k$. In the rest of this section, we will motivate our choice for a given local approximation method and discuss the details.

4.1. Local surface approximation

When it comes to local surface approximation, polynomials are an obvious choice for the corresponding basis functions. However, the minimization procedure and the accuracy of the approximation significantly vary according to the parameterization on which the polynomial surface is defined. For instance, a first possibility would be to compute a 2D local parameterization of the neighborhood and fit a bivariate polynomial surface. However, it remains the difficulties of computing such a (local) parameterization in a consistent fashion across the different combinations, and to compute the projection onto it. A simpler possibility would be to use a planar parameterization by first fitting a reference plane and then fitting a bivariate polynomial height field ($\mathbb{R}^2 \rightarrow \mathbb{R}$) as it was done in former PSS definitions [ABC+03]. However, such an approach requires locally relatively flat data such that the plane fit is stable enough and that the neighborhood can be well represented as a height field [AK04b].

More recently, it has been shown that PSS can be advantageously defined by directly working in the natural cartesian parameterization by fitting algebraic (i.e., implicit) surfaces onto the sample points [AK04a, GG07]. More precisely, such approaches fit trivariate $s: \mathbb{R}^3 \rightarrow \mathbb{R}$ implicit polynomials such that the 0-isosurface $S = \{x \in \mathbb{R}^3; s(x) = 0\}$ is as close as possible to the input sample points. In order to avoid the trivial solution $s = 0$, additional regularization constraints have to be added. While it is possible to fit implicit

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Generic relaxation masks where the weights $c_0^1$, $c_1^0$, and $c_1^1$ depend on the valence $n$ of the square vertex and are given in [BK04]-Table 2. $\alpha = (1 - c_0^1)/n$ and $\beta = 1 - c_1^0 - 2c_1^1$. (fig. 2-c, section 3). In practice this step is similar to the smoothing step of classical subdivisions.}
\end{figure}

3. Relaxation rule

The purpose of the relaxation rule is to locally tend towards more uniform meshes that appears to be a necessary condition to generate surfaces of high visual quality. For instance, shading algorithms have a strong ability to depict mesh irregularities and exhibit odd behaviors in the case of extraordinary vertices. The role of the relaxation rule is to locally tend towards a smooth surface approximating the old vertices in a weighted least-square sense. (fig. 2-d, section 4).

Since the splitting rule remains standard, in the next sections we will only focus on the two other steps of the algorithm, and we will show how boundaries and sharp edges can be properly handled in section 5. Section 6 presents numerical tools to analyze the continuity of our surface.
planes [AA04], such a solution is far to be ideal in our case as it would suffer from the same aforementioned limitations of planar parameterizations. More recently, Guennebaud et al. introduced an efficient Algebraic Sphere Fitting (ASF) method [GGG08] to robustly approximate points equipped with normals by spheres. Their method naturally deals with planar regions, and thanks to the use of the normals, it requires only two input points to be well constrained. Therefore, minimal neighborhoods can be used that is especially important to avoid the need to loop through multiple neighborhood rings that is a tedious and expensive task.

In the next sections we briefly review ASF and show how it is adapted to our case.

4.2. Algebraic sphere fitting

An algebraic sphere is defined as the \( \{ \mathbf{x} = \mathbf{n}^T \mathbf{x}, \mathbf{x}^T \mathbf{n} = k \} \) where \( \mathbf{n} \) is a vector of parameters defining the shape of the sphere, and \( d \) is the dimension of the ambient space (in our case \( d = 3 \)). This representation allows the sphere to naturally degenerate to a plane, which is required to approximate planar regions and to robustly move across intersection points. The key idea of the ASF procedure [GGG08] is to first minimize for the constraint \( \nabla s(p) = \mathbf{n} \), stating that the gradient of \( s \) at each input constraint position \( p \) is as close as possible to the surface normal \( \mathbf{n} \) attached to this point. This allows to solve for the \( d+1 \) coefficients \( u_1 \) to \( u_{d+1} \) of \( \mathbf{u} \). The constant coefficient \( u_0 \) is obtained by minimizing the standard distance constraint \( s(p_i) = 0 \). This leads to a very fast fitting procedure with closed form formulas:

\[
\begin{bmatrix}
  u_{d+1} \\
  \vdots \\
  u_0
\end{bmatrix} = \begin{bmatrix}
  1 \\
  \vdots \\
  1
\end{bmatrix} \begin{bmatrix}
  \sum w_j \mathbf{p}_j \mathbf{n}_j - \mathbf{p}^T \sum w_j \mathbf{n}_j \\
  \vdots \\
  \sum w_j \mathbf{p}_j - 2 \sum w_j \mathbf{p}_j \mathbf{p}_j^T \mathbf{p}_j
\end{bmatrix}
\]

where \( w_j \) are the weights associated to the constraint points \( \mathbf{p}_j \) (such that \( \sum w_j = 1 \)), and \( \mathbf{p} \) is the weighted centroid \( \sum w_j \mathbf{p}_j \). We refer to [GGG08] for more details.

The projection of an arbitrary point onto an algebraic sphere is performed by converting it to an explicit sphere or plane if \( |u_{d+1}| > \varepsilon \) or \( u_{d+1} = 0 \) respectively. If \( u_{d+1} \) is close to zero, then a few newton iterations are required for high stability [GGG08].

4.3. Topological weighting

Most sophisticated PSS methods use for the weights \( w_i \) ellipsoidal weight functions allowing to handle some limited anisotropy in the input data [AA06]. In our case, in order to avoid any sampling issue and to make sure the fitted sphere matches our input, these weights should follow the mesh structure as do the relaxation masks. Moreover, let us remark that the weights of the relaxation masks are barycentric coordinates of the point \( \mathbf{q}_i \) for the old neighbor vertices \( \mathbf{p}_j^{k-1} \). Therefore the most natural and obvious choice for the fitting weights \( w_i \) is to take the exact same masks and weights than the ones used at the relaxation steps.

This choice is also motivated by the following remark. To ensure a smooth transition of the fitted spheres, and thus, a high quality surface, such topological weights should tend to meshless (i.e., ellipsoidal) weights between the point \( \mathbf{q}_i \) and the old neighbor vertices \( \mathbf{p}_j^{k-1} \). This goal can only be reached by the combination of the relaxation rule and an appropriate weighting scheme for the fitting of the spheres.

4.4. Normal update

Finally, the normal \( \mathbf{n}_i^k \) of each new vertex \( \mathbf{p}_i^k \) is set to the normal of the locally fitted sphere at the projection point, i.e., \( \mathbf{n}_i^k = \nabla s(p_i^k) \). These normals will be used at the next iteration to fit the spheres. We emphasize that these normal vectors does not necessarily match well the actual surface, and for shading, more accurate normals have to be computed from the mesh itself. On the other hand, using such normal vectors instead of recomputing them from scratch at each iteration has several advantages: it is faster, it allows to better propagate user specified normals through the subdivision levels, and it yields smoother surfaces since our projection step behaves like a smoothing operator of the normal vectors.

5. Boundaries and Sharp edges

As with any subdivision method, boundaries and sharp edges require some special treatments. For borders we propose to simply employ the exact same three step procedure than for the rest of the surface using the standard boundary masks and weights for both the relaxation and projection steps. The relaxation step ensures the necessary tangential smoothing of the curve, while the projection will make sure the curve is consistently “aligned” with the rest of the surface (see figure 4 for an illustration of the the effect of this projection). In the case of primal diadic subdivision, these masks implement cubic spline subdivision and they are recalled in figure 3. Let us also recall that the fitting procedure we adopted works well with only two vertices, so these very small masks are not an issue at all.

Sharp edges are usually handled by treating them as two separated boundaries. This is no longer possible with our
method since we are taking into account the surface normals and both sides are very unlikely to converge to the same curve. As illustrated in figure 5-left, our solution consists in fitting an algebraic sphere on each side as for boundaries, and then project the common relaxed point onto their intersection curve. The sharpness can be easily controlled by repeating a fixed number of sharp iterations followed by smooth iterations as explained in [DKT98], and shown in figure 5-right.

6. Numerical Analysis

Subdivision surfaces are usually analyzed through the eigen-structure of their respective subdivision matrix (e.g., [ZS00]). Unfortunately, owing to the projection step, in our case the new position of a vertex cannot be expressed as a linear combination of the old vertex positions, and therefore the subdivision matrix cannot be formed. This makes the theoretical analysis of our scheme rather difficult.

In the context of subdivision curves, a well established method to analyze non linear schemes is to numerically compute the Hölder regularity of several limit curves and retains the worst case [Kui98, MLD05]. Even though this is limited to curves, it still allows us to get some initial insights on the effect of our projection operator. Using the methodology described in [Kui98] and applying our LS3 approach to planar 2D curves using the weighting masks of cubic spline subdivision (figure 3-bottom), we found that our projection step does not degrade the Hölder regularity which is in both cases greater than two.

The case of surfaces is more difficult, but also more interesting as it contains extraordinary vertices. To this end, we developed a numerical parameterization-free method to analyze the $C^0$, $G^1$, and $G^2$ behaviors of any subdivision surface. In a nutshell, we generate several 2-ring control meshes around a central vertex $p$ of a given valence, subdivide until numerical convergence, and record some rates of convergence for each continuity degree. In order to be able to perform a lot of iterations, at each iteration we safely remove the two most exterior rings without affecting the result. Note that in all cases, the iterations always converged.

$C^0$ behavior is analyzed by recording at each iteration $k$ the contraction factor $\rho_k$ as the maximal length ratio of the edges around $p$ between two successive iterations:

$$\rho_k = \max_j \frac{\|p^k_j - p^k\|}{\|p^{k-1}_j - p^{k-1}\|}$$

where $p^k_j$ are the neighbor positions of the central vertex of position $p^k$ at the level $k$. $\rho_k$ must be strictly lesser than one and, for a diadic scheme, $\rho_k$ should ideally be equal to 0.5.

$G^1$ behavior is equivalent to analyze the convergence rate of the neighborhood of $p$ towards a planar configuration. To this end, we propose to analyze the eigen-structure of the covariance matrix $D^k = \sum_i (p^k_i - p^k)^T (p^k_i - p^k)$. Let $\lambda^k_1$ be the three eigenvalues of $D^k$ sorted such that $\lambda^k_1 < \lambda^k_2 < \lambda^k_3$. Since the magnitude of each eigenvalue varies quadratically with respect to the scale of the input data, we can estimate the contraction factor $\alpha_k$ towards a planar configuration as:

$$\alpha_k = \left( \frac{\lambda^k_1}{\lambda^k_1 + \lambda^k_2} \right)^{\frac{1}{k}}$$

Again, $\alpha_k$ must be strictly lesser than one and, for a diadic scheme, $\alpha_k$ should ideally be equal to 0.5.

Assuming $G^2$ continuity, we can assign to the vertex $p^k$ (resp. its neighbors $p^k_j$) a normal $n^k$ (resp. $n^k_j$) as the eigenvector of the covariance matrix $D^k$ (resp. $D^k_j$) corresponding to the smallest eigenvalue. Then, $G^2$ behavior can be observed by comparing the convergence speed of the neighbor normals $n^k_j$ to the normal $n^k$ of the central central vertex, to the convergence speed of the vertex positions:

$$\beta_k = \frac{\text{arclength}(n^k, n^k_j)}{\text{arclength}(n^{k-1}, n^{k-1}_j)} \frac{\|p^{k-1}_j - p^{k-1}\|}{\|p^k - p^k_j\|}$$

Here $\beta_k > 1$ means the surrounding normals converge slower to the target normal than the vertices do, meaning the surface is not $G^2$. On the other hand, $\beta_k < 1$ means the normals converge too quickly and we are likely generating a flat spot. So the ideal value is $\beta_k = 1$.

We used these numerical metrics to compare our LS3 scheme for triangular meshes to the standard Loop scheme (Std-Loop), and two other variants proposed by Barthe et al. [BK04] exhibiting bounded curvature (Curv-Loop), and no polar artifact (Polar-Loop) respectively. For regular vertices (valence 6) and neglecting the very few first subdivision steps, in all cases we always found the ideal values $\rho_k = 0.5$, $\alpha_k = 0.5$, and $\beta_k = 1$ that is coherent with our 2D analysis of the Hölder regularity. In other words, our projection step seems to not compromise the $C^2$ continuity in the regular case.

Figure 6 shows worse case plots of these three metrics for thousands of random configurations generated around an
extraordinary vertex of valence 12. For comparison purpose, we also included a LS3 variant using the bounded curvature masks [BK04] for both the relaxation and projection steps (noted LS3-Curv). As can be seen, the LS3 variants behave like their respective linear ones, meaning the projection step has a minimal impact on the continuity of the surface. Notice how polar artifacts are well detected by the C0 metric p while the bounded curvature scheme clearly exhibits a better G1 behavior. This is an important observation showing our numerical method matches well the theoretical results.

7. Implementation and Results

Implementation and performance
The complete LS3 algorithm is summarized as pseudocode in figure 7 in the case of primal diadic subdivision of triangular meshes. For this reason, quadrangular meshes can be handled by adding an additional loop through the faces and using Catmull-Clark masks [CC78]. Figure 10 shows some results of Catmull-Clark and \( \sqrt{3} \) subdivision enhanced with our LS3 method. In summary, LS3 subdivision mainly differs to standard subdivision by the additional projection step and the propagation of surface normals. It is therefore straightforward to adapt any existing implementation of Loop or Catmull-Clark subdivision to its respective LS3 variant. Moreover, since the cost of the projection operator is in \( O(n) \) (where \( n \) is the size of the mask), the theoretical complexity is of the same order as classical schemes. In practice, we observed that our implementation as a Mesh Lab [mes] plugin is half as fast as the initial implementation of the Loop scheme.

Initial normals
Our projection operator makes the final surface dependent on the given vertex normals. Vertex normals are usually computed as a weighted average of the adjacent face normals where classical choices for the weights include the face areas, the angles, or even Max’s method [Max99]. While all these weighting strategies are based on very different interpretation of the underlying mesh connectivity, in practice the differences are seldom perceptible for sophisticated enough models. The results of this paper have been obtained using for the weights the face areas. The same method has been employed to compute the normals of the final refined meshes.

Note that Max’s method [Max99] might be of some particular interest since the computed normals for a group of neighboring vertices lying on a sphere exactly match the sphere normals. This allows an exact reproduction of spherical parts.

Figure 8 shows interesting results can be achieved by allowing the user to edit the input normals. In practice this might be useful to fine tune the shape without increasing the mesh resolution. Interactive tools to edit these normals in an intuitive way have yet to be developed.

Fairness comparisons
In section 6 we found that LS3 seems to possesses continuity properties comparable to weighted centroid schemes. However, this does not bring much insights about the fairness of the surface.

Figure 1 shows that LS3 generates surfaces which are comparable to Loop’s scheme everywhere the control mesh is regular but with a significantly improved behavior around extraordinary vertices. This is even more visible in the saddle example of Figure 9 where LS3 is clearly superior to other variants of the Loop scheme and almost indistinguishable from the original quadric. Only a very small artifact is
visible at the exact location of the extraordinary vertex. As expected by Karčiauskas et al. [KPR04], the bounded curvature variant exhibits severe fairness issues. Figure 10 shows that similar gain on the surface fairness is obtained when applying our $LS^3$ approach to Catmull-Clark and $\sqrt{3}$ schemes.

Figure 11 presents another classical test case where weighted centroid subdivisions exhibit much stronger ripples than $LS^3$. This example also clearly shows that $LS^3$ does not possess the convex hull property since the projection step can move vertices outside the convex hull of the mask. On the other hand, surfaces produced by $LS^3$ are usually closer to the control mesh making the interactive edition more intuitive. Another consequence is that details are better preserved. In practice, the more a region of the mesh is close to a sphere (or a plane), the more the limit surface will be close to interpolate the original mesh.

8. Discussions and Conclusion

We presented $LS^3$, a simple generalization of subdivision surfaces that considerably reduces surface fairness issues of weighted centroid schemes. $LS^3$ treats all vertices in a uniform manner and can handle boundaries and sharp edges. We believe that part of the success of our approach is due to an appropriate use vertex normals to locally approximate the surface by spheres and, consequently, smooth out the oscillations and ripples. The little disadvantage is that increases the influence region of a control point to a 3-ring neighborhood instead of only 2 rings for classical schemes. On the other hand, this brings new edition possibilities to the artists. Designing such a normal edition tool is an interesting and challenging area for future work.

Despite the simplicity of our approach, the resulting subdivision scheme is not linear anymore. As the main consequences, we currently lack a rigorous theoretical analysis, it is not possible to directly compute a point onto the limit surface [Sta98], and invariance to general affine transformations is not guaranteed. Note that $LS^3$ is invariant to translations, rotations, and uniform scalings. We also acknowledge that the sphere fitting can lead to a slight unpleasant growing effect when $LS^3$ tries to smooth a sharp and anisotropic region (e.g., as in figure 5-right). For all these reasons, $LS^3$ does not compete as a replacement of NURBS for CAD applications. On the other hand, in practice we found that these limitations are only visible in simple textbook cases, and they become seldom perceptible for soft and/or sophisticated enough models such as, e.g., characters. Therefore, we believe that for many applications, the outstanding ability of $LS^3$ to remove ripples and other subdivision artifacts, combined with its simplicity and efficiency largely compensate these limitations. This is especially true for game and movie industries where subdivision surfaces are already well established, thus enforcing the artists to spend a large amount of time to hide the extraordinary vertices of their models.

Our $LS^3$ approach also opens the door to many future researches, e.g. by investigating novel local approximation methods. For instance, moving from algebraic spheres to...
Figure 11: Comparison of Loop’s method (middle) and LS$^3$ (right) on a 16-sided double cone. Note the high shrinking effect and oscillations of the surface produced with Loop’s scheme.

generic quadric could potentially solve the aforementioned growing effect and bring affine invariance. However, our initial efforts in that direction showed that quadrics introduce more difficulties than solutions and good regularization terms have to be found.

Acknowledgment

The mannequin model of figure 1 is provided courtesy of MPII by the AIM@SHAPE Shape Repository. This work is part of the the SeARCH project, funded by the french National Research Agency ANR-09-CORD-019-01

References


© 2010 The Author(s)