# A Quasi-Minimal Model for Paper-Like Surfaces

Mathieu Perriollat LASMEA – CNRS / UBP Clermont-Ferrand, France Mathieu.Perriollat@gmail.com

#### Abstract

Smoothly bent paper-like surfaces are developable. They are however difficult to minimally parameterize since the number of meaningful parameters is intrinsically dependent on the actual deformation. Previous generative models are either incomplete, i.e. limited to subsets of developable surfaces, or depend on huge parameter sets.

We propose a generative model governed by a quasiminimal set of intuitive parameters, namely rules and angles. More precisely, a flat mesh is bent along guiding rules, while a number of extra rules controls the level of smoothness. The generated surface is guaranteed to be developable. A fully automatic multi-camera threedimensional reconstruction algorithm, including model-based bundleadjustment, demonstrates our model on real images.

#### **1. Introduction**

The behaviour of the real world depends on numerous physical phenomena. This makes general-purpose computer vision a tricky task and motivates the need for prior models of the observed structures, *e.g.* [1, 4, 8, 10]. For instance, a 3D morphable face model makes it possible to recover camera pose from a single face image [1].

This paper focuses on paper-like surfaces. More precisely, we consider paper as an unstretchable surface with everywhere vanishing Gaussian curvature. This holds if smooth deformations only occur. This is mathematically modeled by developable surfaces, a subset of ruled surfaces. Broadly speaking, there are two modeling approaches. The first one is to describe a continuous surface by partial differential equations, parametric or implicit functions. The second one describes a mesh representing the surface with as few parameters as possible. The number of which must thus adapt to the actual surface. We follow the second approach.

One of the properties of paper-like surfaces is inextensibility. This is a nonlinear constraint which is not obvious to Adrien Bartoli LASMEA – CNRS / UBP Clermont-Ferrand, France Adrien.Bartoli@gmail.com

apply to meshes, as figure 1 illustrates. For instance, Salzmann *et al.* [10] use constant length edges to generate training meshes from which a generating basis is learnt using Principal Component Analysis. The nonlinear constraints are re-injected as a penalty in the eventual fitting cost function. The main drawback of this approach is that the model does not guarantee that the generated surface is developable.



Figure 1. Inextensibility and approximation: a one dimensional example. Curve C represents an inextensible object, A and B are two points lying on it. The linear approximation of arc (AB) is the straight segment AB. When C bows, although the arc length (AB) remains constant, the length of segment AB changes. A constant length edge model is thus not a valid parameterization for inextensible surfaces.

We propose a model generating a 3D mesh satisfying the above mentioned properties, namely inextensibility and vanishing Gaussian curvature at any point on the mesh. The model is based on bending a flat surface around rules together with an interpolation process leading to a smooth surface mesh. The number of parameters lies very close to the minimal one because only the global shape is parameterized. A continuous smooth surface is then interpolated. This model is suitable for image fitting applications. We describe an algorithm to recover the deformations and rigid pose of a paper-like object from multiple views. It does not guarantee to find this minimal set, but it estimates a set of few physical parameters explaining the images.

**Previous work.** Developable surfaces are usually chosen as a basic modeling tool. Most of the work uses a continuous representation of the surface [3, 4, 7, 9]. They are thus not well adapted for fast image fitting, except [4] which initializes the model parameters with a discrete system of rules. [11] constructs developable surfaces by partitioning a surface and curving each piece along a generalized cone defined by its apex and a cross-section spline. This param-

eterization is limited to piecewise generalized cones.

[6] simulates bending and creasing of virtual paper by applying external forces on the surface. This model has a lot of parameters since external forces are defined for each vertex of the mesh. A method for undistorting paper is proposed in [8]. The generated surface is not developable due to a relaxation process that does not preserve inextensibility.

**Roadmap.** We present our model in §2 and its reconstruction from multiple images in §3. Experimental results on image sequences are reported in §4. Finally, §5 gives our conclusions and discusses future work.

## 2. A Quasi-Minimal Model

### 2.1. Principle

**Developable surfaces.** Since developable surfaces form a subset of ruled surfaces, they can be defined as constrained ruled surfaces. Their continuous mathematical formulation is given in *e.g.* [11] by:

$$\begin{cases} X(t,v) = \alpha(t) + v\beta(t), & t \in I \quad v \in \mathbb{R} \quad \beta(t) \neq 0 \\ \det(\alpha'(t), \beta(t), \beta'(t)) = 0. \end{cases}$$
(1)

The first equation defines a ruled surface using a differentiable space curve  $\alpha(t)$ , namely the directrix and a vector field  $\beta(t)$ . The ruled surface is actually generated by the line pencil ( $\alpha(t), \beta(t)$ ). The second equation enforces vanishing Gaussian curvature, making the ruled surface a developable one.

Most of the previous work on developable surfaces exhibits functions that satisfy this system. For example [3] uses a tensor product B-Spline representation, [7] uses 'cone spline surfaces' and [9] combines dual representation with NURBS surfaces.

**Rule-based generation.** We propose an intuitive method to build developable surfaces inspired by the observation of real paper sheets. The main idea is to use a discrete set of rules instead of a continuous formulation. This leads to a piecewise planar surface. The constraint on curvature in (1) turns into a formulation in terms of bending angles. The rules are chosen such that they do not intersect each other, which corresponds to the modeling of smooth deformations.

Generating a surface mesh using our model has three main steps, provided the planar boundary shape. First we extract from the parameter set the position of the guiding rules on the flat shape and their bending angle. Second, we add extra rules by interpolating the positions and the angles of the guiding rules. The number of extra rules controls the smoothness of the generated surface. Third, the flat mesh is bent along the rules. Figure 2 illustrates this generating process. It is guaranteed to be admissible in the sense that the surface underlying the generated mesh is developable. Figure 3 shows the generated surface when the number of rules increases.



Figure 2. Surface mesh generation. (left) Flat mesh with guiding rules (thick and pink) and extra rules (thin and green). (right) Mesh folded along the guiding and extra rules.

Balancing model complexity and surface smoothness. It is obvious that the density of rules is related to the smoothness of the surface: the higher the number of rules, the smoother the surface. It is also linked to the model complexity: the higher the number of rules, the more complex the model. These two observations lead us to consider a huge number of rules to generate a smooth and accurate surface. To avoid an overly large number of parameters, we propose to control a subset of the rules and to interpolate the other ones. They are respectively called guiding and extra rules. This has the advantage to generate a smooth surface with a small set of parameters. The aspect of the final surface depends on both the guiding rules and the interpolation process. Figure 3 illustrates the effect of the proportion between the guiding and extra rules. The surface generated by 6 guiding rules and 12 extra rules is an interesting tradeoff: there are enough parameters to capture all deformations since the smoothness given by the extra rules significantly decreases the error, and adding guiding rules does not really improve the accuracy.



Figure 3. Surface generation behaviour. The reference mesh is estimated by our model with a varying number of parameters. The residual error represents the mean distance to the reference mesh, it is given as a percentage of the meshgrid step.

**Internal consistency constraints.** A rule is valid if it does not intersect other rules on the surface and, in the case of a non convex boundary, if the segment joining the two intersections is entirely on the mesh, see figure 4 for an example.





Figure 6. Bent shape with rules. The thick lines are the guiding rules. The thin lines are the extra rules. The dashed red lines are the region limits. The red dots are the region extremities.

SURFACE GENERATION PROCESS

- 1. Define the shape boundary on the flat mesh
- 2. Gather the rules into regions
- 3. Interpolate the rule positions and their angles
- 4. Resample the interpolating functions to get the extra rules
- 5. Fold the flat mesh

Table 2. Overview of the surface generation process.

## 3. A Multiple View Fitting Algorithm

Our goal is to fit the model to multiple images. We assume that a 3D point set and camera pose have been reconstructed from image point features by some means. We use the reprojection error as an optimization criterion. As is usual for dealing with such a nonlinear criterion, we compute a suboptimal initialization that we iteratively refine.

#### 3.1. Initialization

We begin by reconstructing a surface interpolating the given 3D points. A rule detection process is then used to infer our model parameters.

**Step 1: Interpolating surface fitting.** Details about how the 3D points are reconstructed are given in §4. The interpolating surface is represented by a 2D to 1D Thin-Plate Spline function [2], mapping some planar parameterization of the surface to point height. We use the mean plane. Defining a regular grid on this plane thus allows us to infer a dense set of points on the 3D surface. Figure 7 (top right) and figure 8 (top left) show an example.

**Step 2: Model initialization by rule detection.** The model is initialized from the 3D surface. The side length is chosen as the size of the 3D mesh.

Guiding rules must be defined on the surface. This set of n rules must represent the surface as accurately as possible. In [3] an algorithm is proposed to find a rule on a given surface. It tries rules with varying direction and passing through several points on the surface. We use it to detect rules along the sites visible on figure 7 (bottom left).



Figure 7. Rules detection process. (top left) Feature points. (top right) Reconstructed 3D points and the interpolating surface. (bot-tom left) Points where rules are detected on the interpolated surface. (bottom right) Initial detected rules (circles) and automatically detected guiding rules (red squares).

The rules are described by the arc length of their intersection points with the mesh boundary. The two arc lengths defining a rule can be interpreted as a point in  $\mathbb{R}^2$ , as shown in figure 7 (bottom right). The groups of rules in this figure represent the bending regions of the surface. The guiding rules are chosen in the groups. We fix the number of guiding rules by hand, but a model selection approach could be used to determine it automatically from the set of detected rules.

This gives the *n* guiding rules. The bending angle vector  $\theta$  is obtained from the 3D surface by assuming planarity between consecutive rules. The initial suboptimal model we obtain is shown on figure 8 (top right).

### 3.2. Refinement

The reprojection error describes how well the model fits the actual data, namely the image feature points. We thus introduce latent variables representing the position of each point onto the modeled mesh with two parameters. Let L be the number of images and  $N_i$  the number of points in image *i*, the reprojection error is:

$$e = \sum_{i=1}^{L} \sum_{j=1}^{N_i} (m_{j,i} - \Pi(C_j, M(S, x_i, y_i)))^2.$$
(2)

In this equation,  $m_{i,j}$  is the *j*-th feature point in image *i*,  $\Pi(C, M)$  projects the 3D point *M* in the camera *C* and  $M(S, x_i, y_i)$  is a twodimensional parameterization of the points lying on the surface, with *S* the surface parameters. The points on the surface are initialized by computing each  $(x_i, y_i)$  such that their individual reprojection error is minimized, using the initial surface model.

To minimize the reprojection error, the following parameters are tuned: the surface parameters (the number of guiding and extra rules is fixed), see table 1, the pose of the surface (rotation and translation) and the 3D point parameters.

The Levenberg-Marquardt algorithm [5] is used to minimize the reprojection error. Upon convergence, the solution is the Maximum Likelihood Estimate under the assumption of an additive *i.i.d.* Gaussian noise on the image feature points.



Figure 8. Paper fitting with eight guiding rules. (top left) Interpolated surface. (top right) Initial model. (bottom left) Refined model overlaid with the guiding rules. (bottom right) Estimated model.

### 4. Experimental Results

We demonstrate the representational power of our fitting algorithm on several sets of images. For five of them, we show results. Some threedimensional representation of the sequence are represented on figures 9 and 14. The 3D point cloud is generated by triangulating point correspondences between several views. These correspondences are obtained while recovering camera calibration and pose using Structure-from-Motion [5]. Points off the object of interest are removed by hand. Figure 7 (top) shows an example of such a reconstruction.

**The paper dataset.** The following results have been obtained from five views. We used a model with eight guiding rules and sixteen extra rules. Figures 8 and 10 show the reprojection of the 3D surfaces into the first image of the sequence and the reprojection error distribution for the paper sequence for the three main steps of our algorithm: reconstruction with Structure-from-Motion, initialization and refinement. Although the former one has the lowest reprojection error, the associated surface is not satisfying, since it is



Figure 9. (left) Paper sequence. (right) Book sequence.

not regular enough and does not fit the actual boundary. The initialization makes the model more regular, but is not accurate enough to fit the boundary of the paper, so that important reprojection errors are introduced. Eventually, the refined model is visually acceptable and its reprojection error is very close to the unconstrained set of points obtained by Structure-from-Motion. It means that our model accurately fits the image points, while being governed by a much lower number of parameters than the initial set of independent 3D points. The reprojection error significantly decreases thanks to the refinement step, which validates its relevance. Comparing these errors in the object space leads to the same conclusions: the average distance between the triangulated points and the predicted points before (respectively after) the refinement step is 0.16 cm (respectively 0.06 cm), the paper size being estimated to 25 cm by 21 cm. To make the model converge to the actual paper, we manually selected the four corners in one of the five views.



Figure 10. Reprojection errors distribution for the images shown in figure 8. (left) 3D point cloud. (middle) Initial model. (right) Refined model.

Since we have a 3D model of the paper sheet and its reprojection into the images, it is possible to overlay some pictures or to change the texture map. We use the augmentation process described in table 3 to change the whole texture map of the paper and to synthetically generate a view of the paper with the new texture. The results are shown on figure 11.

**The book dataset.** The second dataset is an image pair of a book. We estimate the page surface with two guiding

#### AUGMENTING IMAGES

- 1. Run the proposed algorithm to fit the model to images
- 2. Choose illumination model and light sources
- 3. For each image, automatically do
  - (a) Transfer the new texture map
  - (b) Apply lighting changes

Table 3. Overview of the augmentation process.



Figure 11. (left) Changing the whole texture map of the paper. (right) Synthetically generated view of the paper with new texture.

rules and eight extra rules. Figure 12 shows the reprojection of the estimated surface and the 3D mesh. The reprojection of the computed model is fine: the reprojection error of the 3D points is 0.26 pixels and the one for the refined model is 0.69 pixels, taking the triangulated points as groung truth, the final error in object space is 0.06 cm for a page size of 18 cm by 13 cm. It means that we accurately recover the page shape with a surface governed by only nine parameters.



Figure 12. Reconstruction of a book's page. (left) Reprojection onto the images. (right) Estimated model.

One application of the algorithm in the case of a written page is shown on figure 13: our surface estimate is used to unwarp the page's texture and to get a rectified image of the text.

**The map dataset.** The third example is a sequence of a wavely folded map shown on figure 14. Although all parts of the paper are seen in several images, the whole paper is never entirely seen in a single image. The fitting algorithm naturally deals with this kind of occlusion because the initialization is based on the reconstruction of 3D points, and the 3D points cloud is dense enough since all parts of the paper are visible in several views. The missing points do not



Figure 13. Unwarping. (left) The original page. (right) The rectified page.

perturb convergence because the bundle adjustment minimizes the distance between the actual image points and the reprojection of the 3D points. Since for each images, the set of visible feature points is known, only the corresponding 3D points are projected to compute the residual error. The reprojection of the model onto one of the original images is shown on figure 15. Since the 3D model of the surface and the position of the cameras are known it is possible to compute an occlusion map for each images. This is useful to unwarp the texture map from each image and to combine them to get the whole texture map. Some partial texture maps and the whole one are shown in figure 15. The reprojection error of the model is 0.45 pixels, very close to the error of the initial triangulation (0.31 pixels), in object space the refined model error is 0.07 cm for a sheet size of 28 cm by 19 cm.



Figure 14. Reconstructed paper and cameras for the map dataset.

**The poster dataset.** The former examples deal with small paper sheets where the developable constraints are always satisfied. A poster is a more challenging object because singularities may appear on the surface due to its larger size. The input data are two images of the poster obtained from a calibrated stereo system, see figure 16. The surface of the poster is smooth enough, enabling our model to capture the deformations: the RMS error of the triangulated 3D points is 0.35 pixels and the one for our model is 0.65 pixels.



Figure 15. Results for the map dataset. (top left) Reprojection onto one of the original images. (bottom left) Partial texture maps. (right) Unwarped texture map.



Figure 16. Poster mesh reconstruction. (left) Estimated Model. (right) Reprojection onto the first image.

**The rug dataset.** For this last example, the model is used to estimate a surface whose physical behavior does not satisfy the developable constraints except under special assumption, for example a suspended piece of fabric or in this case an hanged rug. Even though the results are slightly less accurate, the global shape is well-fitted. The difference between the errors of the triangulated points and the model is representative of the lack of accuracy : 0.34 pixels for the original points against 1.36 pixels for the model. This is mainly visible along the boundary of the rug on figure 17.



Figure 17. Rug mesh reconstruction. (left) Reprojection onto the first image. (right) Estimated Model.

## 5. Conclusion and Future Work

This paper describes a quasi-minimal model for paperlike objects and its estimation from multiple images. Although there are few parameters, the generated surface is a good approximation to smoothly deformed paper-like objects. This is demonstrated on real image datasets thanks to a fitting algorithm which initializes the model and refines it in a bundle adjustment manner. Both a surface and its boundary curve are inferred from images.

There are many possibilities for further research. The proposed model could be embedded in a monocular tracking framework or used to generate sample meshes for a surface learning model. The fitting algorithm should be compared to other surface models and estimation methods, in terms of computation and accuracy performances.

## References

- V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. IEEE *Transactions* on Pattern Analysis and Machine Intelligence, 25(9), September 2003.
- [2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
- [3] H.-Y. Chen and H. Pottmann. Approximation by ruled surfaces. *Journal of Computational and Applied Mathematics*, 102:143–156, 1999.
- [4] N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *ECCV'04*.
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geom*etry in Computer Vision. Cambridge University Press, 2003. Second Edition.
- [6] Y. Kergosien, H. Gotoda, and T. Kunii. Bending and creasing virtual paper. IEEE Computer Graphics & Applications, 14(1):40–48, 1994.
- [7] S. Leopoldseder and H. Pottmann. Approximation of developable surfaces with cone spline surfaces. *Computer-Aided Design*, 30:571–582, 1998.
- [8] M. Pilu. Undoing page curl distortion using applicable surfaces. In CVPR'01.
- [9] H. Pottmann and J. Wallner. Approximation algorithms for developable surfaces. *Computer Aided Geometric Design*, 16:539–556, 1999.
- [10] M. Salzmann, S. Ilic, and P. Fua. Physically valid shape parameterization for monocular 3-D deformable surface tracking. In *BMVC'05*.
- [11] M. Sun and E. Fiume. A technique for constructing developable surfaces. In *Proceedings of Graphics Interface*, pages 176–185, May 1996.