Texturing and Deforming Meshes with Casual Images

I-Chao Shen  Yi-Hau Wang  Yu-Mei Chen  Bing-Yu Chen
National Taiwan University

Abstract

Using (casual) images to texture 3D models is a common way to create realistic 3D models, which is a very important task in computer graphics. However, if the shape of the casual image does not look like the target model or the target mapping area, the textured model will become strange since the image will be distorted very much. In this paper, we present a novel texturing and deforming approach for mapping the pattern and shape of a casual image to a 3D model at the same time based on an alternating least-square approach. Through a photogrammetric method, we project the target model onto the source image according to the estimated camera model. Then, the target model is deformed according to the shape of the source image using a surface-based deformation method while minimizing the image distortion simultaneously. The processes are performed iteratively until convergence. Hence, our method can achieve texture mapping, shape deformation, and detail-preserving at once, and can obtain more reasonable texture mapped results than traditional methods.

Categories and Subject Descriptors (according to ACM CCS):
I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Texture mapping is a very common way to create realistic 3D models and is a very important task in computer graphics. Traditionally, the texture images should be designed carefully by artists to achieve high quality texture patterns and less distortion while mapping to a specific 3D model. However, in recent years, due to the wide spread of digital cameras, the number of (casual) images on the Internet is increasing in a very high speed. Hence, people are thinking to map these images to some template models to create various realistic 3D models. This may lead to increase the freedom of texture mapping, but may also introduce serious distortion and alignment problems since such casual images are not designed well for mapping to the template models.

To obtain visually pleasing textured models with casual images, the serious distortion and alignment problems should be solved. Although many previous papers [LPRM02, DMA02, ZMT05] dealt with the distortion problem, those methods all strived to minimize the distortion mathematically for mapping a texture image onto the surface of a given model. However, if the given model does not look like the shape of the texture image, although the distortion could be minimized mathematically, the textured result will be still very strange. This problem may become much more serious when we want to use the casual image(s) to texturing an arbitrary model.

Hence, in this paper, we propose a novel texturing and deforming method for mapping the pattern and shape of a 2D casual image to a 3D mesh model at the same time based on an alternating least-square approach as shown in Figure 1. Through a photogrammetric method, we project the target model onto the source image according to the estimated cameras models, which are computed by using Tzur and Tal’s method [TT09]. Then, the target model is deformed according to the shape of the source image using a surface-based deformation method while minimizing the image distortion simultaneously. The above processes are performed iteratively until convergence. Hence, our method can achieve texture mapping, shape deformation, and detail-preserving at once, and can obtain more reasonable texture mapped results than traditional methods, since not only the source image is warped to map the target model, but the target model is also deformed to fit the shape of the source image.

The rest of this paper is organized as follows. Section 2 describes the most related work of both of texture mapping and shape deformation. We then present our system overview in Section 3 and introduce our main algorithm in Section 4. Section 5 demonstrates some experimental results and provides some discussions. The conclusion and future work are given in Section 6.

2. Related Work

Texture Mapping is a very important task for recent computer graphics applications. There are many approaches to texture 3D meshes with 2D images. Some of them generate low-distortion planar parameterization from a mesh surface by dividing the surface into a collection of small patches called atlas that can be
I-Chao Shen & Yi-Hau Wang & Yu-Mei Chen & Bing-Yu Chen / Texturing and Deforming Meshes with Casual Images

Figure 1: Overview of our method. Given a casual image obtained from the Internet, and a triangle mesh model, we compute the texture mapping and shape deformation interleaving. This example uses a tiger image downloaded from the Internet with low tails, and its head face to the ground, but the model originally has straight tail in the air, and facing to the front. The output is a model not only textured by the casual image, but also fit the shape of the tiger image.

Figure 2: The system flowchart of our system.

The goal of deformation in our system is to manipulate the shape by the reference image while maintaining the surface details of the target model. Our method relies on the linear rotation-invariant (LRI) coordinates [LSLC05]. Inspired by the projection constraint described in Huang et al.'s work [HSL*06], we take the image camera information into the deformation process in order to get the similar projection with the reference image. LRI defined the discrete form at each vertex, which records the relative position depends on its one-ring neighborhood. Based on the discrete form, each vertex has its own coordinate frame called local frame. Given the desired location of the features, the restructuring procedure is solving two least-square systems for the local frames and the real position of each vertex. LRI is one of the implicit way to deform the mesh. Because the local frame encodes the orientation of the vertex, the deformation result is rotation-invariant, and the restructuring process forms as a sparse linear problem.

3. System Overview

In this paper, we present a novel texturing and deforming approach for mapping the pattern and shape of a casual image to a 3D mesh model at the same time. This approach achieves texturing, shape manipulation and detail-preserving at once, and can obtain more visual pleasing and more reasonable texture-mapped results.

Figure 2 shows the system flowchart of our system. Our texturing and deforming approach takes the following three components as the input as several similar previous methods:

1. A casual image: an image comes from the Internet photo database, which is used as the source texture image.
2. A 3D model: a basic template 3D model as the target mesh model.
3. Some corresponding feature points: a set of user specified corresponding points both on the source casual image and the surface of the target 3D model, which will be used for guiding both of the texture mapping and the shape deformation.

recent works about surface-based deformation include multi-resolution mesh editing [ZSS97, KCVS98], Laplacian or gradient surface editing [SCOL*04, YZX*04, LSLC05, BS08], and coupled prisms [BPGK06]. They all tended to preserve the surface details when directly deforming the target model with some constraints.

For the above texture mapping approaches, although they all tried to minimize the image distortion mathematically, due to the non-similarity between the 2D source image and the 3D target mesh model, the distortion may still be too large. Hence, if the target model could be deformed a little bit to fit the shape of the source image, it will make the image distortion as much more small as possible.

Shape deformation is one of the most useful techniques about manipulating and editing the geometry. Recent works about surface-based deformation include multi-resolution mesh editing [ZSS97, KCVS98], Laplacian or gradient surface editing [SCOL*04, YZX*04, LSLC05, BS08], and coupled prisms [BPGK06]. They all tended to preserve the surface details when directly deforming the target model with some constraints.

The goal of deformation in our system is to manipulate the shape by the reference image while maintaining the surface details of the target model. Our method relies on the linear rotation-invariant (LRI) coordinates [LSLC05]. Inspired by the projection constraint described in Huang et al.'s work [HSL*06], we take the image camera information into the deformation process in order to get the similar projection with the reference image. LRI defined the discrete form at each vertex, which records the relative position depends on its one-ring neighborhood. Based on the discrete form, each vertex has its own coordinate frame called local frame. Given the desired location of the features, the restructuring procedure is solving two least-square systems for the local frames and the real position of each vertex. LRI is one of the implicit way to deform the mesh. Because the local frame encodes the orientation of the vertex, the deformation result is rotation-invariant, and the restructuring process forms as a sparse linear problem.

3. System Overview

In this paper, we present a novel texturing and deforming approach for mapping the pattern and shape of a casual image to a 3D mesh model at the same time. This approach achieves texturing, shape manipulation and detail-preserving at once, and can obtain more visual pleasing and more reasonable texture-mapped results.

Figure 2 shows the system flowchart of our system. Our texturing and deforming approach takes the following three components as the input as several similar previous methods:

1. A casual image: an image comes from the Internet photo database, which is used as the source texture image.
2. A 3D model: a basic template 3D model as the target mesh model.
3. Some corresponding feature points: a set of user specified corresponding points both on the source casual image and the surface of the target 3D model, which will be used for guiding both of the texture mapping and the shape deformation.
Then, through an alternating least-square approach that solves the shape deformation and texture mapping algorithms iteratively and alternatively while minimizing the distortion of the casual image and preserving the surface details of the template 3D model simultaneously.

Inside the system, there are three major stages of our approach as the follows:

1. **Camera Model Estimation**: to estimate a coarse global camera projection model by using Tzur and Tal’s approach [TT09].
2. **Shape Deformation**: using the estimated global camera projection model to form the constraints to deform the target 3D model.
3. **Texture Mapping**: given the updated geometry, each vertex will be textured with the casual image and a new local camera projection model will also be computed.

The above three major processes are performed iteratively and alternatively until convergence. The result of our texturing and deforming approach is not only a reasonable texture mapped 3D model, but also can obtain the fitted shape according to the casual image.

### 4. Texturing and Deforming Meshes

In this section, we elaborate on the details of the entire alternating least-square approach. In the beginning, we first introduce how we use the same set of constraints (corresponding feature points) for both of the texture mapping and shape deformation. Then, the key idea of our approach and the objective function used for optimization are explained. At last, the alternating minimization strategy to solve the optimization problem is described in details.

Given a casual image \( I \) and a 3D template mesh model \( M \), a set of corresponding feature points \( \{ (p_i \in I)_{i=1}^P, \{v_i \in M\}_{i=1}^P \} \) is provided by the users, where \( P \) is the number of the corresponding features points. The corresponding feature point set will not only be used for mapping the image onto the mesh, but also for deforming the mesh to fit the image.

#### 4.1. Objective Function

The objective function of our texturing and deforming approach is defined as:

\[
E = (1 - \alpha) \sum_{i=1}^{N} E_t(v_i) + \alpha \sum_{i=1}^{N} \sum_{j=1}^{P} E_p(v_i, v_j),
\]

(1)

where \( 0 \leq \alpha \leq 1 \) is a weight for controlling the two terms, \( N \) is the number of vertices of the mesh model \( M \). The first term in the objective function is used for preserving the surface details of the mesh model \( M \), while the second term is used for mapping the texture image \( I \) on to the surface of the mesh model \( M \).

We will explain each term of this objective function in the following subsections. Note that adjusting \( \alpha \) will influence the importance between the texture mapping and shape deformation phases. The procedure which comes to the threshold of convergence depends on the weighted error function. The system will only be operated for texture mapping if we set \( \alpha = 1 \).

#### 4.2. Texture Mapping Phase

The second term of the objective function Eq. (1) is used for texture mapping, which is defined as:

\[
E_p(v_i, v_j) = \frac{1}{\epsilon + D(v_i, v_j)^\beta} \| M v_j + c_i - p_j \|^2,
\]

(2)

where \( D(v_i, v_j) \) is the geodesic distance between the two vertices \( v_i \) and \( v_j \) on the mesh \( M \), \( v_j \) and \( p_j \) are the corresponding feature points on the mesh \( M \) and the image \( I \), respectively, and \( \beta \) and \( \epsilon \) are small constants as used in [TT09]. \( M_i \in \mathbb{R}^{2 \times 3} \) and \( c_i \in \mathbb{R} \) are the world-to-image projection matrix and the translation scalar in the image plane, respectively, which are the unknowns in this term.

Hence, by giving the corresponding points \( v_j \) and \( p_j \), we tend to calculate the projection matrix \( M_i \) and the translation scalar \( c_i \) for each vertex \( v_i \) on the mesh \( M \), so that we can map the image \( I \) onto the surface of the mesh model \( M \) through the projection matrix \( M_i \) and the translation scalar \( c_i \).

#### 4.3. Shape Deformation Phase

The texture mapping methods proposed before only considered the original geometry, even though the object shape in the casual image is quite different from the target 3D model. If the casual image does not look like the target model, the textured model will become strange and the image will be distorted very much. Hence, we seek to take the shape difference between the geometry and the image into consideration to achieve much more reasonable textured result. Therefore, we introduce a shape deformation term into the objective function Eq. (1), which is the first term in the equation and is defined as:

\[
E_t(v_i) = \| L(v_i) - L(v'_i) \|^2,
\]

(3)

where \( v'_i \) is the updated vertex of the original vertex \( v_i \) after the shape deformation, and \( L(v_i) \) is the Laplacian coordinate of it as used in similar approaches, like [LSLCO05]. Hence, this term seeks to minimize the error caused by deforming the mesh model and could preserve the surface details.

#### 4.4. Global Camera Estimation

For projecting the target model onto the source image, we estimate a coarse global camera model by using the following equation:

\[
\text{argmin}_{v_i} = \sum_{j=1}^{P} D(v_i, v_j),
\]

(4)

where \( D(v_i, v_j) \) denotes the geodesic distance between a vertex \( v_i \) on the mesh \( M \) and a feature point \( v_j \) also on the mesh \( M \). We then find the camera projection of the vertex with the minimum value as our estimated global camera.

#### 4.5. Alternating Least-Square Optimization

Simultaneously optimizing all the unknowns in Eq. (1) gives rise to an expensive nonlinear least-square problem. However, an initial inaccurate solution to this nonlinear problem can be refined until
convergence by iteratively alternating two simpler and more efficient least-square steps, which are respectively responsible for improving the estimation of the vertex positions and the local camera projection models.

The first step of minimizing the objective function is to estimate a global camera model that project the model onto the image. Then, using the estimated global camera model to calculate the vertices’ positions \( v' = \{ v'_1, ..., v'_N \} \) after the shape deformation. The final step does the opposite, that fixes the vertices’ positions and optimize the projection matrix \( M_i \) and the translation scalar \( c_i \) instead. The above steps are operated iteratively and alternatively until convergence.

5. Experiential Results

5.1. Implementation

Our method was implemented in C++ with CGAL library and tested on a desktop PC with an Intel Core 2 Q9400 Quad 2.66GHz CPU with 4GB RAM. The core computation part is solving the sparse linear system, and the TAUCS solver is used. In order to let user to specify the corresponding features between the 2D image and the 3D model, we developed an interactive system as shown in Figure 3, which is similar with other texture mapping systems. This feature-specification user-interface is designed by using QT and QGLviewer library.

| Model  | Vertex# | Feature# | Iteration# | Time/sec. |
|--------|---------|----------|------------|-----------|
| lion   | 5,000   | 86       | 12         | 33.77     |
| halfman| 3,243   | 81       | 4          | 15.79     |
| bottle | 1,441   | 40       | 3          | 12.52     |
| woman  | 4,865   | 70       | 4          | 15.24     |

Table 1: Test data and performance statistics. From left to right: Number of vertices, number of features, number of iterations to converge, and the computation time of each iteration.

5.2. Result

In our experiment, the system takes five iterations in average to get the error converged. The execution time depends on each case as shown in Table 1. To compute the geodesic distances between the constraints and each vertex takes half of the execution time, and it is directly proportional to the number of the constraints (feature points). Figure 4 shows a comparison with our result and the traditional texture mapping result. The pure texture mapping result is not good enough for human visual sense, but we can improve it by bending the elbow. Figure 6 shows a simple example by using a simple cylinder bottle to fit to a cup image. Since our method can deform the bottle a little bit to fit the cup shape, the textured and deformed result is better than traditional texturing one.

5.3. Limitation

When the image object in the casual image has a quite different shape with the target triangle mesh model, due to the instinct of the surface-based deformation algorithm, the deformation of the result model still may not be as good as expected. At the same time, enforcing the deformation may require the user to specify many feature points.

6. Conclusion

In this paper, we present a novel framework not only for texturing the casual images onto the 3D models, but also transferring the shape of the image object to deform the models. Our texturing and deforming method is based on an alternating least-square approach to optimize the texture mapping and shape deformation problems iteratively. Since it takes account for the shape of image object additionally, it achieves texture mapping, shape deformation, and detail-preserving at the same time, and can obtain more reasonable texture-mapped results than traditional methods.

In the meanwhile, this framework is naturally composed of two separate algorithms, which are photogrammetric texture mapping and surface-based deformation. Due to this instinct, it provides great flexibility that everyone can improve the result and accelerate the algorithms in the near future by replacing the texture mapping algorithm or the surface-based deformation one. In the future, we would like to solve the geodesic distance using GPU, because it is the most time-consuming part in our current system.

References

[BPGK06] BOTSC H. M., PAUL Y. M., GROSS M., KOBBELT L.: PriMo: coupled prisms for intuitive surface modeling. In Proceedings of the 2006 Eurographics Symposium on Geometry Processing (2006), pp. 11–20. 2

[BS08] BOTSC H. M., SORKINE O.: On linear variational surface deformation methods. IEEE Transactions on Visualization and Computer Graphics 14, 1 (2008), 213–230. 2

[DMA02] DESSRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. Computer Graphics Forum 21, 3 (2002), 209–218. (Eurographics 2002 Conference Proceedings). 1, 2

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. ACM Transactions on Graphics 25, 3 (2006), 1126–1134. (SIGGRAPH 2006 Conference Proceedings). 2

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In ACM SIGGRAPH 1998 Conference Proceedings (1998), pp. 105–114. 2

Figure 3: A screen shot of the user interface.
Figure 4: Texturing the half man model using a superman image downloaded from the Internet. (a) The input triangle mesh model. (b) The input image. (c) The texturing (and deforming) result using our algorithm with 86 constraints (feature points). (d) The texturing result of [TT09]. In this example, our method deformed the elbow on the target model respect to the bend elbow in the source image, and leads to more reasonable textured result at the cloth wrinkle.

Figure 5: (a) The input cup image. (b) The input simple cylinder bottle model. (c) The textured (and deforming) result.

[KSG03] Kraevoy V., Sheffer A., Gotsman C.: Matchmaker: constructing constrained texture maps. ACM Transactions on Graphics 22, 3 (2003), 326–333. (SIGGRAPH 2003 Conference Proceedings).

[Lő1] Lévy B.: Constrained texture mapping for polygonal meshes. In ACM SIGGRAPH 2001 Conference Proceedings (2001), pp. 417–424.

[LPRM02] Lévy B., Petittjean S., Ray N., Maillot J.: Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics 21, 3 (2002), 362–371. (SIGGRAPH 2002 Conference Proceedings).

[LSLCO05] Lipman Y., Sorkine O., Levin D., Cohen-Or D.: Linear rotation-invariant coordinates for meshes. ACM Transactions on Graphics 24, 3 (2005), 479–487. (SIGGRAPH 2005 Conference Proceedings).

[LYY08] Lee T.-Y., Yen S.-W., Yeh I.-C.: Texture mapping with hard constraints using warping scheme. IEEE Transactions on Visualization and Computer Graphics 14, 2 (2008), 382–395.

[SCOL’04] Sorkine O., Cohen-Or D., Lipman Y., Alexa M., Rössl C., Seidel H.-P.: Laplacian surface editing. In Proceedings of the 2004 Eurographics Symposium on Geometry Processing (2004), pp. 175–184.

[TBTS08] Tai Y.-W., Brown M. S., Tang C.-K., Shum H.-Y.: Texture amendment: reducing texture distortion in constrained parameterization. ACM Transactions on Graphics 27, 5 (2008), Article No.: 136. (SIGGRAPH Asia 2008 Conference Proceedings).

[TT09] Tzur Y., Tal A.: FlexiStickers: photogrammetric texture mapping using casual images. ACM Transactions on Graphics 28, 3 (2009), Article No.: 45. (SIGGRAPH 2009 Conference Proceedings).

[YZX*04] Yu Y., Zhou K., Xu D., Shi X., Bao H., Guo B., Shum H.-Y.: Mesh editing with poisson-based gradient field manipulation. ACM Transactions on Graphics 23, 3 (2004), 644–651. (SIGGRAPH 2004 Conference Proceedings).

[ZMT05] Zhang E., Mischakow K., Turk G.: Feature-based surface parameterization and texture mapping. ACM Transactions on Graphics 24, 1 (2005), 1–27.

[ZSS97] Zorin D., Schröder P., Sweldens W.: Interactive multiresolution mesh editing. In ACM SIGGRAPH 1997 Conference Proceedings (1997), pp. 259–268.
Figure 6: (a) The input woman image. (b) The input simple woman model. (c) The textured (and deformed) result.

[ZWT*05] ZHOU K., WANG X., TONG Y., DESBRUN M., GUO B., SHUM H.-Y.: TextureMontage. ACM Transactions on Graphics 24, 3 (2005), 1148–1155. (SIGGRAPH 2005 Conference Proceedings).