# Coherent Bump Map Recovery from a Single Texture Image

J.M. Dischler [(a)] K. Maritaud [(b)] D. Ghazanfarpour [(b)]

(a) LSIIT (UMR 7005 CNRS ULP), Pôle API, Bvd S. Brant, 67400 Illkirch Cedex
(b) Laboratoire MSI, ENSIL, 16 rue Atlantis, 87068 Limoges Cedex

*Abstract*

In order to texture surfaces realistically with texture images (e.g. photos), it is important to consider the underlying relief. Here, a method is proposed to recover a coherent bump map from a single texture image. Different visual zones are first identified using segmentation and classification. Then, by linearly separating the relief into a noise-like small-scale component and a smooth "shape-related" large-scale component, we can automatically deduce the bump map as well as an "unshaded" color map of the texture. The major advantage of our approach, compared to sophisticated measurement techniques based on *multiple* photos or specific devices, is its practical simplicity and broad accessibility, while it allows us to obtain very easily, via basic bump mapping or displacement mapping, rendering results of good quality.

*Key words: texturing, shape from shading, texture segmentation, bump mapping.*

## 1 Introduction

Texturing synthetic 3D surfaces using a single texture image obtained from a real-world surface (e.g. using a photograph) is clearly desirable, because of its simplicity. However, it raises a number of difficult practical problems. One of them is related to the fact that most textures are not limited to color variations. Bump mapping [1], 3D and geometric textures [7], bi-directional texture functions [3, 4, 10], polynomial textures [11] and relief texture mapping [12] outline this well. With conventional texture mapping [2] or with straight texture analysis and synthesis [19, 20, 21], only the color information is considered and reproduced on surfaces. Therefore, rendered surfaces still look smooth and flat, instead of looking rough and bumpy. In order to obtain more realistic rendering results, it is important to also consider the underlying geometry.

Many approaches [3, 10, 11, 16, 17, 18] have been investigated in recent years for acquiring the relief and reflectance from real-world surfaces. These approaches all consist in performing a sort of "real" data measurement by using specific devices and/or multiple photos taken under precise or arbitrary lighting conditions and viewpoints. But, multiple photos with different viewpoints and different lighting conditions are not always easily available. Practical manipulations can be complicated, such as for more or less large, outdoor, natural surfaces (ocean waves, sand dunes, tree bark, etc.). The same applies for specific devices and tools, which are not always available, affordable and accessible to users. In addition, measurement devices such as laser scanners do generally not span all orders of geometric scales, e.g. from microscopic relief (a few microns) to macroscopic relief (several miles), which further limits the domain of application.

In parallel to these practical limitations, users might want to take profit of the numerous texture databases on commercial CD-ROMs, free web pages, photo-albums, etc. providing thousands of individual texture images usually shot under completely arbitrary and unknown lighting conditions. This enormous and easily available texture diversity seems, in our opinion, worth to be "better" exploited by computer graphics by adding – at least – the missing bump maps for relief sensation recovery.

In computer vision, a large number of *shape from shading* (SFS) techniques have been investigated; the goal of these techniques being to recover a bump map from a single view under some precise conditions. The works of Horn and Brooks [9] as well as Zhang et al. [22] give a relatively complete overview of the domain. Unfortunately, these methods still fail with "complex" natural textures [22]. Shadows, specular highlights, hue, different roughness scales, and so forth, still strongly disturb their effectiveness. The inherent mathematical difficulty of extracting relief (and eventually reflectance), from a single image – not even necessarily in an accurate way, but at least in a visually consistent way –

certainly explains the absence, until now, of efficient methods in computer graphics applications.

This paper proposes an approach based on image segmentation and filtering to resolve, under some not too constraining conditions of lighting and types of texture, the problem of bump map recovery from a single texture image. Figure 1 summarizes the principles of our approach. We use segmentation to divide the texture into basic visual components (sets of pixels). On one hand, we identify different color zones, including shadows and specular highlights, using color quantization; and, on the other hand, we identify "texture elements" with a specific signification such as spots, bricks, grains, etc. by grouping some color zones. Then, the classified structures are analyzed in order to recover the desired underlying components: relief and "unshaded" color. Therefore, we simply adapt the actually very general SFS problem to the particular case of "bump-like" textures. To simplify the problem, we linearly separate the relief into two distinct scales (large-scale and small-scale), which actually turns out to be valid for many natural textures. In the case of a brick wall for example, the large-scale relief corresponds to the bricks shapes (how they come out of the wall), while the small-scale relief corresponds to the rough nature of the bricks (small, but yet visible, noise-like asperities). The small-scale relief is addressed using a filter computed according to the light source direction. The large-scale relief is addressed by using the identified texture structures. Our method is mostly automatic, since the user only needs to quickly provide some minimal "external" information, such as structures and apparent light source direction.

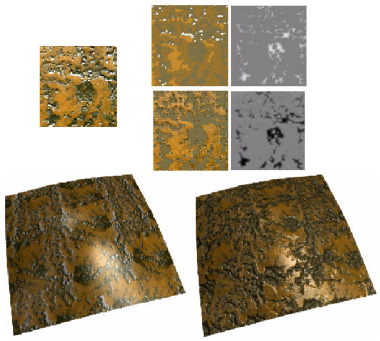

*Figure 1: Method overview. Top left: the user provides a single texture image (here, rusty metal) with unknown relief. Top right: Segmentation (upper-left) into color zones to identify color components; and (upper-right) into "semantic" zones to identify the "structures" of the texture. Just below: extraction of color (left) and relief (right) by using the previous segmentation and classification. Bottom left: lapped texture mapping using the original image. Bottom right: result from our method, using the extracted color and relief.*

The remaining parts of this paper are organized as follows: section 2 states the mathematical problem and the assumptions that our method makes. Sections 3 and 4 respectively concern small-scale and large-scale relief recovery. Section 5 explains how to combine the elements of the two previous sections. Graphical results are shown in section 6. Finally, we conclude the paper and discuss some future directions.

## 2 Texture and lighting conditions

In order to simplify the difficult mathematical problem of relief recovery from a single image, we make a certain number of assumptions concerning the texture, its image and the lighting condition. However, we wish to avoid too drastic constraints as generally demanded by SFS [9] to be able to process a wide range of photos of diverse real-world textures.

The most basic assumptions firstly concern the image and the lighting condition. We suppose that the lighting is uniform over the entire image; so, the natural surface should be locally a plane. One white light source is assumed dominant and sufficiently far away to retrieve a constant lighting direction. Inter-reflections and secondary light sources will be approximated by using a constant "ambient" term (scalar value). Hence, a unique vector $L_i$, plus a single scalar coefficient, can characterize the entire lighting condition. The observer is assumed perpendicular to the image plane, located at a certain distance on the line passing approximately through the center of the image. This line will correspond to the z-axis of the recovered height map (bump map) of the texture. These basic assumptions are common to almost all the usual SFS techniques.

The next assumptions concern the nature of the texture itself. We assume that the relief can be entirely characterized by a height map. This means that fur, wicker, cloth, and so forth are excluded. The reflectance is approximated by a "Phong-like" function; i.e. a diffuse Lambert part, plus a specular part, which we will disregard for relief recovery. We further assume that there are no visible environment reflections on the texture. This excludes highly specular surfaces like mirrors. However, we allow localized specular highlights. Finally, we assume that the texture is composed of individual arbitrarily shaped bumps, which may have different tints, such as bricks in brick walls, "scales" in bark, grooves in stones, etc. Unlike most SFS techniques, we will consider a number of important effects: shadows, specular highlights, different hues and more or less important roughness characteristics. In practice, we distinguish two roughness scales: a small-scale relief that has a relatively small amplitude, which corresponds to noise-like irregularities. Because of this small amplitude, we can assume, without restricting the effi-

ciency of our method, that there are no (or negligible) self-shadowing effects related to this relief (the amplitude may even be null for smooth surfaces). The large-scale relief has a relatively large amplitude, therefore it may also produce self-shadows as well as local specular highlights. We will assume that this type of relief can be characterized by a single elevation curve (1D). Though this seems to be a drastic simplification, many natural textures do verify this condition, including the examples shown in this paper. The use of a single elevation curve has some similarities with the generalized cylinders approach used by Dischler and Ghazanfarpour [6] for reconstructing 3D shapes from two views. Our approach, however, is different, since we do not consider "full" 3D shapes.

Now, let $T(i,j)$, $(i,j)\in[1,n]^2$, be the texture image. With the previous assumptions, we can rewrite our relief recovery problem as the following equation:

$$T(i,j) = K_d C(i,j) L_i N(i,j) + K_s \left(L_i V(i,j)\right)^p + K_a C(i,j)$$

where, $C$ represents the "unshaded" color map, $N$ the normal vectors and $V$ the median line between $N$ and the observer direction. Both, $N$ and $V$ depend on the texture relief $H$; e.g. $N = (\partial H(i,j)/\partial i,\ \partial H(i,j)/\partial j)$. The coefficient $p$ represents the cosine power of the Phong reflectance model, $L_i$ the light source direction, $K_d$, $K_s$ and $K_a$ the other coefficients of the Phong shading model. $K_a$ is constant over the image. In the remaining parts, we will ignore specular highlights, which can be easily identified with our color-based segmentation. They will be reconsidered once the relief and "unshaded" color have been recovered.

Since we separate large-scale and small-scale relief, $H$ can be written as $H=H_l+H_s$. The problem is now to find coherent values for our unknowns, e.g. for $H_l$, $H_s$, and $C$. $K_d$ does not need to be explicitly recovered since it only influences the global brightness of the texture. For example, a user-defined $K_d$, lower than the actual $K_d$ of the texture, will darken the texture. $K_a$ is recovered by using the average brightness of shadow zones. If there are no shadow zones, we assume this value to be zero.

## 3 Small-scale relief recovery

Before dealing with the general case mentioned above, we will first consider the simpler case of small-scale relief. Let us assume first that there are no shadows, no specular highlights, that the color is constant (e.g. gray) and that there are no specific large-scale visual structures. These conditions are typically met by noisy textures. There have been a number of successful investigations in the past for synthesizing "noisy patterns" using an analysis of histograms and wavelet and/or Fourier transforms [5, 8]. Their success comes from the fact that such patterns are characterized by low order statistics, which can be well captured by the frequency domain. In fact, two "noisy patterns" can be considered as visually similar if spatial histograms and frequency distributions are similar.

We will now exploit this principle of visual similitude for relief recovery (not accurately but in a visually coherent and consistent way). Therefore, we assume that the "shading" due to a light source corresponds to a linear filtering of the bump map. Figure 2 illustrates three examples of synthetic bump maps, obtained using Perlin's [13] noise function. On the left, we show the original bump map where the darkest pixels are the deepest. Next, we show the respective frequency domains (note that Perlin's noise produces some important frequencies along the axes, which is a known phenomenon). Then, we show a shaded image using a directional light source (without computing inter-reflections), plus the corresponding frequency domains. It is visible that the shading has modified the frequency domains in two ways. Firstly, some directional frequencies (the direction matches the light source direction, rotated by $\pi/2$) have been filtered out. Secondly, high frequencies have been introduced. The right-most images in Figure 2 show the results that we obtain by applying a "reverse shading" filter (described below) and the corresponding frequency domains.

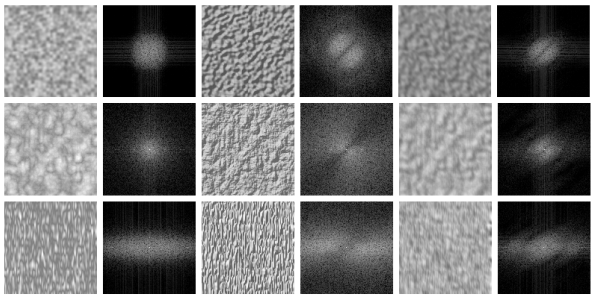

*Figure 2: Three examples of synthetic bump maps and their corresponding frequency domains. (left) the original bump maps, (middle) the shaded bump maps using a directional light source, (right) applying a "reverse shading" filter, as described in this paper.*

We can design a simple filter-based technique in order to recover a consistent bump map from a shaded image by using the two previous assumptions:

(1) noisy textures are visually alike if they have similar low order statistics (similar frequency domains);
(2) the shading due to a unique directional light source modifies in two ways the frequency domain of the original bump map.

We simply need to filter out some high frequencies and augment the amplitudes of the frequencies in the direction perpendicular to the light source direction.

In practice, the removal of high frequencies can be done using a low-pass filter (e.g. Gaussian), while the increasing of the other frequencies can be done with an anisotropic filter for which the direction matches the perpendicular direction of the light source. Figure 3 shows such filters for different directions (combined with the Gaussian filter). We need to make the height (value) of the directional filter depending on how grazing the light direction is with respect to the bump map, since we experienced that the amplitude of the "groove", caused by the shading in the frequency domain, diminishes when the light direction becomes vertical. The tangent to the angle of the light source direction with respect to the plane of the bump map can be used to obtain a filter depending on the light source direction. At strongly grazing angles, the value of the filter is infinite, while, with a vertical light direction, the height is 1 (e.g. frequencies are not modified and the filter is simply a Gaussian one). Indeed, in the latter case, there is no longer any visible groove, but high frequencies remain.

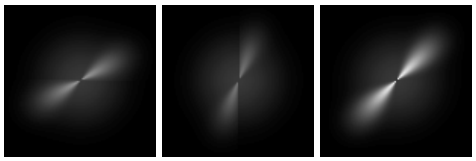

*Figure 3: Examples of filters in the frequency domain. (Note: in the frequency domain, filtering is obtained by multiplication, while, in the spatial domain, it is obtained by convolution).*

Once the filter has been designed, it is possible to obtain, by inverse Fourier transform, a convolution mask for filtering images of bumpy textures in the spatial domain. In practice, we use finite impulse response filters with very small sizes (3x3, or at most 7x7), firstly because of efficiency but also because the zones that we filter are usually small. The right-most part of Figure 2 illustrates the results of the filtering process, which we call "reverse shading filtering". The small size of the discrete filters explains that in our case the filtering produces some artifacts visible on Figure 2 (we do not recover exactly the original frequency domain). For example, it is visible that all the frequencies in the light source direction could not be restored (they were augmented but not "fully").

But, in spite of these artifacts, we experienced that the filtering process was yet already sufficient to restore coherent and consistent bump maps well, especially in the case of "noisy" textures. Figure 4 demonstrates this. We show the three previous synthetic bump maps applied to tori using lapped textures [14] that we extended to bump mapping. Note that the middle tori "naively" use the shaded image as a bump map (i.e. the middle images of Figure 2). This implies that each bump has become a smaller bump and a cavity and, thus, clearly introduces too high frequencies, as well as directional artifacts. Using filtering (Figure 2, right), we obtain results of much better quality (though not completely correct because of a slight "over-blurring" effect). Visible discontinuities on these images may be related to the lapped textures technique, which tends to produce seams on low frequency texture regions.

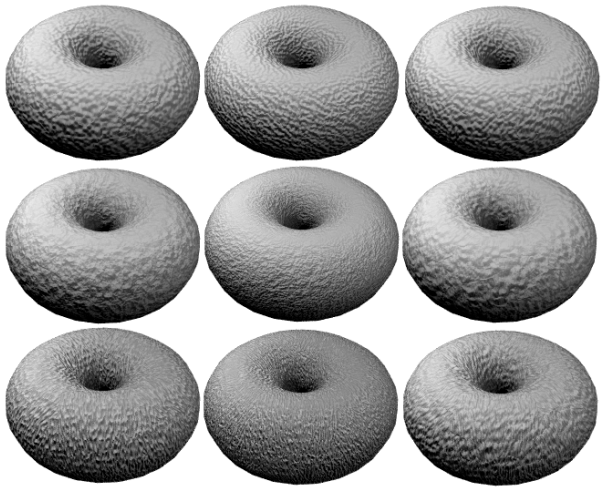

*Figure 4: Applying the synthetic bump maps of Figure 2 to tori. The left-hand tori show the references, the middle ones show the results obtained using a "naive" method consisting in directly applying the shaded images as bump maps, the right-hand ones show the results obtained using our filtering technique.*

## 4 Large-scale relief recovery

Natural texture images seldom meet the restrictive conditions that we set for small-scale relief. So, we will also need to consider shadows, specular highlights, hue, etc. Indeed, most textures are made of areas of different colors; and their shadows and highlights are mostly due to the large-scale structure of the texture.

### 4.1 Identifying large scale structures

The first step consists in identifying different color zones in the supplied texture image. For segmentation, we adopted a technique based on training sets, almost similar to the one used by Premoze et al. [15] for segmenting satellite images. With this technique, the user provides a training set, based on a few selected pixels. Figure 5 illustrates the segmentation of an image representing a synthetic bumpy texture composed of two different colors. Therefore, we obtained two zones after segmenting and the training set was composed of only two selected pixels, one for each color zone. Once the

image has been segmented into $k$ color zones, we can associate to each pixel a color index $c_k$. Each color zone can be composed of one or multiple connected components, which we will call "color structures" (dots in the case of Figure 5).

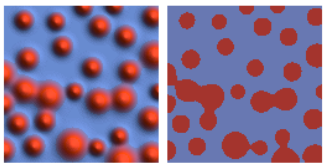

*Figure 5: An example of synthetic bump-texture image. (Left) the image. (Right) the segmented image.*

We now want to identify the "bumps" of the input image. To simplify the problem, we will assume that the bumps match the color structures. However, this does not mean that each bump is necessarily composed of a single color structure. On the contrary, one bump is often a composition (union) of multiple color structures. Grouping and classifying color structures requires users to click only on a few pixels.

We obtain two types of segmentation. The first corresponds to color structures, the second to the bump shapes. To compute the relief, we will now ignore specular highlights and shadows. These particular color zones can be easily identified since specular highlights generally correspond to the brightest color zone, while shadows generally correspond to the darkest zone. But, since not all images have specular highlights and/or shadows, we let users set a Boolean value for each. If the value is true, the system selects the brightest, respectively darkest zones as highlights respectively shadow. Otherwise, it assumes that there are none.

We assume that the relief of the bump structures is related to their shapes and, thus, it can be characterized by a single elevation curve. So, for all bump structures, we compute a discrete distance transform, based on erosion (this is a morphological operator). Figure 6 (left) illustrates this, in the case of the synthetic texture of Figure 5. The distance transform $D(B_s)$ provides for each bumpy structure $B_s$ a value between 0 and 1, where 0 means: "on the border"; and 1: "on the innermost part of the structure" (the innermost pixels can be considered as a sort of skeleton).

We are interested in finding the normal $N(i,j)$ on each pixel of each bumpy structure. The normal is defined with polar coordinates as:

$$\begin{cases} N_x = \cos(\alpha)\cos(\beta) \\ N_y = \sin(\alpha)\cos(\beta) \\ N_z = \sin(\beta) \end{cases}$$

Angle $\alpha$ is obtained using the previously computed distance transform. In fact, for each pixel of the bump, we can compute a closest "innermost" corresponding pixel (a pixel on the "skeleton"), which defines the direction (see Figure 6, middle part).

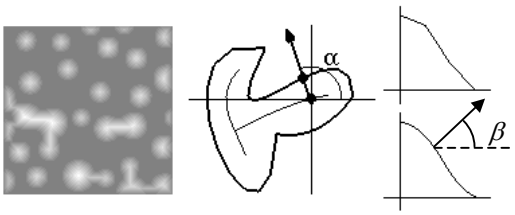


*Figure 6: (Left) erosion is used to provide a distance transform for each bumpy structure. (Middle) computing the normal direction using the skeleton; (right) smoothing the elevation curve.*

Now, we must compute the angle $\beta$ (see Figure 6, right-hand part). Angle $\beta$ is computed by using the shading information given by the texture image $T(i,j)$. $\beta$ is constant for pixels at the same distance $D$, e.g. for all pixels $p_i$, $p_j$, such that $D(p_i)=D(p_j)$. For a diffuse reflector, we have:

$$C(i,j)N(i,j)L_i = T(i,j). \qquad (1)$$

The color $C(i,j)$ depends on the color zone obtained by segmentation. We assume that the tint (hue and saturation) of pixels is rather constant inside one color zone and that the color variations are essentially due to the brightness, which depends on the relief. To obtain the constant color of each zone, we simply keep the color of the brightest pixel inside that zone (specular highlights are excluded), since on this particular pixel the light source direction approximately matches the normal (the cosine value is maximal, i.e. close to 1). Let us call $max_k$ the brightest color of the zone with index $c_k$. Now, in Equation 1, we can replace $C(i,j)$ by $max_k$. The remaining unknown in this equation is $\beta$. By noticing that $\sin^2+\cos^2=1$, we obtain an equation system of the second order that can be straightforwardly solved. Unfortunately, this system provides two solutions, among which we have to make a choice. The choice will be motivated by the following two statements. (1) The large-scale relief varies smoothly, so that, from one pixel to another, the normal only changes slightly. (2) On the skeleton, the normal is vertical. Now, we can compute one angle $\beta$ for each pixel by starting with the innermost pixels (skeleton) and by moving away to the border (on constant distance $D$ levels). Note that this procedure has some similarity with traditional SFS, for which the brightest pixels are also assumed oriented towards the light source. Each $\beta$ is chosen to minimize the difference with the previous one (i.e. on the previ-

ous distance level). For each level, we obtain a collection of angles $\beta$, i.e. for all pixels with the same distance $D$. Theoretically, all should be equal, but due to irregularities in the relief (natural textures are rarely perfectly smooth) and discretization errors, all are generally different. So, we must compute an average to reduce noise. The same is done for all bump structures, which provides us with angle series that are again averaged. We finally obtain, for each distance level $D$, a certain mean angle $\beta$, which can be used to determine a discrete elevation curve (see right part of Figure 6). When $\beta$ equals $\pi/2$, the elevation is null, otherwise we move up or down, proportionally to the cosine of $\beta$ (which can be negative). The obtained curve, often still noisy, can be further smoothed to obtain a continuous elevation variation according to the distance $D$.

Once we have an elevation curve, for the bumpy structures, we can compute a large-scale bump map $H_l(i,j)$ simply using the distance transform.

Figure 7 illustrates the relief we obtained for the synthetic bump texture of Figure 5 (in this case there is no small-scale relief). A rendering on a mushroom is also shown (visible discontinuities are due to the lapped textures). We obtain a result that matches well the "real" relief. The only noticeable problem is that the discrete distance transform has introduced edges on connected dots. This is because the dots are very small in terms of pixels. Thus, errors concerning the computation of $\beta$ are high, especially close to the skeleton (there is an increasing error on , as pixels come closer to the skeleton).

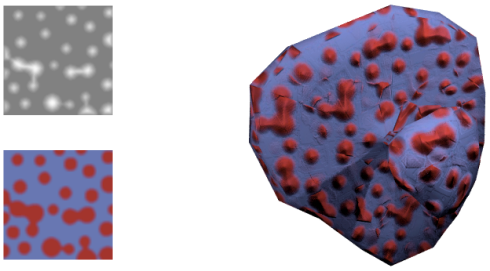

*Figure 7: Left: a recovered bump map (top) and an "unshaded" color map (bottom). Right: applying the texture to a mushroom (viewed from below). The result is consistent with the original image of Figure 5.*

### 4.2 Computation of an "unshaded" color map

The last step consists in computing an "unshaded" color map. It is easy to compute since we assumed that color variations are only due to the relief (except for user specified zones, which are simply ignored, as for Figurel). However, instead of using directly the segmented image as a color map, we blur it to avoid visible discontinuities. Finally, the areas corresponding to the specular highlights and shadows are filled with consistent patterns, either with a constant mean color or by a texture analysis and synthesis technique [20].

## 5 Combining large-scale and small-scale relief

As described in the previous chapters, the small-scale bump map represents all the details of a texture and the large-scale bump map represents its macrostructure. Very noisy textures with low amplitudes (which implies no cast shadows) only require a small-scale bump-map while textures made of large and very smooth parts only require a large-scale bump map. But, these kinds of textures are not the majority. In fact, most of the natural textures have a visible macrostructure made of large parts, which contain small details. Therefore, it is necessary to combine both large-scale ($H_l$) and small-scale ($H_s$) bump maps.

The resulting bump map ($H$) is simply computed by a weighted sum of both normalized bump maps, like this:

$$H = C_{mix}\,\mathrm{norm}(H_s) + (1 - C_{mix})\,\mathrm{norm}(H_l)$$

where $C_{mix}$ is a coefficient of blending from 0 to 1; 0 giving more importance to $H_l$ and 1 giving more importance to $H_s$. Function norm($x$) normalizes the bump map given as parameter so that its depth levels vary from 0 to 1.

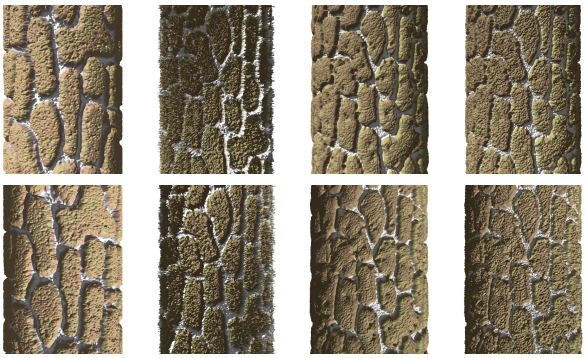

*Figure 8: Combining large scale and small-scale. From left to right: the accurate bump texture (reference image), the "naive" approach, our method with $C_{mix}$=25%, then $C_{mix}$=50%.*

Thus, the user simply chooses a value for $C_{mix}$ corresponding to the provided texture image, whether it is rather structured and smooth or noisy. Figure 8 illustrates an example with a synthetic texture. We compare the reference (obtained with known synthetic bump and color maps) with a "naive" approach that simply uses the gray-scale image as bump map and with our method using different values for $C_{mix}$, (25%, and 50%). Images have been rendered with two different light directions. Here, even a novice user could reckon intuitively that a good value for $C_{mix}$ is around 40% by observing closely

the original texture. Note that the last three sets of color and bump maps were extracted from an image of the reference texture rendered on a plane.

## 6 Results

We first tested the small-scale relief recovery in the case of "real" textures. Figure 9 illustrates some results for leather, bark and roughcast. On the left, we show the input image; in the middle, the straight utilization of this image as height field ("naive" approach) and on the right, the result after "reverse shading filtering". As for the synthetic examples of Figure 4, the straight use of the images introduces excessive noise. The results "look" more correct with the filtering approach even though, in these cases, we cannot compute reference images, since the relief information is not available.

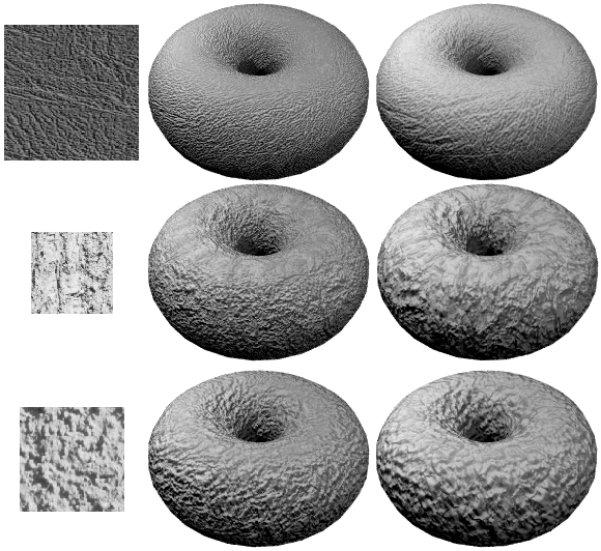

*Figure 9: Small-scale relief recovery for natural textures. From top to bottom: leather, bark and roughcast.*

While Figure 1 only illustrates large-scale relief recovery, Figure 10 and Figure 11 show the results achieved by combining the bump maps obtained with both large-scale and small-scale analysis. In all cases, we obtained consistent rendering results improved compared to conventional texture mapping and compared to a "naive" approach, which is too "noisy".

However, for texturing the trees of Figure 11 more realistically, we used displacement mapping instead of bump mapping. We also note that we did not use an "unshaded" color map on the trees. We used the original image instead, because too many colors were hard to distinguish from shading effects, especially for the second bark texture. Also, the dark cracks of the first bark texture were not only due to shadows. As a consequence, there is a slight "double shading" effect.

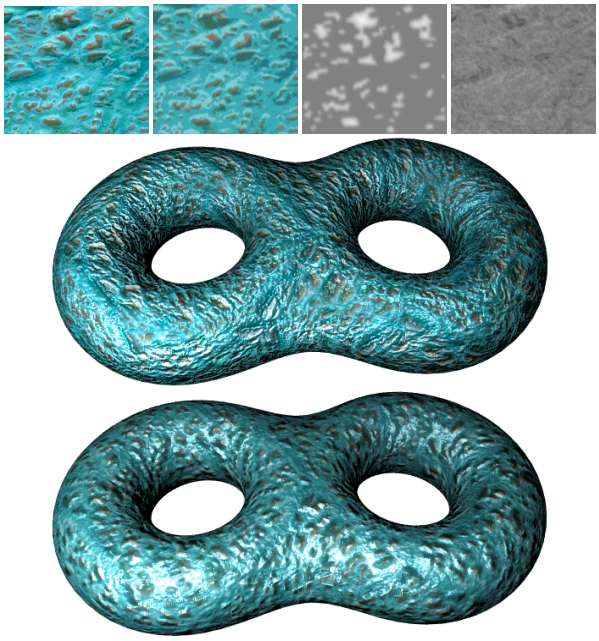

*Figure 10: An example of bumpy surface applied to some object. Top (from left to right): input image; "unshaded" color map; $H_l$; $H_s$. Middle: result with "naive" method. Bottom: result with our method.*

*Figure 11: Two examples of trees with bark textures applied to their limbs, using displacement mapping.*

## 7 Conclusions and future directions

Recovering a coherent relief from a single image is a difficult problem. In this paper, we have proposed a solution that consists in recovering the small-scale and large-scale relief separately. We used a filter-based method plus segmentation and classification. This works for many different types of textures, while users provide only some limited information (light source direction and training sets for segmentation and classification). This avoids a lot of empirical and manual work. In parallel, the use of a single image makes this method broadly accessible, in particular for collections of texture images available in diverse databases.

The method also computes an "unshaded" color map, which is satisfying when most of the details are given by the relief, rather than the colors. Otherwise, the original image should be used, though.

Nevertheless, due to the difficulty of the "shape from shading" problem, we had to make several simplifications. The most constraining is that we assumed that the large-scale relief could be characterized by a single curve. Though this is valid for many textures, it represents a restriction in some cases. So, we will need to improve the method on this point in our future works. A secondary problem is that, currently, our method cannot be used for reflectance recovery, since it does not provide an accurate relief, but only a visually consistent one. However, it should be possible to determine automatically the light source direction since the shading effect appears in the frequency domain. Some other information is given by shadows. The goal of further improvements would be to achieve a completely automatic method, for which the user would no longer have to provide any information at all.

## References


[1] J. F. Blinn. Simulation of wrinkled surfaces. In Computer Graphics (Proceedings of Siggraph'78), 12(3), pages 286-292, 1978.

[2] E. Catmull. A subdivision algorithm for computer display of curved surfaces. In *PhD thesis, Dept. of CS, University of Utah*, 1974.

[3] K. J. Dana, B. van Ginneken, S. K. Nayar, J. J. Koenderink. Reflectance and texture of real-world surfaces. In *ACM TOG*, 18(1), pages 1-34, 1999.

[4] J.M. Dischler. Efficiently rendering macro-geometric surface structures using bi-directional texture functions. In *Rendering Techniques'98 (Proceedings of Eurographics Workshop on Rendering)*, 1998.

[5] J.M. Dischler, D. Ghazanfarpour, R. Freydier. Anisotropic solid texture synthesis using orthogonal 2D views. In *Computer Graphics Forum*, 17(3), pages 87-95, 1998.

[6] J.M. Dischler, D. Ghazanfarpour. Interactive image-based modeling of macrostructured textures. In *IEEE CG&A*, 19(1), pages 66-74, 1999.

[7] J.M. Dischler, D. Ghazanfarpour. A survey of 3D texturing. In *Computers and Graphics*, 25(10), pages 135-151, 2001.

[8] D. J. Heeger, J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Computer Graphics (Proceeding of Siggraph'95)*, pages 229-238, 1995.

[9] B. K. P. Horn, M. J. Brooks. Shape from shading. In *MIT Press*, 1989.

[10] X. Liu, Y. Yu, H.Y. Shum. Synthesizing Bidirectional Texture Functions for real-world surfaces. In *Computer Graphics (Proceedings of Siggraph'01)*, pages 97-106, 2001.

[11] T. Malzbender, D. Gelb, H. Wolters. Polynomial texture maps, In *Computer Graphics (Proceedings of Siggraph'01)*, pages 519-528, 2001.

[12] M. M. Oliveira, G. Bishop, D. McAllister. Relief texture mapping. In *Computer Graphics (Proceedings of Siggraph'00)*, 2000.

[13] K. Perlin. An image synthesizer. In *Computer Graphics (Proceeding of Siggraph'85)*, 19(3), pages 287-296, 1985.

[14] E. Praun, A. Finkelstein, H. Hoppe. Lapped textures. In *Computer Graphics (Proceedings of Siggraph'00)*, pages 465-470, 2000.

[15] S. Premoze, W. Thompson, P. Shirley. Geospecific rendering of alpine terrain. In *10th Eurographics Rendering Workshop*, pages 115-126, June 1999.

[16] Rocchini C., Cignoni P., Montani C., Multiple textures stitching and blending on 3D objects, In *10th Eurographics Rendering Workshop*, pages 127-138, June 1999.

[17] H. Rushmeier, G. Taubin, A. Guéziec. Applying shape from lighting variation to bump map capture. In *Rendering Techniques'97 (Proceedings of Eurographics Workshop on Rendering)*, 1997.

[18] Y. Sato, M. D. Wheeler, K. Ikeuchi, "Object shape and reflectance modeling from observation", In *Computer Graphics (Proceedings of Siggraph'97)*, pages 379-387, 1997.

[19] G. Turk. Texture synthesis on surfaces. In *Computer Graphics (Proceedings of Siggraph'01)*, pages 347-354, 2001.

[20] L. Wei, M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *Computer Graphics (Proceedings of Siggraph'01)*, 2001.

[21] L. Ying, A. Hertzmann, H. Biermann, D. Zorin. Texture and shape synthesis on surfaces, In *EuroGraphics Workshop on Rendering*, 2001.

[22] R. Zhang, P.S. Tsai, J. E. Cryer, M. Shah, Shape from shading : a survey. In *IEEE Transactions on PAMI*, 21(8), pages 690-706, 1999.