

ARTISTIC IMAGE GENERATION BY DEVIATION MAPPING

LIU WENYIN

Microsoft Research China, 49 Zhichun Road, Beijing 100080, China
wyliu@microsoft.com

ZHONG HUA

Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA
zhonghh@cs.cmu.edu

XIN TONG XU YINGQING HEUNG-YEUNG SHUM

Microsoft Research China, 49 Zhichun Road, Beijing 100080, China
{xtong, yqxu, hshum}@microsoft.com

This paper describes a novel technique (deviation mapping) that generates images with artistic appearances, e.g., spray painted wall, embossment, cashmere painting, etc. Our technique employs a deviation map constructed from a single background image in the image generation process. Instead of recovering the exact geometry from the background image, the deviation map can be regarded as a virtual surface. This virtual surface is then painted with a foreground image and illuminated to generate the final result. Interestingly, the synthesized images exhibit some artistic appearances. Our method is very fast and very simple to implement.

Keywords: Artistic Image Generation; Deviation Mapping; Image Synthesis; Illumination.

1. Introduction

Generating an image with an artistic appearance is a challenging problem in computer graphics in general and image synthesis in particular. In computer graphics, a new image can be either synthesized from existing images, or rendered from a 3D model (e.g., a surface model) and its reflectance property in a given lighting condition.

Traditional synthesis methods usually use pixel-wise operations (e.g., simple color blending) or window operations (e.g., filters). Many commercial products such as Adobe Photoshop, Paint Shop Pro, or Microsoft PictureIt have many such functions built in for the users to create special effects. To synthesize images by composing, alpha blending¹ is usually applied. A pixel in an image contains not only color components (RGB), but also an additional transparency component (α). All four components (RGB α) of a resulting pixel in the composite image are weighted averages of their corresponding components in the two input images involved in the composition. Such methods are used not only in static image synthesis, but also in computer animation, e.g., fade-in, fade-out and dissolve.

If geometrical and photometric models of scenes or objects are known, one can always render an image given lighting conditions. Such models are, however, very

difficult to obtain in practice, especially from a single input image. Recovering surface geometry, surface normals, and object reflectance properties has been an active topic in computer vision and computer graphics for a long time. While some simple surface may be recovered using computer vision methods such as shape from shading^{2,3}, they usually do not work for complex surfaces, especially from a single image. Sato et al.⁴ estimated the surface normal (shape) from range images using a least square best fitting plane. Cromwell⁵ used a similar method for choosing the direction for viewing a cloud of small particles, e.g., molecules, in computer graphics. Hoppe et al.⁶ used the surface normal estimation method for surface reconstruction from a cloud of 3D points. However, these methods are not for image synthesis and do not apply for the single image input.

Creating artistic images with synthesis or rendering methods is difficult because it is subjective to define what “artistic” is. An example on synthesizing artistic images is by given Litwinowicz⁷. A Monet-style image is generated by blurring the input images at the right directions and with right size filters. Other excellent examples in non-photorealistic rendering include Pen-and-Ink illustration⁸, watercolor⁹ and digital facial engraving¹⁰.

In this paper, we present a novel technique called “deviation mapping” that can be used for generating images with artistic appearances. In the deviation mapping process, a deviation map is first constructed from a single background image. The deviation map contains only the deviation angle information, which is determined by the intensity of the background image at each pixel. The deviation map can be considered as a virtual surface, on which, a foreground image is then painted and illuminated to generate the final result.

The framework of our synthesis process is shown in Fig. 1, where the deviation map can be achieved using either a second image or the edge map (which is the image whose pixels are edge intensities/orientations of the original image pixels) of the foreground image. With a second image of a brick wall, for example, we can synthesize a spray painted wall effect. Other special effects, e.g., a cashmere texture image, a woven texture image, etc., can also be generated by using different textures as background. On the other hand, using the edge map of the input image, we also generate the special effect of embossment.

At first sight, deviation mapping proposed in this paper is similar to the bump mapping approach¹¹, which is widely used in image synthesis. However, there are several differences between the deviation mapping and bump mapping.

The bump map is generated from a height field that is sampled from the 3D geometric surfaces. By mapping the bump maps to other geometric surface, the accurate perturbed normal is used to compute intensity of the surface. As an image based geometry representation for surface details, the bump map can be represented as the original height field, the offset vectors in on the tangent plane of the surface, or the rotation vectors. Note that, although bump mapping greatly reduces the complexity of the geometry models, it is more expensive to render the bump map than to render the color texture mapping.

Different from the bump mapping, the deviation mapping is not a technique for 3D rendering but for image processing only. The deviation map is derived from a single 2D image without any knowledge of the 3D geometry for the image. Under some assumption of the lighting conditions (which is described in detail in Section 2), each pixel of the deviation map can be directly represented as a single deviation angle. As a result, we can regard the deviation map as a virtual-geometry based image representation or a virtual

surface, which is not sufficient to completely specify a physical surface. Compared with the bump map generation and mapping, the computation complexity of deviation mapping is much cheaper.

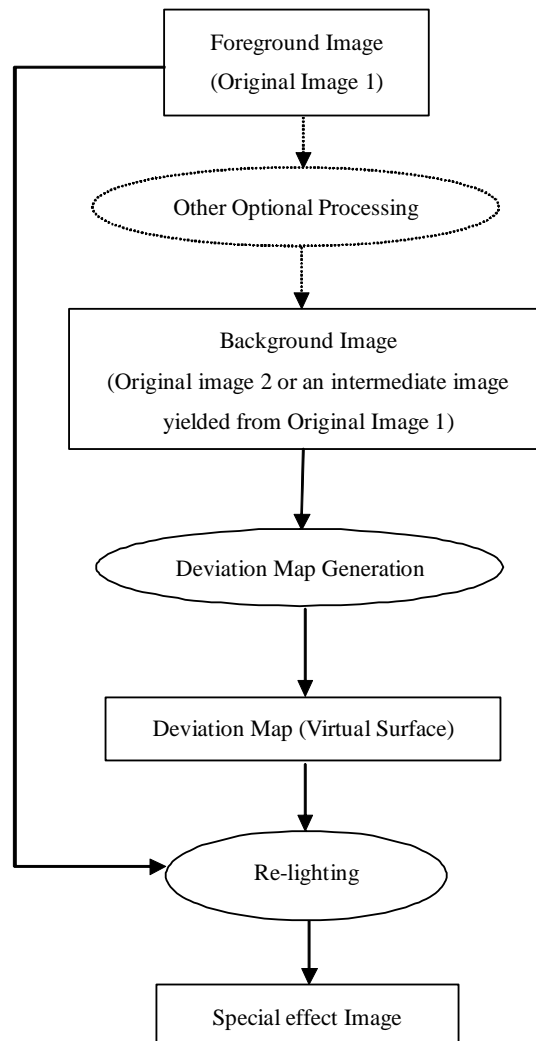


Fig. 1. An overview of our image synthesis method.

The remainder of this paper is organized as follows. After introducing the traditional illumination model, and making proper assumptions on viewing and lighting directions, the deviation mapping technique is explained in Section 2. Experiments with deviation mapping are shown in Section 3. We conclude this paper in Section 4.

2. Deviation Mapping

2.1. Illumination model

If a surface model is known, we can apply the Phong model¹², which is a simple illumination model described in the following equation and illustrated in Fig. 2, to the illumination of the surface.

$$I_{\lambda} = I_a k_a O_{d\lambda} + I_p \lambda [k_d O_{d\lambda} \cos \theta + W(\theta) \cos^n \alpha] \quad (1)$$

where, I is the result image intensity, λ is the color channel, which is one of Red, Green, and Blue, θ is the angle between the lighting direction (\mathbf{L}) and the normal (\mathbf{N}) of the virtual object surface, α is the angle between the viewing direction (\mathbf{V}) and the reflecting direction (\mathbf{R}), I_a is the ambient light, I_p is the lighting color, O_d is the color of the virtual object surface (the foreground image). In order to calculate Eq. (1), we have to specify all of these parameters (about the surface, lighting, etc.). One way to specify parameters is to assign values to them directly. Another way is make some assumptions such that some parameters can be calculated from other parameters, as we do in this paper.

If we assume that the viewing direction and lighting direction are the same as the original imagined plane, we get $\alpha = 2\theta$. Otherwise, the value of α should be specified in more complex ways. Although it is easy to calculate α from a real 3D surface, it is actually quite hard to calculate α from a single image, as in our situation. In this paper, we specify other parameters in the model as follows. O_d is represented by the intensity of the foreground image. I_p takes the value of white light or other tunable value. Other parameters can be constants and also be tunable. For example, $W(\theta)$ is typically set to a constant k_s , the material's specular-reflection coefficient, which ranges between 0 and 1.

If the surface can be seen as a Lambertian surface, which means that the reflected light intensity has no relation with the view direction, the specular part of the reflectance model will disappear. So the only information we need to know is the angle (θ) between the input light and the local normal of the surface, such that we have sufficient information to calculate Eq. (1), as explained in Section 2.2.

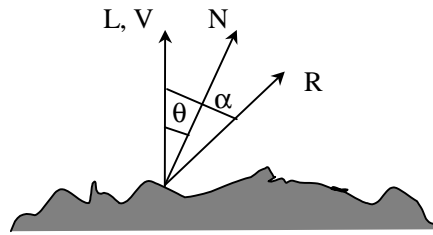


Fig. 2. The Phong illumination model (specular reflection).

Although the Phong model is used in our implementation, we believe that other illumination models can be used as long as the virtual surface provides enough surface

information for the calculation. For example, the Lambertian model can also be applied to our virtual surface, since the deviation angle is all what is required in the model.

2.2. Deviation mapping

A so-called deviation map $D(u,v)$ is derived from background image (in Fig. 1), in which each pixel stores a deviation angle $\theta(u,v)$ determined by the intensity of the background image at its corresponding pixel. The intensity (ranging 0~255) of the background image corresponds to the range of 90~0 degrees of the deviation angle. Hence we refer to the approach as deviation mapping. Given a foreground $I(u,v)$ and the deviation map $D(u,v)$, deviation mapping is executed as in Eq. (2), which is a simplified version of Eq. (1), where the $R(u,v)$ is the result image. As mentioned in Eq. (1), I_a , k_s and n are constant coefficients that can be adjusted by user as necessary.

$$R(u,v) = I_a + I(u,v)\cos(D(u,v)) + k_s \cos^n(D(u,v)) \quad (2)$$

As illustrated in Fig. 3, we can regard the deviation mapping as a simplified relighting procedure. We can imagine the foreground image $I(u,v)$ is painted on a plane that is orthogonal to our viewing direction (V), as shown in Fig. 3(a). After the normal at each pixel (u,v) is deviated by the angle $\theta(u,v)$, Eq. (2) is applied to compute the new color for each pixel. The foreground image is used as the diffuse color of the material. Note that, after deviation, the exact 3D vector expression of the normal is still not known. Hence, we called it “a virtual surface”. The virtual surface may be physically undetermined, i.e., there may be many such surfaces in real world. However, we may still use it as if it is a certain real surface in an illumination process provided that sufficient information (e.g., the angle in our case) is available.

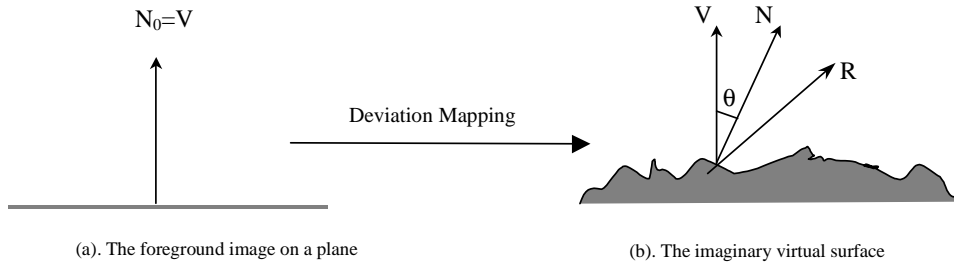


Fig. 3. Illustration of the imaginary virtual surface.

3. Applications

Our method is very fast because the operations involved are very simple. It takes less than one second for the method to synthesize moderate size images on a PII-400 PC with a 128MB memory. In all our experiments, we use the painting “La Lecon de Musique” by Matisse shown in Fig. 4 as the original foreground image.

We have used two different ways to generate the deviation mapping surfaces. The first way is to use a second image. Empirically, we have found that better effects are obtained if the background images are in rough appearances. For example, if a brick wall image is used, one gets the effect of spray painting on a wall, as shown in Fig. 5. We have also generated several other artistic effects, including cashmere painting (Fig. 6) and textile painting (Fig. 7).

The other way is to use a transformed image of the original one. For example, we can use the edge map of the original image as the deviation mapping source to generate the effect of embossment. Fig. 8 shows such embossment effect. The effect looks as though that the painter would have painted with thick oil paint layers.

4. Summary

In this paper, we have proposed a simple and fast image synthesis method—deviation mapping. The method first generates a deviation map of deviation angles of a virtual surface constructed from a background image. The virtual surface is then painted with a foreground image and illuminated to generate the final result. The deviation angle, which is determined by the intensity of the background image at each pixel, is sufficient to re-light the foreground image at such pixel with a simplified illumination model where the viewing direction is parallel to the incident light direction. Our method combines or superimposes two images to generate an artistic image. It has been applied to generate several artistic effects, including spray painted wall (wall painting), color embossment, and cashmere painting, textile painting. Experiments demonstrate that our method is efficient and effective.

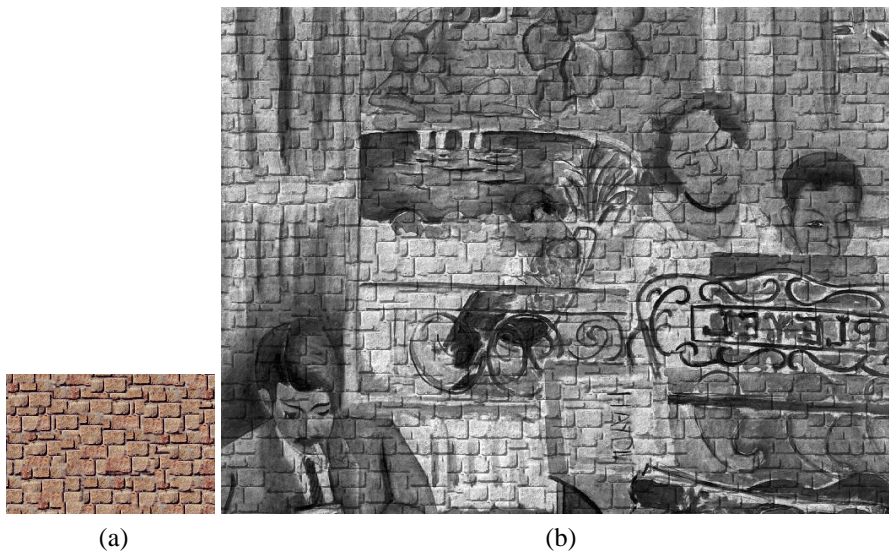
References

1. T. Porter and T. Duff, “Compositing digital images”, *SIGGRAPH* (1984), pp. 253-259.
2. B.K.P. Horn, “Shape from shading: A method for obtaining the shape of a smooth opaque object from one view”, (*MIT MAC TR-79*, Cambridge, MA. 1970) .
3. A.P. Witkin, “Recovering surface shape and orientation from texture”, *Art. Intell.* (1981), 17(1), pp. 17-45
4. Y. Sato, M.D. Wheeler, and K. Ikeuchi, “Object shape and reflectance modeling from observation”, *SIGGRAPH* (1997), pp. 379-387.
5. R.L. Cromwell, “Efficient eigenvalues for visualization”, in *Graphics Gem IV*, ed., P.S. Heckbet, (Academic Press, San Diego, 1994), pp. 193-198.
6. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points”, *SIGGRAPH* (1992), pp. 71-78.
7. P. Litwinowicz, “Processing images and video for an impressionist effect”, *SIGGRAPH* (1997), pp.407-414.
8. M.P. Salisbury, M.T. Wong, J.F. Hughes, and D.H. Salesin, “Orientable textures for image-based pen-and-ink illustration”, *SIGGRAPH* (1997), pp. 401-406.
9. C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D.H. Salesin, “Computer-Generated Watercolor”, *SIGGRAPH* (1997), pp.421-430.

10. V. Ostromoukhov, "Digital Facial Engraving", *SIGGRAPH* (1999), pp. 417-423.
11. J.F. Blinn, "Simulation of Wrinkled Surfaces", *SIGGRAPH* (1978), pp. 286-292.
12. B.-T., Phong, "Illumination for Computer Generated Pictures", *CACM* (1975), 18(6), pp.311-317.



Fig. 4. Original painting by Matisse.



(a)

(b)

Fig. 5. (a) A brick wall texture. (b) Synthesized image by mapping Fig. 4 onto (a).

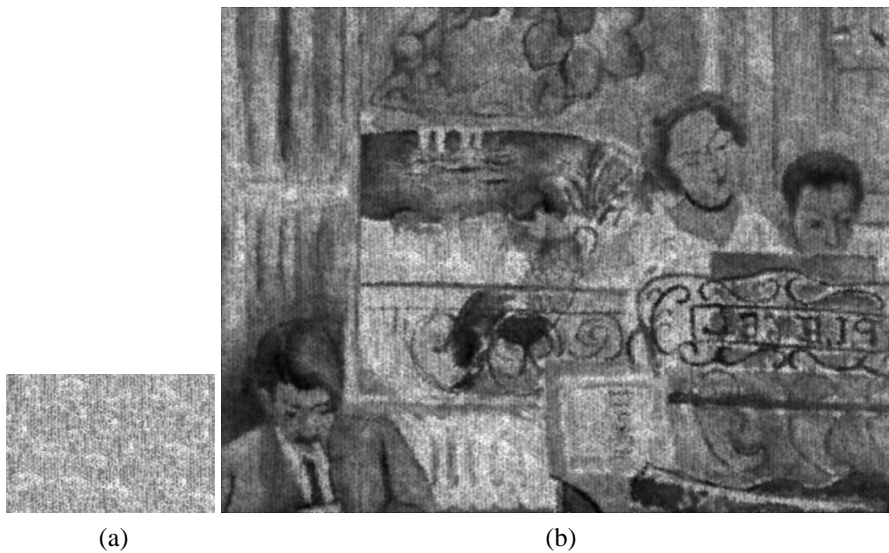


Fig. 6. (a) A cashmere texture used as the background image. (b) Synthesized image by mapping Fig. 4 onto (a).

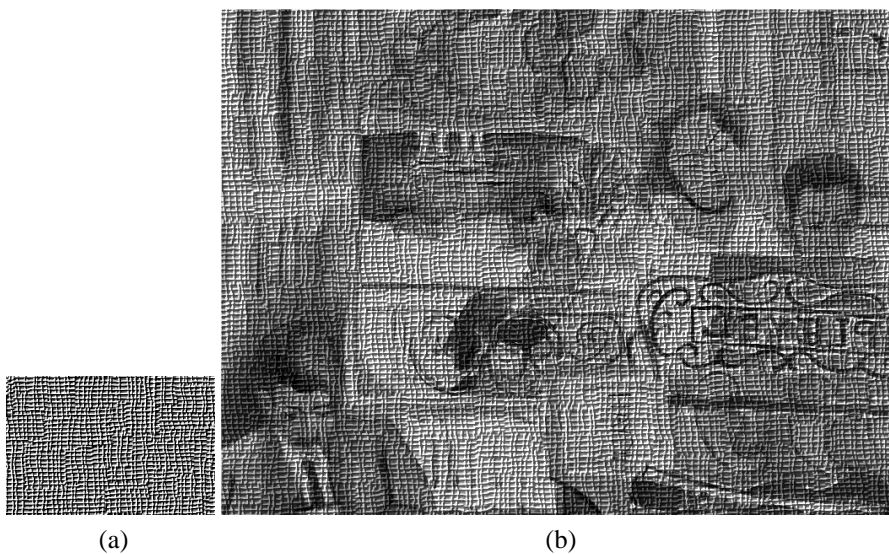


Fig. 7 (a) A woven (textile) texture. (b) Synthesized image by mapping Fig. 4 on (a).



Fig. 8. Color emboss effect.

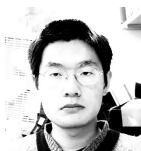
Photo and Biography



Liu Wenyin is a Researcher at Microsoft Research China since 1999. He received his B.Engg. and M.Engg. in Computer Science from Tsinghua University, Beijing in 1988 and 1992 respectively, and his D.Sc. in Information Management Engineering from Technion, Israel Institute of Technology in 1998.

His research interests include Web-based information systems, multimedia information management, graphics recognition, document analysis and recognition, performance evaluation, computer vision and pattern recognition, object-oriented programming, object-process methodology, and software engineering. He has authored more than 70 publications and five patents pending in these areas.

Liu Wenyin played a major role in developing the Machine Drawing Understanding System (MDUS), which won First Place in the Dashed Line Recognition Contest (<http://www.loria.fr/~tombre/grec95.html>) held during the First IAPR Workshop on Graphics Recognition at Pennsylvania State University, 1995. In 1997, he won Third Prize in the ACM/IBM First International Java Programming Contest (ACM Quest for Java'97) (<http://www.acm.org/jquest/webquest1.html>). He had co-chaired the fourth International Contest on Arc Segmentation held during the fourth IAPR Workshop on Graphics Recognition, Kingston, Canada, in September 2001. He is a member of ACM, IEEE, and IEEE CS.



Hua Zhong received his Bachelor degree of engineering from Tsinghua University, Beijing, PR China in 2000. After that, he worked with the Internet Graphics Group at Microsoft Research China from 2000-2001. Now he is a Ph.D student in Computer Science Department of Carnegie

Mellon University, Pittsburgh, USA.

His major interests are computer graphics and computer vision. He has worked in many projects of image based modeling and rendering, video object tracking, animation and morphing.



Xin Tong was born in Hebei, China and received his B.S. degree and Master Degree in Computer Science from Zhejiang University in 1993 and 1996, respectively. He received his Ph.D in Computer Graphics from Tsinghua University, Beijing in July 1999. Now he is a researcher in Microsoft Research China. His research interests include image based rendering and modeling, mesh compression and realistic rendering.



Yingqing Xu joined Microsoft Research China in January 1999 from his post as associate professor at the Chinese Academy of Sciences' Institute of Software. His research interests include computer animation, physically-based modeling, and image-based modeling.

He has won numerous academic prizes including the President's Scholarship of Chinese Academy of Sciences (1997), the Best Paper Prize of Chinagraph '96 and the Top Prize of Scientific and Technological Achievement from the Ministry of Machinery & Electronics Industry (1987). He has authored and co-authored 20 papers and two books in computer animation, computer simulation, computer art, physically-based modeling, and image based modeling.

Yingqing Xu was born in Beijing, China and received his B.S. degree in Mathematics from Jilin University in 1982. He received his Ph.D in Computer Graphics from the Institute of Computing Technology, Chinese Academy of Sciences in July 1997.



Harry Shum is the Assistant Managing Director of Microsoft Research China. Meanwhile, he also serves as the research manager in charge of the Visual Computing Group.

Harry joined Microsoft Research in 1996, after receiving a Ph.D. in Robotics from the School of Computer Science at Carnegie Mellon University. While at Carnegie Mellon, Harry undertook work at Digital Equipment Corporation's Cambridge Research Lab and with Apple Computer's QuickTime VR group, and was a consultant for RealSpace, Inc., a startup company at San Jose.

At Microsoft Redmond, Harry worked on image-based modeling and rendering -- building 3-D environments from a sequence of two-dimensional video images, and wandering around in the virtual environments. He has authored and co-authored over 40 papers in computer vision, computer graphics and robotics. He also has more than 20 US patents pending.

His current research interests include computer vision, image based rendering & animation, computer graphics, multimedia, human computer interface, and robotics. Most recently, he is working on unifying image-based rendering with traditional 3D computer graphics.