



ELSEVIER

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Non-photorealistic, depth-based image editing

Jorge Lopez-Moreno^{a,*}, Jorge Jimenez^a, Sunil Hadap^b, Ken Anjyo^c, Erik Reinhard^d, Diego Gutierrez^a^a Universidad de Zaragoza, Spain^b Adobe Systems, Inc., USA^c OLM Digital, Inc., Japan^d University of Bristol, UK

ARTICLE INFO

Keywords:

Non-photorealistic rendering
Relighting
Image processing
Human visual system
Real-time

ABSTRACT

Recent works in image editing are opening up new possibilities to manipulate and enhance input images. Within this context, we leverage well-known characteristics of human perception along with a simple depth approximation algorithm to generate non-photorealistic renditions that would be difficult to achieve with existing methods. Once a perceptually plausible depth map is obtained from the input image, we show how simple algorithms yield powerful new depictions of such an image. Additionally, we show how artistic manipulation of depth maps can be used to create novel non-photorealistic versions, for which we provide the user with an intuitive interface. Our real-time implementation on graphics hardware allows the user to efficiently explore artistic possibilities for each image. We show results produced with six different styles proving the versatility of our approach, and validate our assumptions and simplifications by means of a user study.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Whether the goal is to convey a specific mood, to highlight certain features or simply to explore artistic approaches, non-photorealistic rendering (NPR) provides an interesting and useful set of techniques to produce computer-assisted stylizations. Most of those techniques either leverage 3D information from a model, work entirely in 2D image space, or use a mixed approach (typically by means of a Z- or G-buffer) [1]. We are interested in exploring new possibilities for stylized depiction using a single image as input, while escaping traditional limitations of a purely 2D approach. For instance, the design of lighting schemes is crucial to communicate a scene's mood or emotion, for which depth information is required.

Our key observation is the fact that a single photograph or painting has richer information than we might expect. In particular, we ask ourselves what layers of information present in an image may have been usually overlooked by stylized depiction techniques? And what would the simplest way to access that "hidden" information be, in a way that allows dramatic manipulation of the look of an image?

This paper presents a set of stylization techniques that deals with a single photograph as input. It is well known that when it comes to stylized depiction, human perception is able to build complex shapes with very limited information, effectively filling in missing detail whenever necessary, as illustrated in Fig. 1 (left). The

power of suggestion and the influence of light and shadows in controlling the emotional expressiveness of a scene have also been extensively studied in photography and cinematography [2,3]: for instance, carefully placed shadows can turn a bright and cheerful scene into something dark and mysterious, as in Fig. 1 (right).

With this in mind, we propose a new class of methods for stylized depiction of images based on approximating significant depth information at local and global levels. We aim to keep the original objects recognizable while conveying a new mood to the scene. While the correct recovery of depth would be desirable, this is still an unsolved problem. Instead, we show that a simple methodology suffices to stylize 3D features of an image, showing a variety of 3D lighting and shading possibilities beyond traditional 2D methods, without the need for explicit 3D information as input. An additional advantage of our approach is that it can be mapped onto the GPU, thus allowing for real-time interaction.

Within this context, we show stylized depictions ranging from simulating the *chiaroscuro* technique of the old masters like Caravaggio [4] to techniques similar to those used in comics. In recent years, both the movie industry (Sin City, A Scanner Darkly, Renaissance, etc.) and the photography community (more than 4000 groups related to comic art on Flickr) have explored this medium. The goal of obtaining comic-like versions of photographs has even motivated the creation of applications such as Comic Life.¹

* Corresponding author.

E-mail address: jlmo@unizar.es (J. Lopez-Moreno).¹ <http://plasq.com/comiclife-win>.



Fig. 1. Left: The classic image of “The Dog Picture”, well known in vision research as an example of emergence: even in the absence of complete information, the shape of a dog is clearly visible to most observers (original image attributed to R. C. James [5]). Right: Example of dramatically altering the mood of an image just by adding shadows.

2. Previous work

Our work deals with artistic, stylized depictions of images, and thus falls under the NPR category. This field has produced techniques to simulate artistic media, create meaningful abstractions or simply to allow the user to create novel imagery [6,7]. In essence, the goal of several schools of artistic abstraction is to achieve a depiction of a realistic image, where the object is still recognizable but where the artist departs from the accurate representation of reality. In this departure, the object of depiction usually changes: a certain mood is added or emphasized, unnecessary information is removed and often a particular visual language is used.

In this paper, we explore what new possibilities can be made available by adding knowledge about how the human visual system (HVS) interprets visual information. It is therefore similar in spirit to the work of DeCarlo and Santella [8] and Gooch et al. [9]. DeCarlo and Santella propose a stylization system driven by both eye-tracking data and a model of human perception, which guide the final stylized abstraction of the image. Their model of visual perception correlates how interesting an area in the image appears to be with fixation duration, and predicts detail visibility within fixations based on contrast, spatial frequency and angular distance from the center of the field of view. Although it requires the (probably cumbersome) use of an eye-tracker, as well as specific per-user analysis of each image to be processed, the work nevertheless shows the validity of combining perception with NPR techniques, producing excellent results.

Instead, we apply well-established, general rules of visual perception to our model, thus freeing the process from the use of external hardware and individual image analysis. The goals of both works also differ from ours: whilst DeCarlo and Santella aim at providing meaningful abstraction of the input images, we are predominantly interested in investigating artistic possibilities.

Gooch and colleagues [9] multiply a layer of thresholded image luminances with a layer obtained from a model of brightness perception. The system shows excellent results for facial illustrations. It is noted that in their approach some visual details may be difficult (or even impossible) to recover. Although in the context of facial stylization this counts as a benefit, it might not be desirable for more general imagery.

Depth information has previously been used to aid the generation of novel renditions. For instance, ink engravings can be simulated by estimating the 3D surface of an object in the image, and using that to guide strokes of ink [10]. This method is capable of producing high-quality results, although it requires the user to individually deform 3D patches, leading to a considerable amount

of interaction. The algorithms proposed by Oh et al. [11] cover a wide range of image scenarios with specific solutions to extract 3D data for each one, but also come at the expense of considerable manual input. Okabe and colleagues [12] present an interactive technique to estimate a normal map for relighting, whereas in [13], painterly art maps (PAMs) are generated for NPR purposes. While both works show impressive results, they again require intensive, skilled user input, a restriction we lift in our system.

In their work, Raskar and colleagues [14] convey shape features of objects by taking a series of photographs with a multi-flash camera strategically placed to cast shadows at depth discontinuities. Akers et al. [15] take advantage of relighting to highlight shape and features by combining several images with spatially varying light mattes, while in [16] details are enhanced in 3D models via exaggerated shading. In contrast, our approach operates on single off-the-shelf images, allows for new, artistic lighting schemes, and requires at most a user-defined mask to segment objects, for which several sophisticated tools exist [17,18].

In the field of halftone stylization based on 3D geometry we should mention the recent work of Buchholz et al. [19], which incorporates information from shading, depth and geometry in order to generate boundaries between black and white regions which run along important geometric features for shape perception (like creases).

Bhat et al. [20] proved the potential of gradient-based filtering in the design of image processing algorithms like painterly rendering or subtle image relighting.

A 2.5D approach has been explored in the context of video stylization [21], aiding the production of hatching and painterly effects. This method, however, requires the specific calibrated capture of the 2.5D video material to be processed, which is still either cumbersome or expensive.

Lopez-Moreno et al. [22] showed that 2.5D approximations suitable for NPR can be obtained from off-the-shelf images by applying current theories about the perception of shape. In the present paper we have extended their work in a two-fold manner: First, we have explored two new stylization methods based on more complex rendering methods; ambient occlusion and global illumination. And second, we have developed an interactive editing interface which complements the edition of lighting by providing the user with full artistic control over the generation of color, shading and shadows.

3. Perceptual background

At the heart of our algorithm, which will be described in the next section, lies the extraction of *approximate* depth information from the input image. Since we do not have any additional information other than pixel values, we obviously cannot recover depth accurately, and therefore the result will potentially contain large errors. However, given that we are interested in stylized depictions of images, we will show that we do not require physical accuracy, but only approximate values which yield pleasing, plausible results. Our depth approximation algorithm leverages some well-known characteristics of the human visual system. Although the inner workings of human depth perception are not yet fully understood, there exist sufficient indicators of some of its idiosyncracies that enable us to approximate a reasonable depth map for our purposes. In particular, we rely on the following observations:

1. Belhumeur et al. [23] showed that for unknown Lambertian objects, our visual system is not sensitive to scale transformations along the view axis. This is known as the *bas-relief ambiguity*, and due to this implicit ambiguity large scale errors

along the view axis such as those produced in many single view surface reconstruction methods tend to go unnoticed.

2. Human vision tends to reconstruct shapes and percepts from limited information, for instance filling in gaps as shown in Fig. 1, and is thought to analyse scenes as a whole rather than as a set of unconnected features [24,25].
3. Causal relationships between shading and light sources are difficult to be detected accurately [26]. The visual system does not appear to verify the global consistency of the light distribution in a scene [27]. Directional relationships tend to be observed less accurately than radiometric and spectral relationships.
4. There is evidence that human vision assumes that the angle between the viewing direction and the light direction is 20–30° above the view direction [28].
5. In general, humans tend to perceive objects as globally convex [29].

In the following three sections we describe our algorithm and its applications while, in Section 7 we will show the results of a user test validating our assumptions.

4. Algorithm

We rely on prior knowledge about perception, summarized above, to justify the main assumptions of our depth approximation algorithm. In particular, the bas-relief ambiguity (Observation 1) implies that any shearing in the recovered depth will be masked by the fact that we will not deviate from the original viewpoint in the input image [30]; in other words, we assume a fixed camera. The second and third observations suggest that an NPR context should be more forgiving with inaccurate depth input than a photorealistic approach, for instance by allowing the user more freedom to place new light sources to achieve a desired look, as we will see. Finally, the combination of the first, fourth and last observations allows us to infer approximate depth based on the dark-is-deep paradigm, an approach used before in the context of image-based material editing [31] and simulation of caustics [32].

The outline of the process is as follows: first the user can select any object (or objects) in the image that should be treated separately from the rest. Usually the selection of a foreground and a background suffices, although this step may not be necessary if the image is to be manipulated as a whole. We assume that such selection is accomplished by specifying a set of masks using any existing tool [17,18].

In the last step of the process, the user can specify new light sources as necessary (for which object visibility will be computed), and choose from a variety of available styles.

4.1. Depth recovery

Our goal is to devise a simple depth recovery algorithm which works well in an NPR context and offers the user real-time control for stylized depiction. We aim to approximate the main salient features without requiring a full and accurate depth reconstruction. We take a two-layer approach, following the intuition that objects

can be seen as made up of large features (low frequency) defining its overall shape, plus small features (high frequency) for the details. This approach has been successfully used before in several image editing contexts [33,34,16], and has recently been used to extract relief as a height function from unknown base surfaces [35]. We begin by computing luminance values on the basis of the (sRGB) pixel input using $L(x,y) = 0.212 \cdot R(x,y) + 0.715 \cdot G(x,y) + 0.072 \cdot B(x,y)$ [36]. Then we decompose the input object in the image into a base layer $B(x,y)$ for the overall shape as well as a detail layer $D(x,y)$ [33], by means of a bilateral filter [37]. Additionally, as the methods based on the dark-is-deep assumption tend to produce depth maps biased towards the direction of the light, we smooth this effect by filtering $B(x,y)$ with a reshaping function [31] which enforces its convexity, producing an inflation analogous to those achievable by techniques like *Lumo* [38].

The detail layer D can be seen as a bump map for the base layer B . We decouple control over the influence of each layer and allow the user to set their influence in the final image as follows:

$$Z(x,y) = F_b \cdot B(x,y) + F_d \cdot D(x,y) \quad (1)$$

where $Z(x,y)$ is interpreted as the final, approximate depth, and F_b and F_d are user-defined weighting factors to control the presence of large and small features in the final image respectively, both independent and in the range [0, 1]. Fig. 2 shows the results of different combinations of the base and detail layer of the teddy bear image, using the halftoning technique described in Section 5. The depth Z is stored in a texture in our GPU implementation (lower values meaning pixels further away from the camera). Fig. 3 shows 3D renderings of the recovered depth for an input image; it can be seen how depth inaccuracies are more easily noticed if the viewpoint changes, while they remain relatively hidden otherwise.

The depth map Z serves as input to the relighting algorithm. Although a normal map could be derived from the depth map, it is not necessary for our purposes (except for the color relighting effect explained in Section 5).

4.2. Computing visibility for new light sources

The user can now adjust the lighting of the scene by defining point or directional light sources, to obtain a specific depiction or mood of the image. In the following, we assume a point light source



Fig. 3. Recovered depth from a given image. Errors remain mostly unnoticed from the original viewpoint (left), but become more obvious if it changes (right). Light and shadows have been added for visualization purposes.



Fig. 2. Different combinations of the detail and base layer yield different depictions (here shown for the halftoning technique). From left to right: original image, base and detail layers, plus different depictions with a fixed $F_b=1.0$ and increasing F_d from 0 to 1 in 0.25 increments.

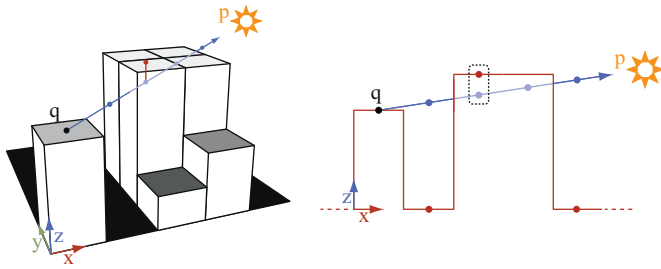


Fig. 4. 3D and lateral views of the visibility computations for each texel.

at $\mathbf{p} = (p_x, p_y, p_z)^T$. There are no restrictions on where this light source can be placed.

Visibility is then computed on the GPU (in a similar fashion as other techniques such as parallax mapping [39]): for each pixel in the framebuffer $\mathbf{q} = (x, y, z(x, y))^T$ belonging to an object we wish to relight, the shader performs a visibility test for the light (see Fig. 4), by casting a ray towards its position. The pixels visited between \mathbf{q} and \mathbf{p} are given by Bresenham's line algorithm. The z -coordinate of the ray is updated at each step. Visibility is determined by querying the corresponding texels on the depth map. This information will be passed along to the specific NPR stylization techniques (see Section 5). Once a pixel visibility has been established, we can apply different NPR techniques to produce the desired stylized depiction of the image.

5. Stylization examples

We show a variety of examples which are currently implemented in our system. In each case, the defining difference over existing NPR work is the ability to relight the original image on the basis of the recovered 2.5D depth information. This adds versatility and artistic freedom. The different effects can be combined in layers for more complex looks, as some of our results show.

Halftoning: By simply mapping pixels visible from a light source to white and coloring all other pixels black, a halftoned rendition of the image is achieved. Fig. 5 shows two examples of new relighting from an original input. Starting from a single image, we first create a halftoned version similar to what can be achieved with other systems (we use the implementation described in [34], where the authors present a method based on segmentation from energy minimization). The remaining sequence of images in this figure shows the application of two novel lighting schemes that leverage the recovered depth information, thereby extending the capabilities of previous approaches. In the first one, a point light source has been placed at (165,240,450) (in pixel units), whereas the second is lit by a directional light in the x direction. The weighting between detail and base layers is $(F_b, F_d) = (1.0, 0.9)$ for both images.

Multitoning: The spatial modulation of more than two tones (such as the black and white used in halftoning, plus several shades of gray) is known as multitoning or multilevel halftoning. In our implementation the user sets the position of a light source, after which a set of new lights with random positions located nearby the original is automatically created (the number of new lights is set by the user). This approach creates visually appealing renditions without having to place all light sources manually. Visibility is then computed separately for each light, and the results are combined in a single output by setting the value of each pixel in the final image to the average of the corresponding pixels in each layer. Results are shown in the second and sixth images in Fig. 17 (in reading order) and the middle image of Fig. 6 for three different inputs.

Dynamic lines: When sketching, an artist may draw lines towards the light source to add a more dynamic look to the scene. We can



Fig. 5. From left to right: Input image. Output yielded by halftoning as described in [34] (both images courtesy of D. Mould). Result lit by a close point light. Another result lit by a directional light.

emulate a similar technique just by direct manipulation of the depth map. We randomly select a set of object pixels; the probability of choosing a specific pixel is set to be inversely proportional to the Euclidean distance to the position of the considered light source. The depth values of the selected pixels are altered, effectively changing the results of the visibility computations in the image and casting shadows which are perceived as lines. The third and ninth image in Fig. 17 show final examples using this technique.

Color relighting: For each pixel belonging to the object, we compute a normalized surface normal $\vec{n}(x, y)$ from the gradient field $\nabla z(x, y)$ [31]:

$$\vec{g}_x(x, y) = [1, 0, \nabla_x z(x, y)]^T \quad (2)$$

$$\vec{g}_y(x, y) = [0, 1, \nabla_y z(x, y)]^T \quad (3)$$

$$\vec{n}(x, y) = \vec{g}_x \times \vec{g}_y / \|\vec{g}_x \times \vec{g}_y\| \quad (4)$$

Using this normal map as well as the 3D position of a light source, it is straightforward to modify pixel luminances or shading as function of the angle between the normals and the lights. Figs. 6, 7 and 17 show examples with Gouraud shading. The color is extracted from the original image RGB values, converted to its corresponding value in Lab space and its luminance is set to a middle constant value. The initial albedo is obtained by combining the RGB original value with this luminance-attenuated value. The user can control this mixing, which is limited to pixels originally not clamped to black or white (where chromatic information is not available). The result is used as multiplying albedo by the color stylization methods.

Ambient occlusion: Local render methods like Phong shading fail to achieve the visual quality obtained by global illumination techniques. A crude yet effective method of approximating global illumination is the usage of ambient occlusion. It allows us to take into account attenuation of light due to occlusion of near surfaces. Occlusion is calculated by casting rays in the upper hemisphere of the rendered point, which allows us to obtain a binary value that describes whether the ray is occluded by a surface or if it is able to reach the background, usually referred to as the sky. An average is performed on these binary values, obtaining a visibility value. This visibility value is then usually multiplied with the ambient term of the lighting equation. In our case we multiply it with the output of the color relighting shader, as we are looking for an stylized result. In the following lines, the method used for computing ambient occlusion will be described, which is based on the Starcraft II approach [40]: it is one of the most elaborated methods, and already proven to work in a general, non-controlled environment.

However, casting rays in every direction of the hemisphere rules out real-time manipulation. Solutions to this problem have been presented in the form of screen-space methods that approximate occlusion by using simple depth comparisons. Thus, instead of casting rays in each direction, a randomized n -set of (x, y, z) offsets are used to query depth at different positions. Then, a depth value $Z(x, y)$ is compared with the z component of the corresponding offset; if z is greater than $Z(x, y)$, it is assumed that there is no geometry blocking at that offset. The result of this comparison is a binary value that is averaged similarly to the ray casting approach,



Fig. 6. Application of our method to a very diffusely lit image. In this example we aim to obtain different moods by changing the light environment and the degree of stylization. Left: Original input image. Middle: A very stylized and dark version of the input by multitoned depiction with four point light sources at (140,400,300), (140,400,350), (140,400,400) and (140,400,900) and using $(F_b, F_d)=(1.0,0.2)$. Right: Less stylized depiction obtained by combination of multitone and color relighting effects with lights at (134,530,290), (115,15,270), (315,695,350), (100,400,1000) and (589,325,325). No mask was used for these depictions.

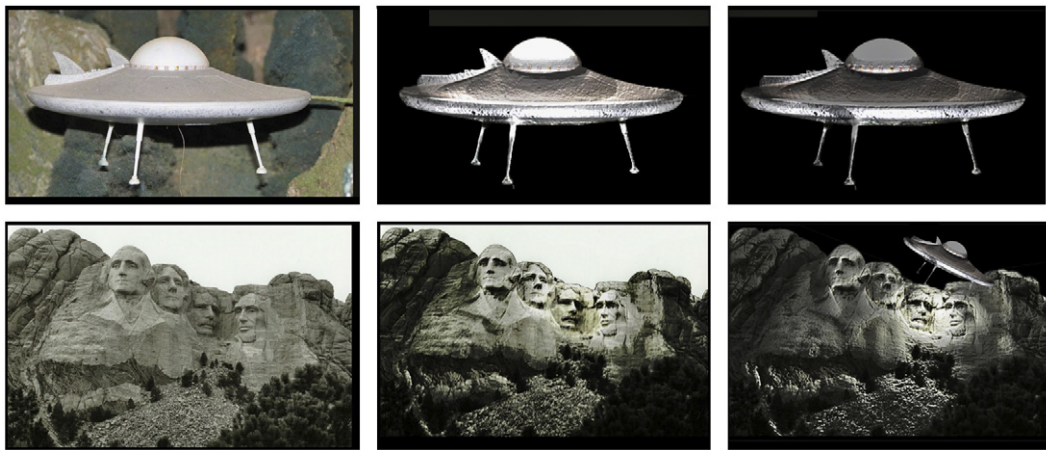


Fig. 7. Composition of results. Top row, left: Original input image. Top row, middle: Color relighting with five point light sources: two from above at $x=480, y=520, z=(500, 250)$ and three surrounding the disk at $x=(50,550,100), y=400, z=1000$, and using $(F_b, F_d)=(1.0,0.1)$. Top row, Right: Result of multiplying a shadow layer created by a light source at (580,0,500) and the relighted image (middle). Second row, from left to right: Original input image, stylized depiction by combination of color relighting and halftone, and result of compositing the relighted UFO from top row and a new relit version of the input image.

which yields an approximated visibility term for a pixel (x,y) :

$$V(x,y) = \sum_{i=0}^n \frac{Z(x_i,y_i) < z_i}{n} \tag{5}$$

To achieve real-time rendering, only a few samples can be used, usually between 8 and 32. Sampling uniformly using such low sample counts leads to banding artifacts. To improve image quality, randomized sample positions are used. An 8×8 randomized texture containing normalized vectors is tiled in screen-space, giving each pixel its own random vector r . However, a set of n random offsets is required for each pixel. They are passed as fragment shader constants and reflected using each pixel's unique random vector r , effectively giving a semi-random set of n offsets for each screen pixel. To avoid self-occlusion, offset vectors are flipped when they point inwards with respect to the surface normal (which is obtained in the same way as in the color relighting shader).

Randomizing the sampling position trades banding for noise. It yields better results, but by itself it is unable to produce high-quality results. To deal with the resulting noise, a smart Gaussian blur is performed that takes into account differences in depth, which enables the removing of noise from the calculated visibility while avoiding visibility bleeding along object edges.

An important piece of a screen-space ambient occlusion shader is the attenuation function. It must be chosen with care, in order to



Fig. 8. Ambient occlusion effect. Left: Input image. Right: The result of combining three different attenuation values. By increasing b (2.0, 4.0 and 8.0) we obtain local occlusion (detail) and medium-range occlusion (smooth shading). The depth map was generated with $F_b=1.0$ and $F_d=0.3$.

prevent far away objects from generating occlusion among themselves. Instead of simply comparing depth with the z component of the offset, a delta $e=z-Z(x,y)$ is calculated. This delta is then modified by the attenuation function. As done by Filion and McNaughton [40], a linearly stepped attenuation function is used, where delta values less than an artist-chosen constant c give an occlusion of 0, whereas values higher than c are modified using $a \cdot \text{abs}(e)^b$. All the images used in this work have empirically fixed

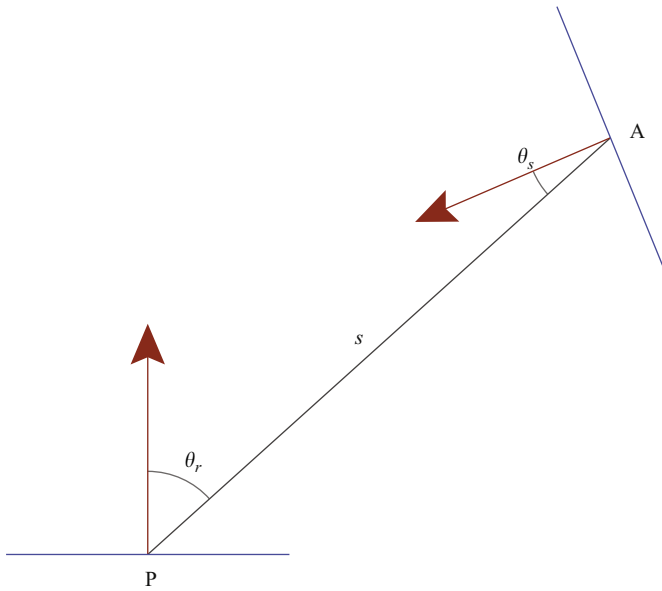


Fig. 9. Radiance is transmitted from sender point *A* to receiver point *P*. The distance between both points is used to calculate the attenuation term $1/s^2$. On the other hand, the angles θ_s and θ_r are used to compute how much radiance is arriving at point *P*, as it depends on the orientation of both surfaces. Figure adapted from [42].

values of $c=0.05, a=10.0, b=2.0$. The right image in Fig. 8 shows the result of multiplying three passes of ambient occlusion with different attenuation values ($b=2.0, 4.0$ and 8.0) to obtain a pencil-style depiction of a photograph.

The most correct approach for sampling is to convert depth values to camera space, add the randomized offsets, then project to screen space. However, we cannot transform depth values to eye-space positions as the projection matrix of an image is not known. Therefore, a simpler approach is used, where sampling is entirely done in screen space [41].

For more details about screen-space ambient occlusion we refer the reader to the existing bibliography [40–42].

Global illumination: A natural extension to ambient occlusion is the inclusion of an indirect bounce of global illumination [42]. The scene must be modified first using color relighting, and stored in the direct radiance texture *L*. Then, in a second pass, ambient occlusion and global illumination are calculated together. For each sample position given by the randomized offset vectors, the radiance contribution $L(x,y)$ from sampled point *A* to current point *P* is calculated taking into account both the normal at the sampling position *A*, the normal at current point *P* and the attenuation produced as the light travels between the two points (see Fig. 9):

$$L_{ind}(x,y) = \sum_{i=0}^n \frac{L(x_i,y_i) \cdot \cos(\theta_{s,i}) \cdot \cos(\theta_{r,i})}{s_i^2}, \tag{6}$$

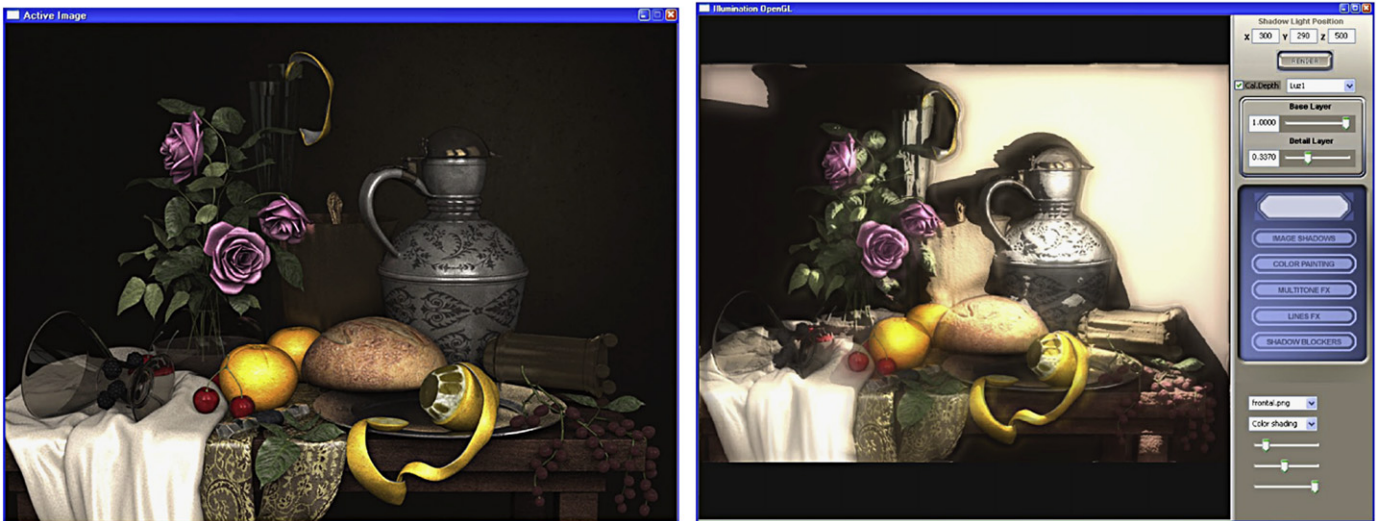


Fig. 10. Example of the global illumination user interface. The dials (at the bottom of the right panel) are set to (offset) = 0.15 (maximum screen offset to take samples), $\alpha = 0.5$ and $\beta = 1.0$.



Fig. 11. Some examples of global illumination effect. From left to right: Input image, relighting with $\alpha = 1.0$ and $\beta = 1.0$ and light source at $(80,1000,500)$, relighting with $\alpha = 1.0$ and $\beta = 2.0$ and light source at $(570,500,597)$. In this case the offset is set to 0 to over illuminate the image, producing an interesting glow effect. Finally, relighting with two light sources at $(50,920,230)$ and $(315,400,438)$. α and β are set to $(1.0,0.8)$. Note the color bleeding (red) produced at the jaw. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\theta_{s,i}$ and $\theta_{r,i}$ are the angles between the transmission direction and the sender and receiver normals respectively, and s is the distance between the points P and A .

The final pixel value, using both ambient occlusion and global illumination is given by the following equation:

$$P(x,y) = ((1-\alpha) + \alpha \cdot V(x,y)) \cdot L(x,y) + \beta \cdot L_{ind}(x,y), \quad (7)$$

where α and β control the strength of the ambient occlusion and global illumination effects, respectively. The parameter α has a valid range of values of $[0..1]$. On the other hand, floating point values greater than or equal to zero are appropriate for the β parameter. See Fig. 11 for some examples of this effect. Fig. 10 shows another relighting example with our user interface. With three dials, the user can control α and β values and the range for the



Fig. 12. Top row: Example of relighting with ambient occlusion and global illumination effects (with a light source placed at (570,320,710)). The iron figure in the right was masked out from the input image and was affected by two additional light sources at (468,535,420) and (376,200,500) to produce highlights in the body and illuminate the shadowed area of the head, respectively. Middle row: Input image and the result of combining multitone rendering with global illumination from three light sources (one placed in front of each eye and a third centered in the mouth). α was set to 1.0 and β to 2.0 to overexpose the original colors, producing a watercolor-comic book effect. Bottom row: From left to right: Input image, color relighting with a top-left light source at (146,1000,532) and global illumination relighting ($\alpha = 1.0$ and $\beta = 1.0$) with a bottom light at (334,65,464). Note the effect of the light bouncing in the area marked by the white rectangle.

offset of the samples taken. Additional results of stylization techniques are shown in Fig. 12.

6. Image retouching interface

In order to incorporate local control over the stylization process we have developed a real-time interactive brush. The artist can paint directly over the image with the mouse to alter the underlying geometry of the image thus altering the resulting stylization: modify the shading, set how shadows are cast, highlight areas, etc. Our tool allows for edits like those shown by Todo and colleagues [43] in stylized depictions of 3D models. However, our work is based on a depth map without an associated implicit 3D surface therefore this kind of edition fits in the same category as approaches like gradient painting [44] or depth painting [45]. This tool is motivated both by the increased degree of artistic control it provides and the inherent inaccuracy of automatic depth map generation. In most cases, the automatically generated depth maps produce perceptually plausible depictions. However, in some

scenarios this method yields results which may be non-plausible at certain regions of the image. This can be due to number of reasons such as the limitations of the shape from shading technique, the violation of our assumptions about the input (materials, global convexity, ...), or even when reconstructing well-known geometries like a human face.

Shadow blockers: To further enhance artistic control over the generation of specific shadows, the user can paint directly over the image with the mouse, and the depth associated to the corresponding pixels is modified to block light and thus cast shadows. Fig. 13 shows an example of a projected pattern and user-defined cast shadows. Note that these can be colored as well.

Depth sculpting tools: We have implemented the basic depth painting operations described by Kang [45]: shift depth by addition and subtraction (carve) and both global and local bump (see Fig. 14). Both bump effects have an area of influence which is inversely weighted by the distance to the central pixel in the screen plane. However, in the case of the local bump the difference in depth is also considered. Additionally we have developed a smoothing brush which performs a gaussian convolution of Z values (see Fig. 15 for an example of use).

Albedo painting: For color relighting techniques, the user can modify the albedo color of the image without affecting its 3D shape. The initial albedo is combined with the color of the brush in Lab space.

Lighten/darken: This tool allows the user to freely add localized light and shadows to an object in a manner that is consistent and seamlessly integrated with the current light environment and it is inspired by the work of Todo et al. [43] which shows how to add intentional, even unrealistic, shade and light edits in NPR cartoon stylization. Intuitively, they force the shade and light boundaries to follow the user strokes as much as possible while yielding a plausible solution. To do so, they establish a set of boundary constraints based on the user strokes and try to find a displacement function for the underlying surface which, taking into account the light direction, yields the desired shade/light boundary. In order to make their strategy computationally tractable at interactive rates, they represent the offset function with a sum of radial basis functions (RBF) and solve the linear problem for the desired curvature and boundary restrictions.

In our case, rather than affect shade/light boundaries, we intended to lighten or darken a local area by modifying its shading while keeping boundary coherency with the rest of the surface. To achieve this we have to shift their normals towards the light's direction (and do the opposite to darken it). Our approach is based on the convolution of the depth map with a gaussian function; the brush has a radial area where the influence of the brush decays exponentially having a value equal to zero in its boundary. Additionally each user's stroke has only a delta addition/subtraction to the depthmap values, subsequently shifting the normals towards the light direction in a small quantity. This behavior is analogous to the RBF technique in the sense that there will be a smooth blending between the modified area (sum of gaussian radial functions produced by multiple strokes) and the original depthmap of the image.



Fig. 13. Adding mystery with shadows, cornerstone of the *noir* genre. Left: Original image. Right: Output yielded by a simple blocker which simulates light coming through blinds. $(F_b, F_d)=(0.6,0.9)$.

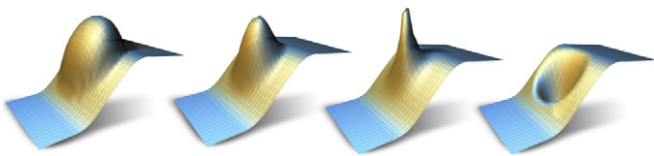


Fig. 14. Results of applying different brushes to the depth map. The artistic control is given by the parameters of a Gaussian function centered at the brush. From left to right, the degree of decay is increased (*pinch effect*) with the rightmost figure showing a carving example (depth subtraction).

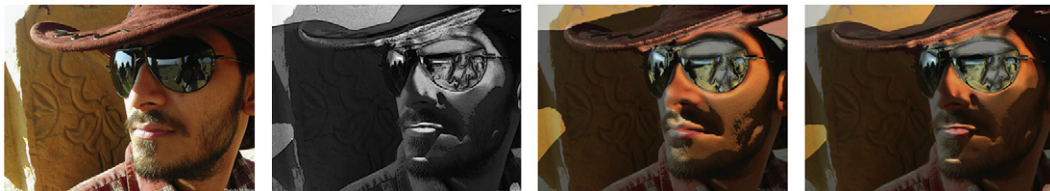


Fig. 15. Example of depth editing. From left to right: Input image and relighting result with light source at $(550,400,460)$, image obtained with automatic depth map generation and after being edited by an artist with our tools for 5–10 min. The retouching tools helped in both correcting noticeable mistakes from depth generation (the emboss effect of the sunglasses) and creating a more interesting combination of shading and shadows (nose, lips, cheeks, jaw, ...).

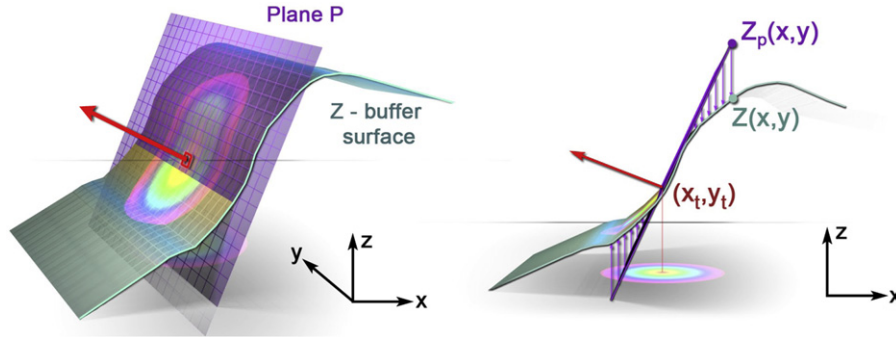


Fig. 16. Two views of the depth map Z showing the virtual plane P used to shift the normals in the area of the brush (centered at pixel (X_c, Y_c)). The false radial colors illustrate the decay of the effect applied by the brush, which is adding $(Z_p(x,y) - Z(x,y))$ to the depth value of each pixel $Z(x,y)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To force the local normals to be oriented in a particular direction, we built a plane defined by that direction and the 3D position of the pixel corresponding to the center of the brush. We then modified each of the neighboring pixel's depth $Z(x,y)$ in direct relation to their distance to the plane P (see Fig. 16). The computed variation of depth per pixel is weighted by its distance s to the center of the brush (t_x, t_y) in the screen plane (See Eq. (8)). The distance is computed by using a Gaussian distribution with a scale λ and a standard deviation σ set by the user. A minimum value of one third of the brush's radius for σ ensures a smooth interpolation near the boundaries:

$$s = \lambda \cdot e^{-((x-t_x)^2 + (y-t_y)^2)/\sigma^2}$$

$$Z(x,y) = Z(x,y) + s \cdot (Z_p(x,y) - Z(x,y)) \quad (8)$$

where $Z_p(x,y)$ is the depth value of the plane P at pixel (x,y) .

All the aforementioned techniques can be applied to both base and detail layers independently or in a combined way. In this fashion the artist has control over the range of the tool, editing the overall shape (base) and/or the local bumps (detail).

7. Evaluation

In order to test our algorithm and the assumptions it relies on, we devised a psychophysical experiment to objectively measure how inaccurate the recovered depth is, compared to how well these depth maps work in an NPR context. The test is designed as follows: we take a rendered image of a 3D scene of sufficient diversity, having both complex and simple shapes, and a wide range of materials including transparent glass. Since it is a synthetic scene, its depth information is accurate and known, and we can use it as ground truth. We then generate two additional depictions of the same scene, changing the lighting conditions. The original image has the main light falling in front of the objects at an angle from right-above; we thus create two very different settings, where light comes (a) from the camera position (creating a very flat appearance) and (b) from behind the objects. Together, the three lighting schemes (which we call original, front and back) plus the variety of shapes and materials in the scene provide an ample set of conditions in which to test our algorithm. Fig. 18, top, shows the three resulting images.

We then compare the ground-truth depth map of the 3D scene with each of the approximate depths recovered using our image-based algorithm (with $F_b=1.0$ and $F_d=0.3$ according to Eq. (1)). Fig. 18 (middle and bottom rows) shows the four depth maps, the alpha mask used to define foreground and background, and the base and detail layers for each approximate depth map. Note that

the ground-truth depth is the same for the three images, whereas our approximated depth is different since it depends on pixel values.

Table 1 shows the results of the L_2 metric and correlation coefficient (considering depth values pixel by pixel): our algorithm cannot recover precise depth information from just a single image, but the correlation with the ground truth is extremely high. Additionally, we also compare with a gray-scale version of the Lena image and with gray-level random noise (with intensity levels normalized to those of the 3D scene render), in both cases interpreting gray levels as depth information; both metrics yield much larger errors and very low, negative correlation. These results suggest that our simple depth extraction method approximates the actual depth of the scene well (from the same point of view, since we are dealing with static images). The question we ask ourselves now is, is this good enough for our purposes? In other words, is the error obtained low enough to achieve our intended stylized depictions of the input image, without a human observer perceiving inconsistencies in the results?

One of the main advantages of our approach over other image-based stylization techniques is the possibility of adding new light sources. We thus explore that dimension as well in our test: for each of the three input images, we create two new lighting schemes, one with slight variations over the original scheme, and one with more dramatic changes. Finally, for each of the six resulting images, we create halftoning, multitoning and color relighting depictions, thus yielding a total of 18 images.

Given that the ultimate goal of our test is to gain some insight into how well our recovered depth performs compared to real depth information, for each of the 18 stimuli we create one version using real depth and another using recovered depth. We follow a two-alternative forced choice (2AFC) scheme showing images side-by-side, and for each pair we ask the participants to select the one that looks better from an artistic point of view. A gender-balanced set of 16 participants (ages from 21 to 39 years old) with normal or corrected-to-normal vision participated in the experiment. All participants were unaware of the purpose of the study, and had different areas of knowledge and/or artistic backgrounds. The test was performed through a web site, in random order, and there was no time limit to complete the task (although most of the users reported having completed it in less than 5 min). Fig. 19 shows some examples of the stimuli, comparing the results using real and approximate depth, for the three stylized depictions.²

Fig. 20 summarizes the results of our test, for the three styles (halftoning, multitoning and color relighting) and two light variations (similar, different). The bars show the percentage of participants that chose the depiction using our method over the one

² Please refer to the supplementary material for the complete series.



Fig. 17. Stylized results achieved with our method. Top row, left: Original input image. Top row, right: Multitone depiction with two point light sources at (506,276,1200) and (483,296,900), and using $(F_b, F_d) = (0.5, 0.8)$. Second row, left: Multitone image with two layers of dynamic lines added, generated from the same light at (500,275,1000). Second row, right: Result of multiplying color relighting with the multitone version. Third row, from left to right: Mask with foreground objects (window painted manually for artistic effect and motivate subsequent relighting), multitone depiction of *Vanitas*, and result of multiplying two layers of color relighting and five layers of dynamic lines (please refer to the supplementary material to see the individual layers). Fourth row, from left to right: Original input image, *Dynamic lines* version placing a light source at both headlights, and a multilayer combination similar to *Vanitas* figure above.

generated with real depth (ground truth). Despite the relatively large errors in the approximate depth (as the metrics from Table 1 indicate), the results lie very closely around the 50-percent mark. We run a significance test on our results. Our hypothesis is that, despite the sometimes obvious differences in the depictions due to the different depths employed, there is no significant difference in the participants' choices when judging the resulting artistic stylizations. The differences in preference percentage for each of the aforementioned techniques are 0.04762, 0.09524 and 0.02439, which is in all the cases below 0.15121, the standard error for a 95% of confidence. Therefore, we can assure that there is no significant preference for actual depth over approximated depth in our test.

8. Discussion

We have shown results with a varied number of styles, all of which have been implemented on the GPU for real-time interaction and feedback, including relighting.³ Our simple depth approximation model works sufficiently well for our purposes, while allowing for real-time interaction, which more complex algorithms may not achieve. On a GeForce GTX 295, and for a 512×512 image and a single light source, we achieve from 110 to 440 frames per second. Performance decays with the number of lights: in our

³ Please refer to the video.

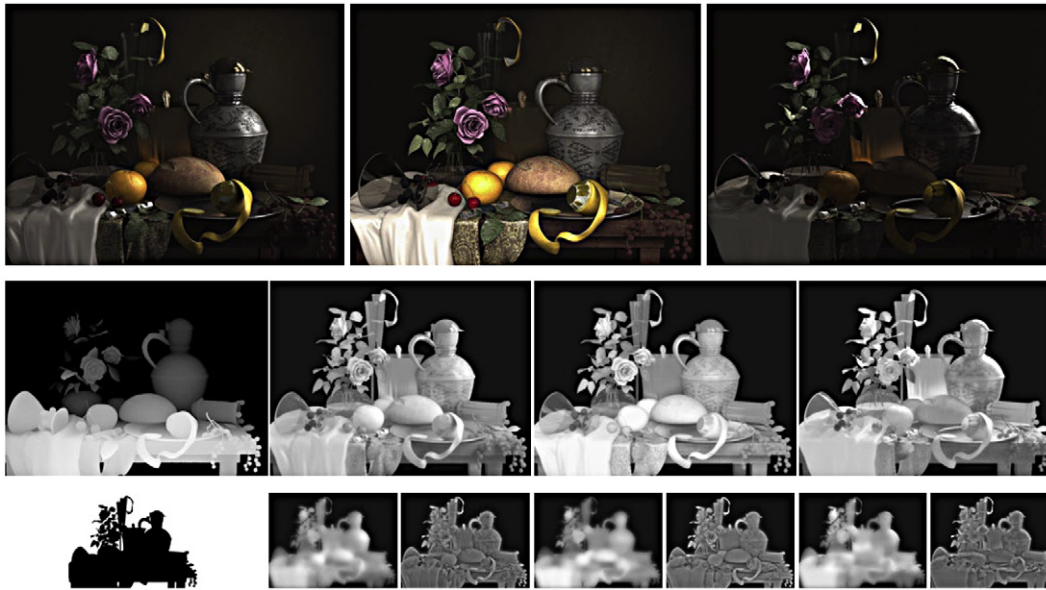


Fig. 18. First row: The three rendered images used as input in our test, lit by the original, frontal and back illumination schemes respectively. Second row: Ground-truth depth map obtained from the 3D information of the scene (bumpmaps not included), plus approximate depths recovered for each of the input images. Third row: Alpha mask, plus the base and detail layers of each image, used to obtain the corresponding depth maps.

Table 1

Results of the L_2 metric and correlation coefficient comparing the ground-truth depth of the 3D scene with the approximate depth extracted from each input image, plus a gray-scale version of the Lena image and gray-level random noise (interpreting gray levels as depth).

Input image	L_2	Corr
Original	100.16	0.93
Front	120.47	0.952
Back	121.66	0.925
Lena	383.92	-0.138
Random noise	524.74	-0.00075

tests, real-time operation can be maintained with up to 5 light sources on average.

Our approach has several limitations. If the convexity assumption is violated, the depth interpretation of our method will yield results which will be the opposite to what the user would expect them to be. For small features it usually goes unnoticed, but if the object is not globally convex the results may not be plausible. Wrong depth interpretations from the dark-is-deep paradigm, such as the teddy bear’s nose in Fig. 2, can also be taken as intrusive regions; thus, expected cast shadows and relighting may look wrong in that area. Our method also assumes relatively Lambertian surface behavior: while highlights could be removed through thresholding or hallucination techniques, our assumptions on the perception of depth are broken in the case of highly refractive or reflective objects. In the latter case, shape-from-reflection techniques could be investigated. Also, since we do not attempt to remove the original shading from the image, our technique could potentially show artifacts if new lights are placed in the same direction of existing shadows (see Fig. 21). However, our results confirm that quite large shading inaccuracies tend to go unnoticed in a NPR context. We think that future research with different shape from shading techniques could clarify if simpler methods [46] can still produce plausible depictions or even if more sophisticated techniques might extend the applicability to photorealistic image editing. Finally, since we recover only depth information from camera-facing object pixels, completely accurate shadows cannot be produced.



Fig. 19. Examples of the stimuli used in our user test, for the halftoning (top row), multitone (middle row) and color relighting styles (bottom row). Left and right columns were obtained with approximate and real depths, respectively.

Our method could potentially be used for video processing, for which temporal coherence should be considered. For the dynamic lines stylization technique proposed here, this could be very complicated since it would most likely require tracking features at pixel level. Video segmentation is also a difficult task that would be necessary to address (although as some of the images in this paper show, compelling results can also be achieved in certain

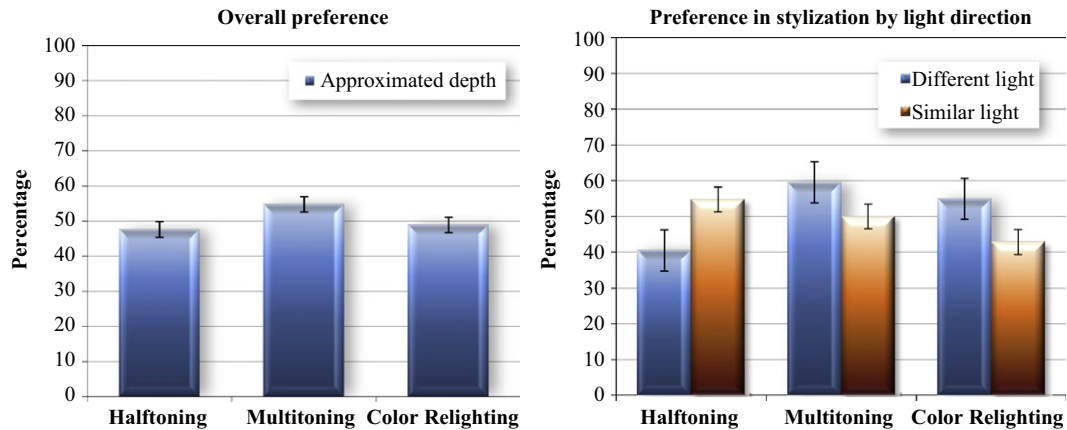


Fig. 20. Percentage of participants that chose the depiction using approximate depth over the one generated with real depth, for the three styles (halftoning, multitoning and color relighting). Left: Average preference for all the images used in the test. Right: Preference in stylization considering the light direction: similar and differing from the original light source in the relighted images.

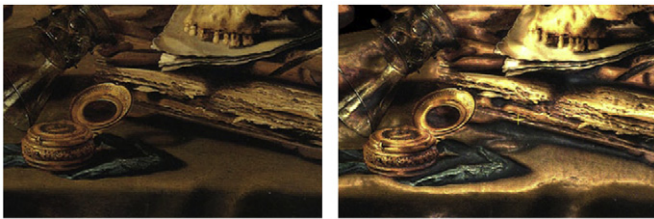


Fig. 21. Artifacts due to original shadows in the image. Left: Detail of the original image depicted in Fig. 17. Right: Relighting with a light source at (510, 520, 740) wrongly illuminates the shadowed areas.

cases by processing the image as a whole). Finally, we expect that advances in the fields of perception and shape-from-shading will provide more exciting new grounds for artistic depiction of images and video.

9. Conclusions

We have presented a new methodology to develop NPR techniques based on the recovery of information about the depth from input images. Relying on known characteristics of human visual perception, our work offers more flexibility and artistic freedom than previous approaches, including the possibility of extreme relighting of the original image. Accurate extraction of depth information from a single image is still an open, ill-posed problem for which no solution exists. In this work we have shown that while our recovered depth is not accurate enough for certain applications, non-photorealistic stylization of images provides a much more forgiving ground, masking possible inconsistencies and leaving the abstraction process unhampered.

The fact that the algorithm also works well with a painted image (*Vanitas*) is quite interesting: a human artist painting the scene performs inaccurate depth recovery and very coarse lighting estimation, and the perceptual assumptions made by our algorithm seem to correlate well with the human artistic process. Future work to develop a system that mimics this process more closely can give us valuable insight and become a very powerful NPR tool.

Our 2.5D interpretation of objects in images yields an appropriate basis for appealing visual effects. We have shown several applications for this approach, such as halftoning, multitoning, dynamic lines, color relighting, ambient occlusion and global illumination, but many more effects could be devised (e.g., relighting with non-Lambertian reflectance models). Furthermore, we have developed a set of real-time

tools which allows the user to overcome the limitations of our automatic depth acquisition, providing full artistic control over the generation of color, shadows and shading. The work by Bhat et al. [20] could be combined with our approach in order to produce a wider range of visual effects. We think that future computer-aided 2D image editing techniques will benefit from a similar combination of underlying geometry (automatically generated and/or user-made) and the knowledge of the related human perception processes.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments and suggestions, Stephen Withers for the 3D still scene used in Fig. 18, and Ignacio Echevarria for the different renderings. Also the following Flickr users should be credited for the images used in our examples: M. Omair, kk+, onkel_wart, drspam, jasohill, Jan Van Schijndel, JF Sebastian, gmeurope, 844steamtrain, and Sunset Chaser. This research was partially funded by a generous gift from Adobe Systems Inc, a Marie Curie Grant from the Seventh Framework Programme (Grant agreement no. 251415), the Spanish Ministry of Science and Technology (TIN2010-21543) and the Gobierno de Aragón (projects OTRI 2009/0411 and CTPP05/09). Jorge Jimenez is funded by a grant from the Gobierno de Aragón.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version of [10.1016/j.cag.2010.11.007](https://doi.org/10.1016/j.cag.2010.11.007).

References

- [1] Durand F. An invitation to discuss computer depiction. In: NPAR '02: Proceedings of the second international symposium on non-photorealistic animation and rendering. New York, NY, USA: ACM; 2002. p. 111–24.
- [2] Kahrs J, Calahan S, Carson D, Poster S. Pixel cinematography: a lighting approach for computer graphics. In: ACM SIGGRAPH course notes; 1996. p. 433–42.
- [3] Alton J. Painting with light. Berkeley: University of California Press; 1945.
- [4] Civardi G. Drawing light and shade: understanding Chiaroscuro (the art of drawing). Search Press; 2006.
- [5] Marr D. Vision. New York: W. H. Freeman and Company; 1982.
- [6] Strothotte T, Schlechtweg S. Non-photorealistic computer graphics; 2002.
- [7] Gooch B, Gooch A. Non-photorealistic rendering; 2001.
- [8] DeCarlo D, Santella A. Stylization and abstraction of photographs. ACM Trans Graph 2002;21(3):769–76.

- [9] Gooch B, Reinhard E, Gooch A. Human facial illustrations: creation and psychophysical evaluation. *ACM Trans Graph* 2004;23(1):27–44.
- [10] Ostromoukhov V. Digital facial engraving. In: *SIGGRAPH '99: Proceedings of the 26th annual conference on computer graphics and interactive techniques*. New York, NY, USA: ACM; 1999. p. 417–24.
- [11] Oh BM, Chen M, Dorsey J, Durand F. Image-based modeling and photo editing. In: *SIGGRAPH '01: Proceedings of the 28th annual conference on computer graphics and interactive techniques*. New York, NY, USA: ACM; 2001. p. 433–42.
- [12] Okabe M, Zeng G, Matsushita Y, Igarashi T, Quan L, yeung Shum H. Single-view relighting with normal map painting. In: *Proceedings of pacific graphics 2006*; 2006. p. 27–34.
- [13] Yan C-R, Chi M-T, Lee T-Y, Lin W-C. Stylized rendering using samples of a painted image. *IEEE Trans Visualization Comput Graph* 2008;14(2):468–80.
- [14] Raskar R, Tan K-H, Feris R, Yu J, Turk M. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In: *SIGGRAPH '04: ACM SIGGRAPH 2004 papers*. New York, NY, USA: ACM; 2004. p. 679–88.
- [15] Akers D, Losasso F, Klingner J, Agrawala M, Rick J, Hanrahan P. Conveying shape and features with image-based relighting. In: *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. Washington, DC, USA: IEEE Computer Society; 2003. p. 46.
- [16] Rusinkiewicz S, Burns M, DeCarlo D. Exaggerated shading for depicting shape and detail. In: *SIGGRAPH '06: ACM SIGGRAPH 2006 papers*. New York, NY, USA: ACM; 2006. p. 1199–205.
- [17] Li Y, Sun J, Tang C-K, Shum H-K. Lazy snapping. In: *Siggraph*. Los Angeles, CA: ACM; 2004. p. 303–8.
- [18] Rother C, Kolmogorov V, Blake A. GrabCut: interactive foreground extraction using iterated graph cuts. In: *Siggraph*. Los Angeles, CA: ACM; 2004. p. 309–14.
- [19] Buchholz B, Boubekeur T, DeCarlo D, Alexa M. Binary shading using geometry and appearance 2010; 29(6):1981–92.
- [20] Bhat P, Zitnick L, Cohen M, Curless B. Gradientshop: a gradient-domain optimization framework for image and video filtering. *ACM Trans Graph* 2010;29:1–14.
- [21] Snavely N, Zitnick CL, Kang SB, Cohen M. Stylizing 2.5-d video. In: *NPAR '06: Proceedings of the fourth international symposium on non-photorealistic animation and rendering*. New York, NY, USA: ACM; 2006. p. 63–9.
- [22] Lopez-Moreno J, Jimenez J, Hadap S, Reinhard E, Anjyo K, Gutierrez D. Stylized depiction of images based on depth perception. In: *NPAR '10: Proceedings of the eighth international symposium on non-photorealistic animation and rendering*. ACM; 2010.
- [23] Belhumeur PN, Kriegman DJ, Yuille AL. The bas-relief ambiguity. *Int J Comput Vision* 1999;35(1):33–44.
- [24] Loffler G. Perception of contours and shapes: low and intermediate stage mechanisms. *Vision Res*. 2008;48(20):2106–27.
- [25] Elder JH, Zucker SW. Computing contour closure. In: *Proceedings of fourth European conference on computer vision*; 1996. p. 399–412.
- [26] Ostrovsky Y, Cavanagh P, Sinha P. Perceiving illumination inconsistencies in scenes. *Perception* 2005;34:1301–14.
- [27] Langer M, Zucker S. Casting light on illumination: a computational model and dimensional analysis of sources. *Comput Vision Image Understanding* 1997;65:322–35.
- [28] O'Shea JP, Banks MS, Agrawala M. The assumed light direction for perceiving shape from shading. In: *ACM applied perception in graphics and visualization (APGV)*; 2008. p. 135–42.
- [29] Langer M, Bülthoff HH. Depth discrimination from shading under diffuse lighting. *Perception* 2000;29(6):649–60.
- [30] Koenderink J, Doorn AV, Kappers A, Todd J. Ambiguity and the mental eye in pictorial relief. *Perception* 2001;30(4):431–48.
- [31] Khan EA, Reinhard E, Fleming R, Bülthoff H. Image-based material editing. *ACM Trans Graph (SIGGRAPH 2006)* 2006;25(3):654–63.
- [32] Gutierrez D, Lopez-Moreno J, Fandos J, Seron FJ, Sanchez MP, Reinhard E. Depicting procedural caustics in single images. *ACM Trans Graph* 2008;27(5): 1–9.
- [33] Bae S, Paris S, Durand F. Two-scale tone management for photographic look. *ACM Trans Graph* 2006;25(3):637–45.
- [34] Mould D, Grant K. Stylized black and white images from photographs. In: *NPAR '08: Proceedings of the sixth international symposium on non-photorealistic animation and rendering*. New York, NY, USA: ACM; 2008. p. 49–58.
- [35] Zatzarinni R, Tal A, Shamir A. Relief analysis and extraction. *ACM Trans Graph (Proceedings of SIGGRAPH ASIA 2009)* 2009;28(5):1–9.
- [36] I.T.U., Basic parameter values for the HDTV standard for the studio and for international programme exchange. Geneva, Ch. ITU-R Recommendation BT.709, Formerly CCIR Rec. 709; 1990.
- [37] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: *Proceedings of the IEEE international conference on computer vision*; 1998. p. 836–46.
- [38] Johnston SF. Lumo: illumination for cel animation. In: *NPAR '02: Proceedings of the second international symposium on non-photorealistic animation and rendering*. New York, NY, USA: ACM; 2002. p. 45–ff.
- [39] Tatarchuk N. Shader X5. In: *Practical parallax occlusion mapping*. Charles River Media; 2006. p. 75–105.
- [40] Filion D, McNaughton R. Starcraft II: effects & techniques. In: *Tatarchuk N, editor. Advances in real-time rendering in 3D graphics and games course*; 2008.
- [41] Kajalin V. Screen-space ambient occlusion. In: *Engel W, editor. Shader X7*. Charles River Media; 2009. p. 413–24. [Chapter 6.1].
- [42] Grosch T, Ritschel T. Screen-space directional occlusion. In: *Engel W, editor. GPU Pro*. A.K. Peters; 2010. p. 215–30. [Chapter 4.2].
- [43] Todo H, Anjyo K, Baxter W, Igarashi T. Locally controllable stylized shading. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*. New York, NY, USA: ACM; 2007. p. 17.
- [44] McCann J, Pollard NS. Real-time gradient-domain painting. *ACM Trans Graph* 2008;27(3):1–7.
- [45] Kang SB. Depth painting for image-based rendering applications. U.S. Patent no. 6,417,850, granted July 9, 2002; 1998.
- [46] Sing Tsai P, Shah M. Shape from shading using linear approximation. *Image Vision Comput* 1994;12:487–98.