ARTICLE IN PRESS

Graphical Models xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Graphical Models

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

Region-based bas-relief generation from a single image

Qiong Zeng [a], Ralph R. Martin [b], Lu Wang [a], Jonathan A. Quinn [b], Yuhong Sun [c], Changhe Tu [a,*]

[a] School of Computer Science & Technology, Shandong University, Jinan, China
[b] School of Computer Science & Informatics, Cardiff University, Cardiff, United Kingdom
[c] Computer Science College, Qufu Normal University, Rizhao, China

ARTICLE INFO

Article history:
Received 7 August 2013
Accepted 1 October 2013
Available online xxxx

Keywords:
Non-photorealistic rendering
Bas-relief
Layering

ABSTRACT

Bas-relief is an art form part way between sculpture and drawing. In this paper, we present an algorithm for generating a bas-relief from a single image, inspired by the process that artists use to create reliefs. We do not aim to recover exact depth values for objects in the image, which is a tricky computer vision problem, requiring assumptions that are rarely satisfied. Instead, we determine layers based on relative depth ordering of objects (and their parts) in the image, and use this information to construct surfaces in the 3D relief model. Feature lines are extracted and used to build a new region-based representation of the input image. During surface construction, a base surface is first generated; it is then augmented using both intensity and gradient information from the original image. To prevent depth errors arising due to augmentation, a feedback process is used to refine the output. Our experimental results show the generated bas-reliefs have smooth boundaries with appropriate height relationships, a key property of bas-reliefs created by artists. We demonstrate that our algorithm works well for a range of input images, including human faces, flowers and animals.

1. Introduction

Bas-relief is a well-known form of sculpture depicting a scene using flattened objects attached to a base surface; a common use is portraiture on coinage. This special art form is often similar to a drawing in the use of obvious outlines, with flattened layers representing scene elements. The key idea is that these layers correctly indicate relative depths of objects, rather than their actual depths; depth gaps between objects in different layers are removed during flattening.

Bas-reliefs can be found on various objects, such as buildings, product packaging and coins, and can be made of a variety of materials including stone, metal, plastic and cardboard. When an artist sculpts a bas-relief on stone, a typical sequence of work is to first make an ink drawing on the stone, followed by carving the stone along the inked lines as a basis for further work. Subsequently, successive layers of material are removed in a front-to-back order to produce the scene [1]. This process relies heavily on the use of lines to delimit outlines of scene regions. Fig. 1 shows the sculptor's steps in this process.

More recently, attention has been given to digital generation of bas-relief from 3D models. However, 3D models for use in such algorithms are not readily available. Raw scanner input often needs laborious preprocessing, and 3D scanners are still uncommon (and expensive). 2D images are much more readily available, and previous work has also considered generation of bas-reliefs from images, using techniques such as shape from shading to determine a plausible 3D shape for the relief [2]; this approach is specific to frontal views of human faces. Even though image-based relief generation algorithms in principle provide much greater flexibility of choice of input scene, they are susceptible to illumination problems in the input image and may need user assistance to improve the quality of the results [3].

* Corresponding author. Tel.: +86 13791067667.
E-mail address: chtu@sdu.edu.cn (C. Tu).

1524-0703/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.gmod.2013.10.001

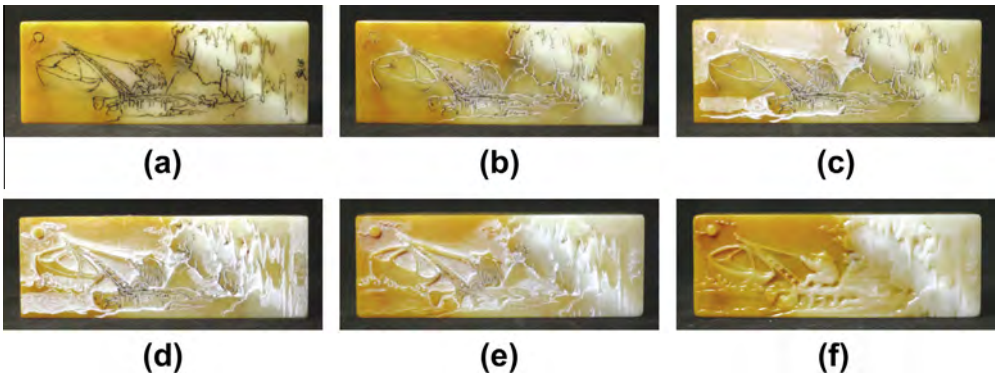


Fig. 1. How an artist makes a bas-relief: (a) drawing contours on a Shoushan stone, (b) knife cuts along the contours, (c) using chisel for layering, working from the left of the image to the right, (d) outlining the scene again to emphasize unclear parts and details, e.g. lines depicting clouds around the sun, (e) further chiseling to produce layers based on outlines in step (d), and (f) the final bas-relief. (These pictures are taken from [1].)

Our goal here is to generate a bas-relief from an input image by simulating the approach used by a sculptor. We use a region-based bas-relief construction method, which is simple and applicable to many kinds of input images, such as human faces, flowers, and animals. The results of this work can be directly used in industrial applications, or for simulating brick and stone reliefs for archaeological and historical explanation, for example. The contributions of this paper are as follows:

1. We give a simple method to divide an image into regions suitable for bas-relief generation, and show how to build up layers based on regions semi-automatically using a two-level graph. Layering is a key step in crafting hand-made bas-reliefs, and determining front-to-back relationships in an image is crucial. Our region-based layering method is also of potential use in other applications, such as stereo display of images.
2. We give a feedback-based method to refine the bas-relief using intensity and gradient information in the image. This process adds details to the base surface, generating a realistic and vivid bas-relief, while globally preserving appropriate front-to-back depth relationships.

2. Related work

2.1. Relief generation from 3D models

Various papers have considered digital bas-relief generation from 3D scenes, [4] being the earliest. The authors consider relief generation in terms of transforming depths of objects, using a perspective transformation to cause objects far from the viewer to undergo greater depth compression than nearer ones. This method fails to remove depth gaps between objects, and is relatively unsuccessful at preserving detail in the compressed result.

More recently, researchers starting with Song et al. [5] have drawn a parallel between the bas-relief generation problem and the tone mapping problem in high dynamic range (HDR) imaging, using algorithms devised for the latter to inform algorithm development. The method in [5] uses differential coordinates to represent the given 3D shape, and adapts HDR processing methods to preserve salient features while de-emphasizing others. However, their method is somewhat complicated, and the generated effects can sometimes look distorted and exaggerated. Kerber et al. [6] describe an algorithm which transforms depths in the gradient domain, achieving high compression of the depth data and preserving perceptually important features. This algorithm is simple and produces generally satisfactory bas-relief output, although again some areas can be over-emphasized. An improved version is given in [7], using a nonlinear compression function and different processing strategies based on finding fine and coarse features. Sun et al. [8] generate bas-reliefs using adaptive histogram equalization; depth compression works directly on the height field rather than gradients in this method. It can generate bas-reliefs which preserves features well, but high-quality detail preservation depends on generating a high resolution height field from the input 3D model.

Even though these methods can often produce bas-reliefs of acceptable quality, the difficulty of obtaining suitable input 3D models restricts their applicability.

2.2. Relief generation from 2D images

An alternative with wider application is to generate bas-reliefs from photographs. Unfortunately, this is an ill-posed problem: it is hard to estimate depth information from a single image. Recently, several researchers have explored recovering depth information from images specifically for the purpose of generating bas-reliefs.

2.2.1. Gradient and intensity based methods

Wang et al. [9] describe a method to construct bas-reliefs from 2D images by use of gradient operations. The authors first calculate image gradients, then attenuate gradients to smooth shape changes, while using unsharp masking to boost fine features. The modified gradient field is used to reconstruct a height image via a Poisson equation. The pixel values (depths) are then compressed, and a triangle mesh representing the bas-relief is determined by placing a vertex at each pixel position. This algorithm attempts to preserve features, but no consideration is made for preserving front-to-back relationships between different image regions.

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

Li et al. [10] present a two-level approach to construct brick and stone reliefs from rubbing images, based on separation of the relief into low and high frequency components. Low frequency components are estimated using a partial differential equation-based mesh deformation scheme, while high frequency detail is estimated directly from the rubbing image. A blended height map is produced from the base relief and high frequency details to obtain a simulated brick or stone relief. This method works well for reliefs based on brick or stone, but is unsuited to general photographs.

2.2.2. Computer vision based methods

Other work uses shape-from shading (SFS) to recover 3D shape for bas-relief generation from 2D images. SFS is a relatively mature method for constructing 3D shape from an image; see Zhang's survey [11]. The most important problem in SFS-based bas-relief generation is how to balance the relationship between illumination and bas-relief ambiguity.

Alexa and Matusik [12] describe a method to generate reliefs which reproduces diffuse reflection for an input image under known directional illumination. They adopt a discrete surface model in which each pixel is connected with several elements of the surface, allowing sufficient degrees of freedom for surface generation, thus overcoming theoretical limitations of SFS. Unfortunately, the results generated by this approach are sensitive to illumination changes, and distortions may result if the relief is viewed under different lighting, or from a different direction, than that intended.

Wu et al. [2] also use SFS to generate bas-reliefs. Their input is a photograph of a frontal human face, from which a bas-relief portrait is produced. In an offline learning phase, they learn a mapping function between an image of a 3D head model under standard lighting, and a corresponding image of a bas-relief algorithmically generated from the same 3D model (under the same lighting); this is done for two standard lighting conditions. To construct a bas-relief from a new photograph, the input image is relit to match the two standard illumination conditions, and the mapping function is used to produce two bas-relief images. A bas-relief surface is constructed from each bas-relief image using SFS, and the resulting two bas-relief surfaces are averaged to give the final bas-relief. This method is robust to changes in facial appearance and lighting direction, but results are specific to the category of human faces (or any other narrow object class learnt offline).

2.2.3. Sketch based methods

Our goal in this paper is a bas-relief generation algorithm which simulates the sculpting process and indicates relative depths of regions, based on region outlines and other feature lines. Kolomenkin et al. [13] have somewhat similar aims, although they start with a line drawing rather than a photograph or other continuous tone image. They first analyze the line drawing to understand its strokes in terms of curves, junctions, margins, and step edges. They also give an approach for adding depth to each curve, reducing this problem to a constrained topological ordering of a graph constructed from connectivity between boundaries and curves. From these computed heights of curves and step edges, they use smooth interpolation across regions without curves to generate the bas-relief surface. Their method provides flexible control over each stroke and works well for simple and coherent line drawings. However, it does not consider how to generate bas-reliefs with surface detail: their approach is limited to using information contained in a line drawing.

Much work has considered interpreting depth information from 2D line drawings based on line labeling methods [14,15]. A labeling process classifies line segments (which may be straight or curved) as *concave*, *convex* or *occluding*, constrained by valid combinations of such labels at junctions where lines meet. Such information can give clues to relative depths of regions bounded by line segments, and can even be used to build models for simple sketches of objects with planar faces. Building simple CAD-like objects is, however, far from understanding the full 3D structure of curved objects and complex scenes.

Inspired by such previous work, we propose a novel method based on the artistic sculpting process, with the aim of producing bas-reliefs from general photographs, suitable for a wide range of applications.

3. System overview

Like in drawings, in bas-relief work, lines are often used to indicate changes in the height function [13]. Lines help to clarify distinctions between different objects, or parts of objects, and help to convey their front-to-back relationships. Sculptors working in stone typically create bas-reliefs following two main steps, firstly carving outlines, and subsequently adding details by chiseling away different layers [1]. Similar techniques are used for other media. As the layers are created, the sculptor focuses on depicting which objects or parts are in front of others, rather than absolute height values for each region. We propose an approach for generating a bas-relief from an image based on these observations of real sculpting practice.

Thus, we explore the 3D geometric information implicit in the input image to help generate the layering structure of the final bas-relief, and subsequently add appearance detail to these layers. The main goals are to represent proper front-to-back relationships, to emphasize salient features, and to suppress extraneous detail. To meet these goals, we use a two-step region-based bas-relief generation algorithm, in which region-based layer determination is performed first, followed by relief construction. Fig. 2 shows the framework of our algorithm.

During region-based layer determination, we first extract a 2D feature line image (see Section 4.1). Next, seed points are derived from these lines, and regions are found from them using a region growing process. Line segments and junctions are then found based on regions (see Section 4.2). We define concepts of *connected* and *ambiguous* to help explain relationships between regions. This leads to a scene representation based on regions, segments, junctions and relationships between them, using a two-level undirected graph. We turn it into a *directed* graph based on a simplified line labeling method, and determine

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

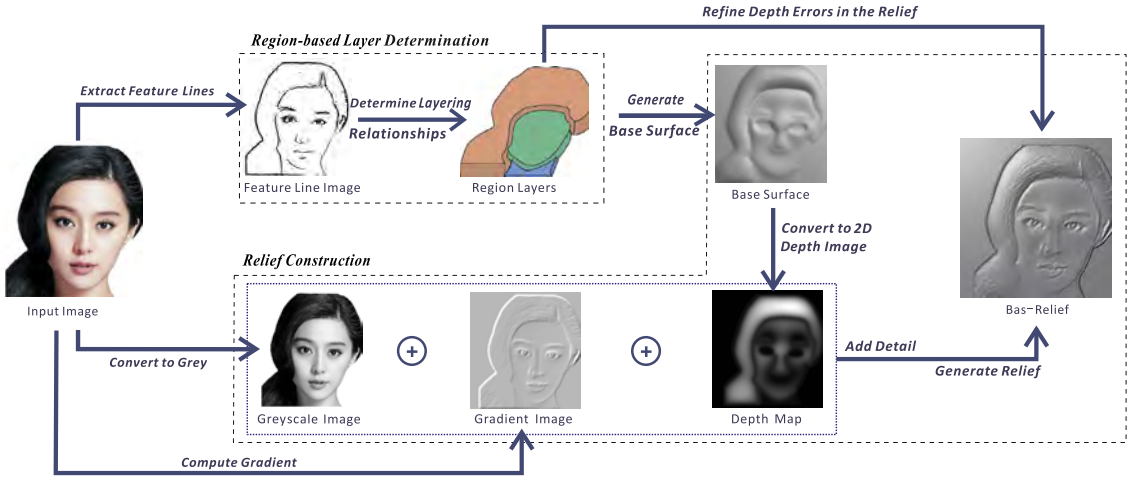


Fig. 2. Framework, showing region-based layer determination and relief construction. In layer determination, a line drawing is extracted and used to decide region front-to-back relationships. Relief construction initially builds a base surface, then adds detail using intensity and gradient information. A refinement process based on determined region layers helps optimize the result.

relative depths of regions using topological sorting. Finally, we deduce height values for each segment according to region depths and line labels (see Section 4.3).

In the relief construction step, we use the height values of each segment as constraints to build a base surface, taking into account the determined depth relationships between regions (see Section 5.1). We add detail by refining the generated base surface using gradient and greyscale information from the initial image. To avoid relative depth errors caused by adding detail, a refinement process is used to ensure correct front-to-back relationships between regions (see Section 5.2).

4. Region-based layer determination

In this step, the goal is to determine regions, and the correct front-to-back depth ordering between adjacent regions in the input image. At the same time, we replace the pixel representation by a two-level graph based representation linking regions bounded by smooth curves.

We first extract a simple 2D feature line image from the input image, and use it to find regions, segments and junctions. We then build a two-level graph recording adjacency relationships between regions, and direct it as we infer the relative heights across each line segment separating two regions.

4.1. Feature line extraction

A 2D image contains implicit information about the corresponding 3D scene, through regions, their boundaries, shading, and textures. Much useful scene information is presented by boundaries. We thus extract simple and coherent feature lines to help understand the scene.

Several line extraction algorithms exist which provide comparatively smooth results. The approach in [17] is one of the best examples of its type, and can generate coherent and smooth lines from an image. However, it also produces short lines caused by shading rather than geometry, which are undesirable for our work. We thus modify it. As small details in an image disappear at low resolution, we use a pyramid-based improvement scheme to eliminate such short lines.

The input image typically contains more detail than is appropriate for making a bas-relief, so we use L_0 smoothing [16] to remove unwanted detail. Next, we downsample the smoothed image to generate an image pyramid $P = (P_0, P_1, \ldots, P_n - 1), n = \lfloor \log m \rfloor + 1$ and $m = \min(I_W, I_H)$, where I_W, I_H are the width and height of the input image. P_0 is the coarsest image. For each level P_i in P, the method in [17] is used to generate a corresponding line drawing image, giving a line drawing pyramid $L = (L_0, L_1, \ldots, L_{n-1})$; these are binary images. To obtain feature lines in the image, we iteratively process these images as follows, starting from the coarsest level. Image L_0 is upsampled to the same size as image L_1, then we compute the intersection of this upsampled image with L_1 to generate a new image, called L_1'. This new image is upsampled in turn and combined with L_2 to obtain L_2'. Upsampling is performed by simply duplicating pixel values in the coarse image which cover the same location as subpixels in the finer image. The final image L_{n-1}', the *feature line image* (still in pixel form), provides feature lines with few short lines, as required. The process is shown in Fig. 3.

4.2. Regions, segments and junctions

To build a bridge between the 2D image and the 3D bas-relief, we next convert the feature line image to a representation composed of 2D geometric elements: regions, segments and junctions.

Regions represent coherent parts of the scene, and are used to determine relative depth ordering. The most challenging problem here is that our feature line image does not immediately represent regions with completely closed boundaries (see Fig. 4(a)). Finding regions using e.g. the flood fill algorithm will lead to leakage. Xie et al. [18] overcome this problem using active contours. We

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

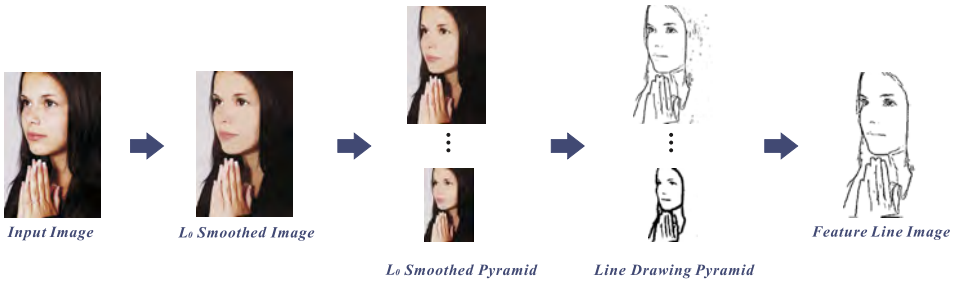


Fig. 3. Extracting a line drawing. Left to right: input image, L_0 smoothed image generated by [16], L_0 smoothed image pyramid, line drawing pyramid extracted using the method in [17], feature line image. This approach helps avoid short edges due to shading effects in our feature line image, aiding scene understanding.

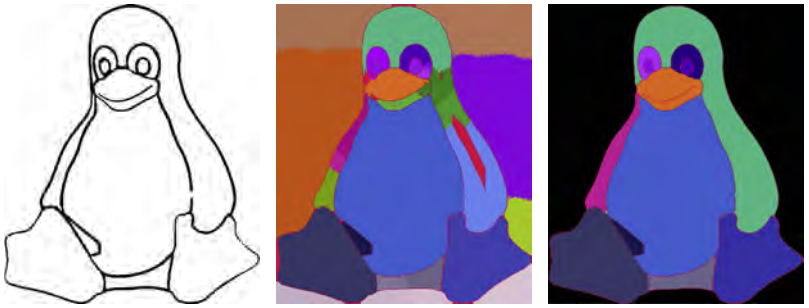


Fig. 4. Region determination process: (a) input 2D feature line image, (b) regions found by the method in [18], with red boundary pixels, (c) merged regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

adopt their method and improve it for recognizing regions, as follows.

Given the feature line image, we firstly thin the lines to single-pixel-wide lines using the algorithm in [19] to give the skeleton pixels for each line. Then, pixels are found which are a local maximum of the unsigned distance field measuring distance from the nearest skeleton pixel. Starting from this set of locally maximal points, a region growing algorithm is then used to separate the initial image into regions: the seed points are placed into a heap, sorted according to their distance field values. The top of the heap is grown first; growth stops when a skeleton point, or a pixel already belonging to a previously grown seed, is encountered.

This growth process leads to an initial set of regions separated by skeleton pixels (see Fig. 4(b)), but usually the result has too many regions. To reduce them to a more useful and natural number, we merge regions with multiple touching pixels not separated by skeleton pixels: for each point, if an n–n window around it contains different regions, but no skeleton points, the regions are merged. Note that leakage can occur if the window size n is smaller than the length of gaps in feature lines. Fig. 4(c) shows a result of merging regions in the background and foreground. We denote this set of regions $R = (R_0, R_1, \ldots, R_{k-1})$, where k is the number of regions; it is equal to the number of loops in the feature line image.

Having obtained regions, segments and junctions are easily determined: segments are skeleton lines between different regions, while junctions are places where segments meet. We denote the segments and junctions respectively by $S = (S_0, S_1, \ldots, S_{l-1})$ and $J = (J_0, J_1, \ldots, J_{m-1})$, where l and m are the numbers of segments and junctions. Our region-based representation is illustrated in Fig. 5(a)): black circles label regions, black dots indicate junctions and colored lines meeting at junctions are segments.

4.3. Region graphs and relative depths

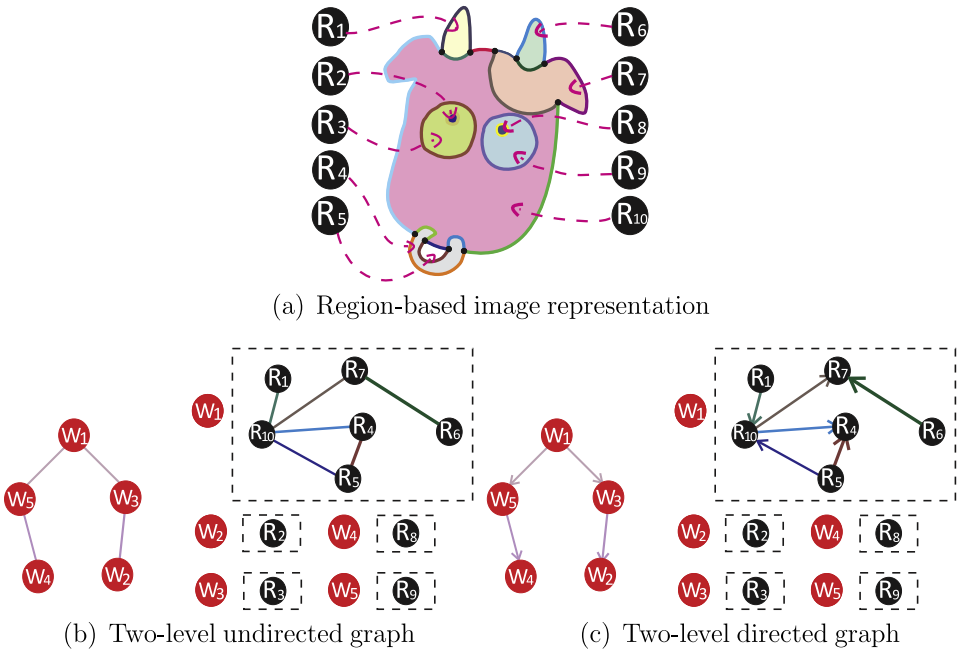
Regions in an image may have different connectivity relationships, including *containing*, *neighboring* and *disjoint*. Connectivity between regions, segments and junctions inspires us to map the region layering determination problem to a graph problem, in which regions are nodes, and edges link regions adjoining a common segment. The layering relationship is represented by a directed graph, in which graph edge direction indicates relative height.

To take into account different possible relationships between regions, we use a two-level graph and a simplified line labeling method to deduce directions of each edge. The resulting directed two-level graph gives front-to-back layering relationships between regions needed for bas-relief generation.

4.3.1. Region relationships and two-level graph

It would lead to excessive complication to try to represent all region connectivity relationships in detail. Instead, we analyze relationships between regions based on

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001



(a) Region-based image representation

(b) Two-level undirected graph

(c) Two-level directed graph

Fig. 5. Region-based layer determination: (a) the feature line image is represented by regions (black circles indicate regions represented as colored patches), region boundary segments (colored lines surrounding patches), and their junctions (black dots). Region relationships are either *connected* or *ambiguous*, determined by the connectivity of segments and junctions; these are used to determine an undirected two-level graph (b). The first level in (b) comprises ambiguous region groups (red circles); the second level comprises connected region groups (black circles). On the first level, if an ambiguous region group is directly contained within another ambiguous region group, an edge is placed between them. On the second level, an edge is placed between two regions if they share a common boundary segment. A directed two-level graph (c) is generated semi-automatically; the region at the head of each arrow is higher than the one at its tail. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

segments and junctions, and build our two-level graph based on two particular relationships between regions, *connected* and *ambiguous*.

A *connected* relationship exists between regions joined to each other by segments having junctions. For example, R_1, R_4, R_5, R_6, R_7 and R_{10} in Fig. 5(a) are connected regions.

Regions may also be separated from all other adjacent regions by loops without any junctions. Such regions have an *ambiguous* relationship: see R_3 and R_{10} in Fig. 5(a).

The rationale for this distinction is that junctions help us to determine relative depths when processing connected regions. Lack of junctions for ambiguous regions means that we cannot infer relative depths without user assistance.

The definition above leads to a transitive property between regions. If a region is connected to some regions and has ambiguous relationships with other regions, then regions to which it is connected also have ambiguous relationships with its ambiguous regions. For example, in Fig. 5(a), R_3 and R_{10} are ambiguous regions and at the same time R_{10} is connected to R_1, R_4, R_5, R_6 and R_7. According to the transitive property, R_1, R_4, R_5, R_6 and R_7 are ambiguous regions with respect to R_3.

Based on the transitive property, regions can be separated into ambiguous region groups—each group comprises several connected regions, which leads to the construction of a two-level graph $G = (W, E)$ and $W = (R, F)$. The nodes of the first level graph G are ambiguous region groups W, while E depicts the edge set of the first level graph. If an ambiguous region group lies inside another ambiguous region group, we join the corresponding two nodes by an edge. Each ambiguous region group consists of several connected regions $W = (R, F)$. Regions connected to each other are nodes of the second level graph $W = (R, F)$; an edge joins two nodes if a segment lies between the corresponding two regions. F is the edge set of the second level graph. We show a two-level undirected graph in Fig. 5(b).

4.3.2. Relative depth determination

So far, the two-level graph indicates connectivity (or adjacency) relationships between regions. We now attempt to direct edges in the graph (the direction proceeding from the lower region to the higher region).

Edges in the two-level graph are defined whenever regions are separated by segments. We can achieve our goal of directing the graph by labeling directions of each segment between regions, using the usual line-labeling convention that an arrow on a segment indicates the region on the left occludes the region on the right. (We do not use *convex* or *concave* labels). We use heuristics and user assistance to label segments semi-automatically:

Labeling Rules

1. The background is lower than the regions touching it. This heuristic lets us label segments touching background regions automatically. In Fig. 6 for example, the red segment and yellow segment are labeled: the

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

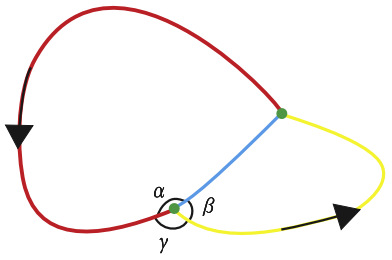


Fig. 6. Occluding segment labeling. The red, yellow and blue lines are segments of the line drawing, while green points are junctions of the line drawing. Black arrows are occluding labels according to Labeling Rules, indicating occluding regions to their left. α, β and γ are angles between each pair of segments, which we use as clues for labeling the blue segment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

left side of each arrow is higher than the right side (background).

2. The relationship between regions at the top graph level is inherently ambiguous. We cannot tell whether a completely contained region is higher or lower than the surrounding region from a single image, so in such cases we ask the user to explicitly indicate the relative height, allowing us to label those segments connecting different ambiguous regions.
3. In connected regions, two situations can arise according to junction types: *three segments going into a junction* and *more than three segments going into a junction*. The latter again requires user assistance. In the former case, we repeatedly use the following approach until all segments have been labeled:
   Find angles between each pair of segments, called α, β, γ in Fig. 6. Choose the angle nearest to 180°. The two segments forming this angle are nearest to lying on a straight line, so they should have the same occlusion label.
   (a) If one of two segments is already labeled, then label the other.
   (b) If both are unlabeled, we ignore them for now and consider other junctions.
   (c) If both are labeled the same, this adds no new information, and we proceed.
   (d) If both are labeled differently, it is likely that some inconsistency has occurred, so we ask the user for assistance to decide the correct label.

The above approach allows us to label segments between adjacent regions, providing depth relationships for the directed two-level graph. Topological sorting is then used to determine front-to-back depth relationships between regions. A depth for the lowest region and a depth differences for regions are chosen by the user, allowing us to calculate a height value of each region and then deduce the height h_i for each segment i. Certain feature lines do not bound regions (see Fig. 4(a)), and we require the user to assign them an explicit height and a segment label indicating which side is higher.

5. Relief construction

As noted, artists typically initially use outlines to construct layers, and finally add detail. Our relief construction step is modeled on this process. We first generate a low-frequency base surface using the geometric information determined by region processing. We then refine the base surface using further 2D image information, using a feedback mechanism to ensure relative depths of layers remain satisfactory.

5.1. Base surface generation

A relief can be defined as an area of a surface with sculpted features different from those of the underlying surface [20]. The sculpted features often change strongly near contours of the relief and are smooth far from contours. In [13], Kolomenkin et al. proposed a simple method to construct a relief surface from curves. It smoothly interpolates heights in regions without curves, applying given height constraints in margins (narrow strips on either side of curves). Their method can handle multiple lines, and uses a simple process to generate a relief surface. We adopt their method to generate our base surface.

We use an offset function to represent the base surface, which describes sculpted features above a background plane of height zero. The offset function has a large gradient in the margins, orthogonal to segments in the feature line image, and has a smaller and smoother gradient in areas between segments. Note that the height of each segment in the feature line image has been determined previously.

5.1.1. Sampling and triangulation

The final 3D base surface is represented as a triangle mesh. We firstly place sample points in the feature line image according to the desired gradient and from them construct a triangular mesh; heights of the vertices are then adjusted to give the final base surface.

In detail, to perform sampling of the image, B-splines are fitted to segments in the 2D feature line image. These B-spline curves are sampled with regular spacing, and used as some of the mesh vertices, to ensure a high quality triangulation which conforms to the feature lines. Further mesh vertices are generated in 2D using the density-controlled low-discrepancy sampling method described by Quinn et al. [21]. We define the density function over the domain as the normalized pixel addition of the extracted feature line image combined with a Gaussian smoothing of the greyscale input image. This density function has smooth gradients and well defined boundaries; use of the feature line image ensures that a high sampling density is achieved in margins, where the height gradient is large. The generated sample points are then connected via Delaunay triangulation [22], giving a 2D mesh.

5.1.2. Computation of local offset functions

In the margins, we compute the offset function $z(p(x, y))$ locally, where $p = (x, y)$ is the position of a sample point. In

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

other areas, the offset function should be the smoothest function that is in agreement with the offset function in the margins.

Height changes in the relief should be strongest orthogonal to segments in the 2D feature line image. We use a smoothed step edge function to approximate relief shape changes and compute the offset value of each sampling point in margins:

$$step(d) = h_i(1 + \tanh(d/w_i)), i = (0, 1, \ldots, l-1) \quad (1)$$

where h_i is the height of a segment, w_i is the width of the margin of segment i, l is the number of segments and d is a signed distance to the segment, computed using a method from [23]. Note that alternative step functions can be used to produce different effects. Fig. 7 shows cross sections of two step edge functions and their corresponding side views of local relief shape.

5.1.3. Computation of global offset functions

The relief surface in areas outside margins should be a smooth offset function compatible with the local offset function in margins of bounding segments. We determine heights at sample points in areas outside margins using an interpolation method based on the determined local offset function. Two stages are used to compute the global offset function. The first stage uses a smoothness requirement to determine the desired Laplacian $L(p)$ of the offset function $z(p)$. We then use the Laplacian value $L(p)$ to calculate $z(p)$ in the second stage.

We require that the Laplacian of the relief is zero in order to satisfy the smoothness requirement, which is approximately equivalent to the requirement that the mean curvature of the surface changes linearly [13]. Let Δ denote the Laplacian operator. L should be equal to the Laplacian of z of sample points in margins and its Laplacian $\Delta(L)$ should be zero elsewhere, i.e.:

$$\begin{cases} L(p) = \Delta z(p) & p \in \text{ margins}, \\ \Delta L(p) = 0 & \text{elsewhere}. \end{cases} \quad (2)$$

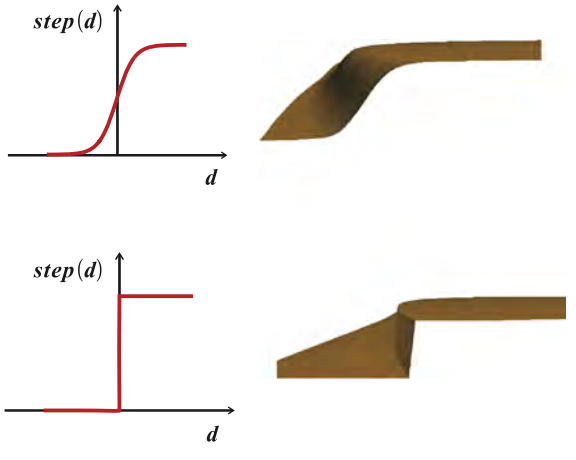


Fig. 7. Different step functions. Cross sections of two step functions and their corresponding relief surfaces in the margin (side view).

The Laplacian Δ of a scalar function f (here it may be z or L) at point p on a mesh is calculated using [24]:

$$\Delta f(p) = \frac{1}{2A} \sum_{j \in N(p)} (\cot(\gamma_i) + \cot(\delta_i))(f(p) - f(p_j)) \quad (3)$$

where $N(p)$ is the one ring neighbors of point p, A is the area of the Voronoi cell of p, and γ_i and δ_i are angles opposite the edge $[p, p_j]$of the triangles adjacent to this edge [13].

We solve this linear system using the stabilized biconjugate gradient method [25].

Given the Laplacian of the offset function z, we then calculate the offset function using the following system of linear equations:

$$\begin{cases} z(p) = z(p) & p \in \text{ margins}, \\ \Delta z(p) = L(p) & \text{elsewhere}. \end{cases} \quad (4)$$

In this equation system, the unknowns are the values of z at sample points outside margins; it is again solved using the stabilized biconjugate gradient method.

5.2. Image-based 3D relief refinement

The generated base surface provides correct basic front-to-back region relationships, but does not yet show any detailed texture. We use the initial input image to provide such detail—intensity provides texture, and gradient is used to emphasize changes. The relief surface is refined using this information.

We first convert the 3D base surface to a 2D depth map which we then refine by combining it with gradient information and greyscale information extracted from the input image. This may lead to relative depth errors, so we use a feedback process to preserve correct front-to-back relationships between regions determined in Section 4.

5.2.1. Combination refinement

We denote the greyscale image, gradient image, and generated depth map respectively by $I(x, y)$, $G(x, y)$, $D(x, y)$, and the new relief depth by $R(x, y)$, where (x, y) is the position of each pixel. We compute

$$R(x, y) = \delta I(x, y) + \lambda G(x, y) + \mu D(x, y) \quad (5)$$

where δ, λ, μ are parameters of $I(x, y)$, $G(x, y)$ and $D(x, y)$. Adjusting these parameters controls the modulating effects of the additional information.

5.2.2. Front-and-back relationship refinement

Combining depths with greyscale and gradient information will not preserve correct height relationships. Hence, the generated relief $R(x, y)$ must be adjusted to satisfy the front-to-back relationships determined in Section 4. Our goal thus is to adjust the current height relationships between regions, measured by the average height value $A[i]$ of pixels in region i, to ensure the desired relationships.

It would be complicated to refine all unsatisfactory regions at the same time, so we constrain the adjustment process in two directions, from above and below.

Initially, for a region i, we define its *lowest above region* `lowest_above[i]` and *highest below region*

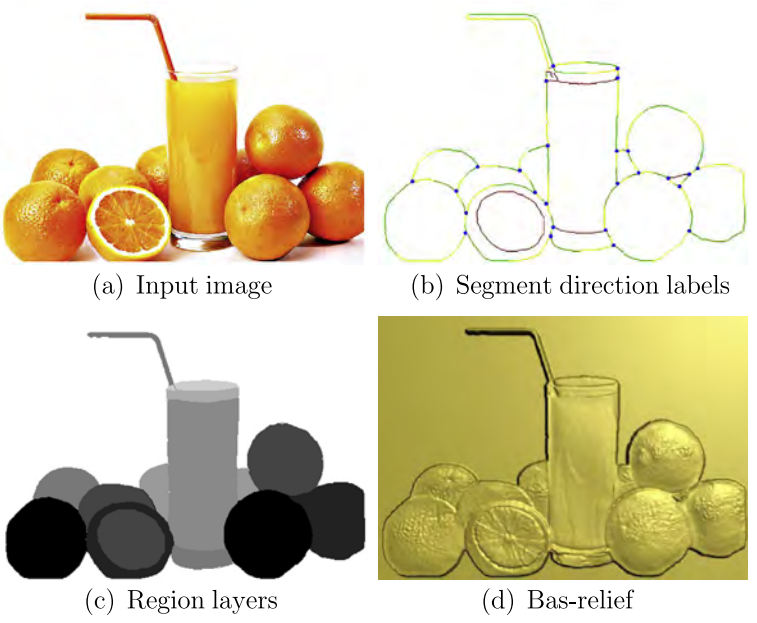


Fig. 8. Bas-relief of a fruit image with complicated region layering relationships. (a) Input image. (b) Segment direction labels: colored segments have directions from green to yellow, labeled by the rules in Section 4.3.2; red segments were labeled by the user; blue points are junctions. Relative depths are shown in (c): the blacker the region, the more forward the corresponding layer. (d) Corresponding bas-relief. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

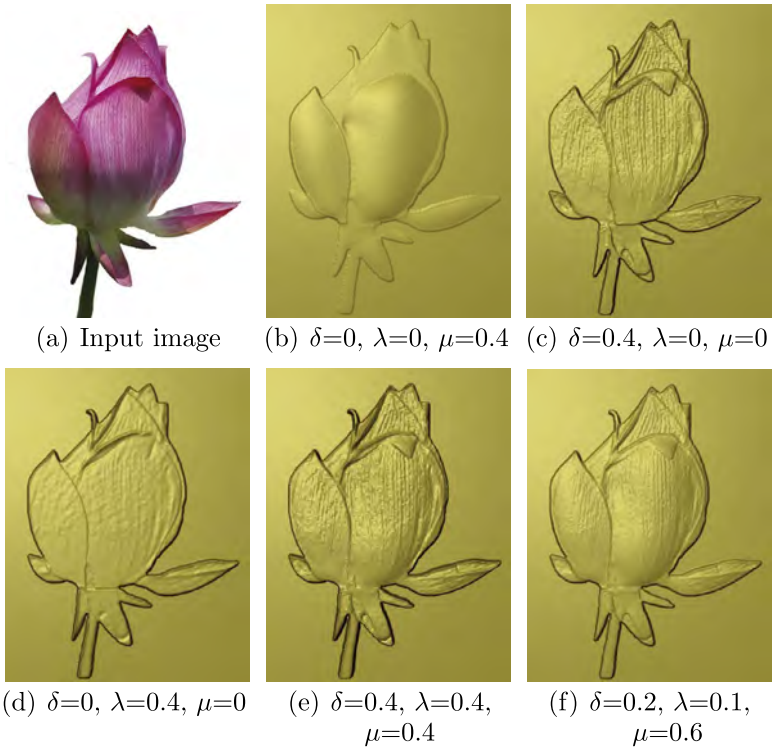
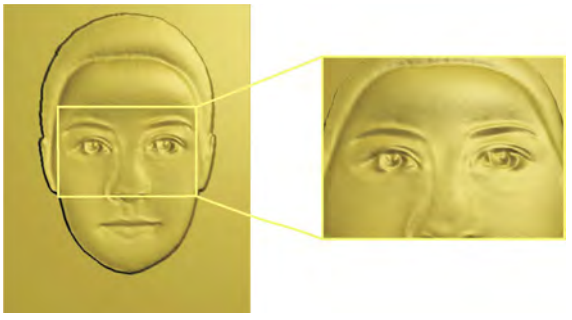


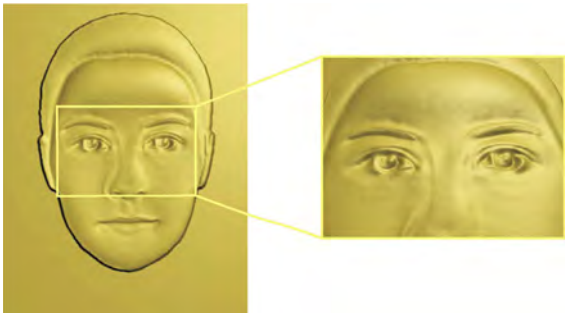
Fig. 9. Bas-reliefs generated using different base and detail parameters δ, λ and μ: (a) Input image. (b–e) Bas-reliefs using different parameters. Increasing μ strengthens the influence of the base surface. Increasing δ enhances details. Increasing λ strengthens gradient effects near boundaries.

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

(a) Input image



(b) Relief without front-to-back relationship refinement



(c) Relief with front-to-back relationship refinement

Fig. 10. Bas-reliefs generated by our algorithm from a picture of a face, with and without front-to-back relationship refinement. The eyebrows of the bas-relief are partly sunken without refinement (b), but are raised after refinement (c).

`highest_ below[i]`. The lowest above region for region *i* is that region with smallest average height amongst those regions which are required to be higher than region *i*; highest below region is defined analogously.

We use the lowest above region and highest below region to compare relative heights of regions. If `highest_below[i]< A[i]< lowest_above[i]`, region *i* has a correct relative height. Otherwise, we perform higher region adjustment and/or lower region adjustment as needed. Firstly, we adjust regions which should be higher than the current region. If a given region *i* satisfies `lowest_above[i]< A[i]`, regions exist that should be higher than it which have smaller average values, so we adjust those regions. Lower region adjustment is done in a similar way. We perform these steps for all regions and iterate the whole process until all regions are in correct positions.

6. Results and discussion

We now present various experimental results generated by our algorithm, which was implemented in C++ on a PC with a 2.33 GHz Intel Core Duo CPU and 2 GB memory. It takes about 10 min to produce a bas-relief based on 10,000 sample points (as in Fig. 10) and about 13 min for a bas-relief with 15,000 sample points (see Fig. 12).

Fig. 8 shows the bas-relief generated for an image with complicated layering relationships, and also shows intermediate labeling and layering results. Fig. 8(b) shows segment direction labels determined by our labeling rules. Colored segments were directed automatically, with directions from green to yellow points; red ones did not meet the requirements for automatic labeling and so were labeled by the user. Blue points mark junctions. Fig. 8(c) shows region layers determined by the constructed two-level graph, in which nearer layers are presented by blacker regions. Fig. 8(d) shows the generated bas-relief surface, which has suitable, correct front-to-back relationships, smooth boundaries and salient features.

Fig. 9 illustrates the variation in results which can be generated by modifying the parameters used to combine the base surface with details derived from the image intensity and gradient. Fig. 9(b–d) illustrates how the base surface, gradient image and greyscale image respectively control relative depths, boundary effects and details. If we combine the three images with equal weights, details are overemphasized (see Fig. 9(e)); and Fig. 9(f) shows a more satisfactory result, with both adequate detail and clear depth layering relationships. The depth map should generally be given the greatest weight, to help preserve layer relationships.

Fig. 10 shows a generated bas-relief surface, and compares it to a bas-relief generated without the front-to-back relationship refinement process. The result without refinement is somewhat sunken around the eyebrows (see Fig. 10(b)). With refinement, the bas-relief displays more satisfactory height relationships (see Fig. 10(c)).

Fig. 11 shows results generated by Li et al. [10] using a two-level method and our approach, for a male actor and a cartoon penguin. The male actor image in Fig. 11(a) is highly detailed and has complicated shading around the neck. Our method generates a bas-relief with a smoother and more natural surface, which also retains salient detail as well as overall features of the input image; too much detail is present in the bas-relief generated by Li's method, e.g. in the beard. Fig. 11(b) shows bas-reliefs of a simple cartoon penguin. Our approach generates a smoothly curved surface with appropriate layering relationships; the bas-relief generated by Li's method has flat regions. However, our algorithm does less well when applied to complicate shadings: the bottom relief in Fig. 11(a) is noisy because of the highly detailed nature and complicate shading of the initial image.

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

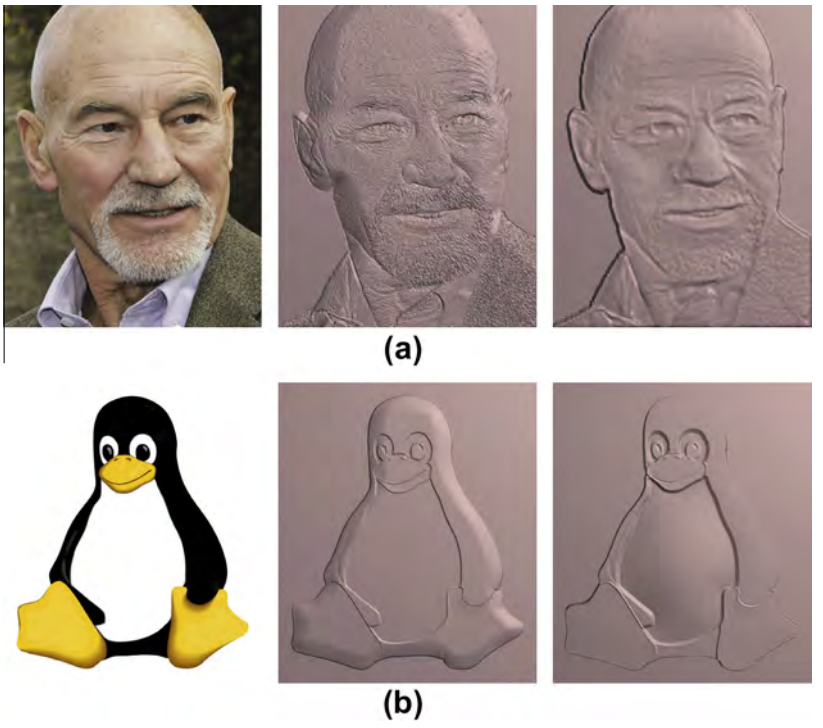


Fig. 11. Bas-reliefs of a male actor and a cartoon penguin (left column) generated by the method of Li et al. [10] (middle column) and our method (right column).

Fig. 12. Bas-relief of a woman face (left) generated by the method of Wu et al. [2] (middle) and our method (right).

Table 1
User interaction for Figs. 8–12.

	Total number of segments	Number of segments labeled by the user	Interaction ratio (%)
Fig. 8	46	6	13
Fig. 9	41	1	2
Fig. 10	19	8	42
Fig. 11(a)	51	15	29
Fig. 11(b)	34	5	15
Fig. 12	21	9	43

Fig. 12 compares our approach with a bas-relief generated by Wu et al. [2] using shape-from-shading; that method relies on learning information specific to frontal faces. Our result is generally smoother, with less noise, resulting in clearer, more recognizable features. It also has a more rounded appearance. However, even though we have suitable height relationships between regions, e.g. the neck and face, depth errors occur in certain details, caused by the addition of intensity and gradient information, such as a slightly sunken line on the nose. Refinement has not completely overcome such problems. If our approach is to be used for domain specific applications, a better prior could be used to help construct the base surface. For example, for reliefs of human faces, face recognition software could be used to find the face in the image, to ensure that the nose region is appropriately raised in the relief.

In our algorithm, user interaction is required in certain labeling situations, as noted earlier. User interaction is also always required to assign a height value for the lowest region and a depth gap value, which we ignored in our further discussion. We may assess the degree of user

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001

interaction required by measuring the interaction ratio, the number of segments labeled by the user compared to the total number of segments in the feature line drawing.

Table 1 summarizes the user interaction needed for Figs. 8–12. The amount of user interaction required is related to the geometric complexity of the input image. Our heuristic labeling method works well for connected region groups, but less well for ambiguous region groups. For images with several separating region groups, e.g. Figs. 10, 11(a) and 12, more user interaction is required. Fig. 9 has the lowest interactive ratio, as its segments are well connected. Even though Fig. 8 has complicated layering relationships, the interaction ratio is still quite low: the method does not require much work from the user.

7. Conclusion

We have described an approach to generating bas-reliefs from a single image, based on region extraction and depth relationship inference; our approach attempts to follow the steps used in hand-crafting bas-reliefs—outlining and chiseling layers. Our method can construct bas-relief surfaces for a range of objects, but works best for relatively simple input images. Experiments show that our method is generally capable of inferring correct front-to-back depth relationships between regions, with limited user-assistance, leading to a bas-relief with correct global shape. More work is needed to improve this approach for images with many details, and more objective evaluation criteria for generated bas-reliefs would be helpful.

Acknowledgments

We are grateful to the anonymous reviewers for helpful suggestions. We are also grateful to Jing Wu and Zhuwen Li for providing us with output bas-reliefs produced by their methods. This work was supported by NSFC (61272242, 61003149 and 61332015), NSFC-Shandong (ZR2010FQ011), NSFC-Guangdong Joint Fund Project (U1035004), National High-Tech Program Project (2012AA01A306) and the Higher Education Funding Council for Wales (RIVIC).

References

[1] S. Zheng, The Graphical Techniques of Shoushan Stone Carving, Fujian Art Publishing House, 2010.
[2] J. Wu, R. Martin, P. Rosin, X. Sun, L. F., Y. Lai, A. Marshall, Y. Liu, Making bas-reliefs from photographs of human faces, Computer Aided Design 45 (3) (2013) 671–682.
[3] J. Kerber, M. Wang, J. Chang, J.J. Zhang, A. Belyaev, H.-P. Seidel, Computer assisted relief generation – a survey, Computer Graphics Forum 31 (8) (2012) 2363–2377.
[4] P. Cignoni, C. Montani, R. Scopigno, Computer-assisted generation of bas- and high-reliefs, Journal of Graphics Tools 2 (3) (1997) 15–28.
[5] W. Song, A. Belyaev, H. Seidel, Automatic generation of bas-reliefs from 3D shapes, in: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007, 2007, pp. 211–214.
[6] J. Kerber, A. Belyaev, H. Seidel, Feature preserving depth compression of range images, in: Proceedings of the 23rd Spring Conference on Computer Graphics, SCCG 2007, 2007, pp. 110–114.
[7] J. Kerber, A. Tevs, A. Belyaev, R. Zayer, H. Seidel, Feature sensitive bas relief generation, in: 2009 IEEE International Conference on Shape Modeling and Applications, 2009, pp. 148–154.
[8] X. Sun, P. Rosin, R. Martin, F. Langbein, Bas-relief generation using adaptive histogram equalization, IEEE Transactions on Visualization and Computer Graphics 15 (4) (2009) 642–653.
[9] M. Wang, J. Chang, J. Pan, J. Zhang, Image-based bas-relief generation with gradient operation, in: The Eleventh IASTED International Conference on Computer Graphics and Imaging.
[10] Z. Li, S. Wang, J. Yu, K. Ma, Restoration of brick and stone relief from single rubbing images, IEEE Transactions on Visualization and Computer Graphics 18 (2) (2012) 177–187.
[11] R. Zhang, P. Tsai, J. Cryer, S. M, Shape from shading: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 690–706.
[12] M. Alexa, W. Matusik, Reliefs as images, ACM Transactions on Graphics 29 (4) (2010) 60:1–60:7.
[13] M. Kolomenkin, G. Leifman, I. Shimshoni, A. Tal, Reconstruction of relief objects from line drawings, Computer Vision and Pattern Recognition (CVPR) (2011) 993–1000.
[14] J. Malik, Interpreting line drawings of curved objects, Journal of Computer Vision 1 (1987) 73–103.
[15] P. Varley, R. Martin, Estimating depth from line drawings, in: Proc. 7th ACM Symposium on Solid Modeling and Applications, SM02, 2002, pp. 181–191.
[16] L. Xu, C. Lu, Y. Xu, J. Jia, Image smoothing via L_0 gradient minimization, in: Proceedings of the 2011 SIGGRAPH Asia Conference, 2011, pp. 174:1–174:12.
[17] H. Kang, S. Lee, C.K. Chui, Coherent line drawing, in: ACM Symposium on Non-Photorealistic Animation and Rendering (NPAR), 2007, pp. 43–50.
[18] H. Xie, K. McDonnell, H. Qin, Surface reconstruction of noisy and defective data sets, in: Proceedings of the Conference on Visualization '04, 2004, pp. 259–266.
[19] R.C. Gonzalez, R.E. Woods, Digital Image Processing Second Edition, Publishing House of Electronics Industry, 2008.
[20] S. Liu, R. Martin, F. Langbein, P. Rosin, Background surface estimation for reverse engineering of reliefs, International Journal of CAD/CAM 7(4) (2007).
[21] J.A. Quinn, F.C. Langbein, R.R. Martin, G. Elber, Density-controlled sampling of parametric surfaces using adaptive space-filling curves, in: Geometric Modeling and Processing – GMP 2006, Lecture Notes in Computer Science, vol. 4077, 2006, pp. 465–484.
[22] J. Wang, W. Wang, C. Tu, C. Yang, Computational Geometry and its Application, Science Press, 2011.
[23] W. Wang, H. Pottmann, Y. Liu, Fitting b-spline curves to point clouds by squared distance minimization, ACM Transactions on Graphics 25 (2) (2006) 214–238.
[24] M. Meyer, M. Desbrun, P. Schröder, A. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, Visualization and Mathematics III (2002) 34–57.
[25] H.A. van der Vorst, Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (2) (1992) 631–644.

Please cite this article in press as: Q. Zeng et al., Region-based bas-relief generation from a single image, Graph. Models (2013), http://dx.doi.org/10.1016/j.gmod.2013.10.001