Pacific Graphics (2014)
J. Keyser, Y. J. Kim, and P. Wonka (Editors)

A Simple Artistic Rendering Method for Stereoscopic Images

Dajin Li[†1] and Chengjie Bai[2]

[1]Digital media Department, Shandong Normal University of China
[2]Physics & electronics science school, Shandong Normal University of China

Abstract

General painterly rendering algorithms cannot be used to render stereo image pair separately because they may produce random noise and lead to inconsistency between binocular images. In this paper, a generic rendering framework of artistic stereoscopic images for virtual 3D scenes is presented. Firstly, an artistic image that contains entire visible surfaces of a 3D model from two eyes was rendered. During the projection transformation, the texture coordinates which correspond to vertices of the model were recorded. Then, the artistic image is mapped to object surface so as to obtain the stylized model. Finally, the stylized 3D model is projected to the left and right cameras for artistic stereoscopic images. The proposed rendering framework can guarantee the consistency between binocular images; moreover, it also can be applied to various existing artistic rendering algorithms.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithm

1. Introduction

The combination of conventional painting with stereoscopic display technology breaks through the 2D expression of conventional painting by enhancing the depth perception and providing an improved visual experience to viewers. Like other artwork, artistic stereoscopic image can be considered as an independent and special visual art. Furthermore, it's rendering technology is an interesting and promising research field.

The stereoscopic display requires two images should be kept consistent in the corresponding positions of object [KT04], otherwise the images may fail to be fuses. Painting the binocular images by hand in traditional way is a trivial and time-consuming task. The study of computer graphics on artistic rending makes it possible to automatically generate the artistic stereoscopic images. However, the existing artistic rendering algorithms have severe randomness and uncertainty in simulating various artistic strokes and paint effects. Therefore, adoption of existing artistic rendering algorithms to generate binocular images separately will lead to inconsistency among the image pair.

To solve these problems, a rendering method of artistic stereoscopic images for 3D scenes is presented in this paper. The goal of this study is to provide an universal stereoscopic image rendering framework that can be applied to various artistic rendering algorithms. In our method, a stylized texture is mapped onto object surface and dual cameras are used to capture binocular images. So the consistency between the stereo image pair can be guaranteed. The contributions of this work are summarized as:

- An artistic stereoscopic image generation framework independent of specific artistic rendering algorithm is developed.
- The artistic image used for texture mapping should contain all visible information in binocular images. We give a simple and efficient method to capture two views' visible areas in a single image.

2. Related work

Automatic artistic rendering algorithms: In general, the automatic artistic rendering algorithms can be classified into two classes: a model-based method that uses geometric information of 3D model and illumination information to control color or strokes rendering([GGSC98] [KNBH12]). The other is image style conversion ([ZZXZ09] [Her98]) that realizes the style conversion by adding artistic strokes based on an existing image. Many algorithms have been

† email:ldjwqc@163.com

DOI: 10.2312/pgs.20141267

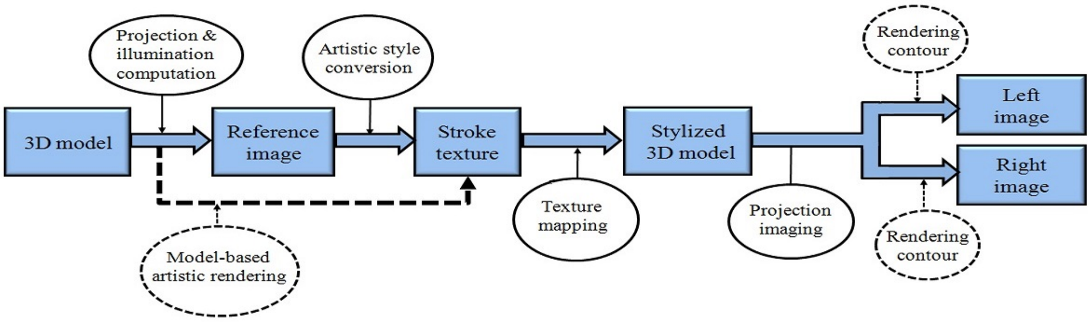

Figure 1: *Rendering flow of the framework in this paper.*

developed to simulate various art styles such as pen-and-ink [KNBH12],watercolor [CAS*97], ink-wash painting [LB14], oil painting [ZZXZ09] and cartoon [BBS*13] [LMHB00].However, in most of these artistic rendering algorithms, the strokes are strongly random and can hardly be applied to rendering the right and left images separately.

Artistic stereoscopic image making: One method for generating stereoscopic images is using two horizontally placed cameras to directly shoot binocular images [HGG*11]. The other method for stereoscopic image making is based on depth map. Using additional depth information, the method generates binocular images by image warping [ZW05] [ZVK11] [LHW*10].

The depth-map-based stereoscopic generation method provides a fine tool to make the artistic stereoscopic images. For example, [SG04] [SG05] used existing image pair to compute depth map and apply artistic rendering to one image. Then acquire the other image by stroke warping. However, holes would occur due to pixel translation and holes painting is still a challenging topic in computer vision. To get depth map, [SSJ*10] gave a fast algorithm for cartoon based on a set of sparse points with equal and unequal depth. The method requires a assigning point depth manually, it is mainly applied in conversion from 2D to 3D. [NAK12] [NAK13] proposed another method. They merged the image pair into one image containing all visible areas of the binocular images. After artistic process, the merged image is decomposed into left and right images. Since final binocular images are from the same artistic image, the consistency can be guaranteed. [KLKL13] discussed the main factors that cause binocular rivalry when creating stereoscopic images using lines only, and suggested to remove the lines without epipolar-slidable point pairs because these lines may disrupt stereo fusion.

The rendering framework provided in this paper is completely different from above methods. In our method, artistic strokes are mapped to 3D model surfaces through texture mapping, and dual cameras directly obtain binocular images. The artistic rendering process is implemented once for generating artistic texture, so randomness of strokes would not affect the consistency of binocular images.

3. Method overview

We use texture mapping to attach an artistic image to 3D model surface so as to get artistic 3D model. The artistic image is called as "stroke texture". Stroke texture can be obtained by model-based artistic rendering methods or style conversion methods based on photorealistic image of model. The photorealistic image is called as "reference image". Since the model will be projected from two different viewing points, reference image and stroke texture must contains entire visible scope from both of the viewing points. Additionally, when generating stroke texture, the texture coordinate must be computed for later texture mapping. About the calculation of texture coordinate will be dicussed in section 5.

Fig.1 shows the rendering process of our framework. The boxes represent the results of each step and the ellipses represent operations. Given a 3D scene, the system will deal with models in the scene one by one. The rendering flow is:

1. Render the reference image first, which contains all visible content of two viewing points. Calculate the vertices' texture coordinate during projection transformation.

2. Apply an artistic style conversion algorithm to the reference image to generate stroke texture with a specific art style. (If a model-based rendering algorithm is applied, the stroke texture can be produced simultaneously at the first step, which represented by a thick dashed line in Fig.1.)

3. Map the stroke texture to model surface, a 3D model with artistic style is acquired as a result.

4. Disable lighting and carry out projection in two camera space to obtain the binocular images. If necessary, render object's contour lines according to model geometry (the operation represented by two dashed ellipses in the last step shown in Fig.1).

4. Stroke texture

The reference image (or stroke texture) must contain all the visible area from two eyes. Our study is similar to the works of [NAK13] and [KJC*11]. [NAK13] merged the existing image pair into one image by map layer in image space. But their method cannot generate texture coordinates. [KJC*11] synthesized all visible points into a buffer (VPB), then refered to the VPB, refracted the projection ray to render two views' visible areas of objects in one image. This method is very accurate but complicated. In this paper, we give a simple approximate solution to capture the visible contents of both view points in one image specially for artistic stereoscopic rendering .

4.1. Render reference image

As shown in Fig.2: Assuming a circle with diameter of $2R$, when use two horizontally placed cameras C_l and C_r to capture binocular images, the solid and dashed lines indicate the visible and invisible parts respectively. Fig.2-a and Fig.2-b individually represent two cases when $2R$ is larger and smaller than the two camera's inter-axis distance c. It is clear that shooting scope varies with the ratio of $2R$ to c.

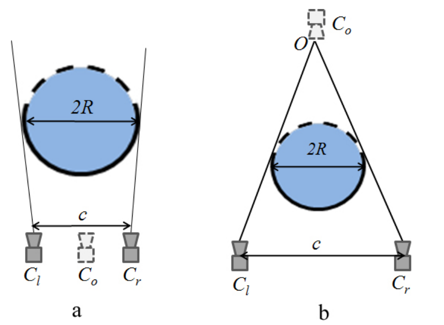

Figure 2: *Shooting scope of dual cameras, a and b are two cases when circle diameter is smaller and larger than camera's shaft distance. Solid and dashed lines indicate the visible and invisible parts respectively.*

In the first case that the circle diameter is more than in-ter-axis distance of the cameras, all scopes that can be shot by both the cameras is less than 1/2 circle. Parallel projection by one camera can obtain a half-circle visible scope, so in this case, an auxiliary camera C_o can be set in the center of C_l and C_r (see Fig.2.a) to get the reference image through parallel projection.

In the second case that $2R$ is less than the inter-axis distance c, the shot scope is larger than 1/2 circle. Assuming the rays from two camera tangent cylinder and intersect at the point O, when the circle is observed from O, the part represented by dashed line become visible while the part of solid line is occluded. In most image rendering systems (OpenGL, DirectX, etc.), the pixel occlusion is judged from the pixel depth value (which stored in Z buffer) after the projection transformation. If the depth values of vertices are changed through projection transformation to make the shaded part to be vidible, then the visible scope from camera C_l and C_r can be rendered with the point O as the viewing point. In this case, an auxiliary camera C_o can be set at the point O (see fig.2.b). We also can see that the closer the camera to the circle, the larger shooting scope can be obtain. So C_o can also axially move forward to some position closer to the circle for a broader scope shot.

Based on the above discussions, we can derive the method of rendering reference image for standard cylinderical objects. If cylinder's diameter is smaller than inter-axis distance, as shown in Fig.3, the auxiliary camera can be positioned at the intersection point of the two rays that from two cameras and tangent the cylinder at which half-high spot. The auxiliary camera is aimed at the cylinder's center. Contrarily, if the diameter is larger than inter-axis distance, we can use a center camera with parallel projection to gain the reference image.

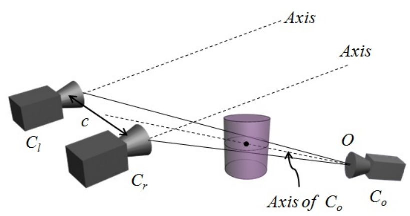

Figure 3: *Using an auxiliary camera C_o to capture the reference image of a cylinder-shaped object in 3D space when the object's diameter is less than inter-axis distance.*

However. in actual 3D scenes, models are not standard cylinder. Although the whole size of an object may be significantly larger than inter-axis distance, we cannot directly use parallel projection to render reference image, because a complex object often has concave surfaces or some parts of which sizes are smaller than inter-axis distance, and parallel projection could lose some visible information of the these areas. The ideal solution is to split the whole model into a series of convex objects and handle them individually. But it would make the problem become very complex and inefficient. Here, we give an approximate solution. To minimize information loss, we use the method for setting auxiliary camera in the second case for all objects. The position and direction of auxiliary camera is determined according to a cylindrical bounding box around the model. The bounding cylinder is created in real time in left(or right) camera space and the cylinder axis is always vertical.

As shown in Fig.4, the auxiliary camera C_o is placed at the line that connects the middle point of dual cameras with the center of bounding cylinder, and aimed at the cylinder's center. The horizontal and vertical view fields of camera

C_o are set to contain the whole bounding cylinder exactly. Assuming the angle of horizontal view range is θ, the distance between the camera and the cylinder axis is d. So, d=R/sin(θ/2). The smaller the d is, the larger the shot scope is, However, small d results in large θ, consequently introduces deformation in result image due to perspective projection. The deformation make the strokes on the closer surfaces become smaller and those further become larger. Here, we give a trade-off solution that θ is set as 90°and $d \approx 1.41R$.

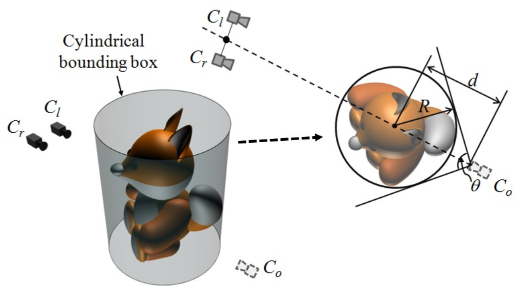

Figure 4: *Use model's cylindrical bounding box to set auxiliary camera.*

Because the auxiliary camera is set behind the object, in order to make the shaded part visible, pixel occlusion relationship should be reversed. Achieving this only requires order reversion of z values through projection matrix. For example in OpenGL, z value is transformed into [-1,1], it is only need to change the sign of z.

The specular calculation requires a view vector which is defined as pointing to viewing point from model vertex in general graphic system. Since the occlusion relationship is reversed, the auxiliary camera points towards the back face of those visible polygons, so the intensity of specular reflection cannot be calculated correctly. Therefore, we define the view vector as pointing to vertex from viewing point.

4.2. Artistic rendering

The stroke texture can be rendered using existing image artistic style conversion algorithms based on the reference image, or obtained directly using model-based artistic rendering algorithms. Most of the existing automatic artistic rendering algorithms can be applied to our framework. Some artistic style need contour lines, so the last step in the framework gives an optional operating module used to generate contour lines when projection is performed in both camera spaces. Many methods can be used to render line, such as [DFRS03] and [NM00]. Further, [KLKL13] proposed an efficient method to ensure binocular consistency when stylize lines using texture mapping. For some artistic style, such as cartoon, the simplest way to ensure consistency is to keep the lines' color and width constant.

5. Texture coordinates

An essential issue to be solved when texture mapping is determination of texture coordinates. When mapping the stroke texture to the model, the texture coordinate to which the model's certain vertex corresponds should be equal to the pixel position where the vertex is projected into reference image (or stoke texture). When projection matrix transforms the model into clip space, the projection point of each vertex corresponds to a coordinate on the imaging plane, i.e. coordinate x_c,y_c in clip space (see Fig.5). Hence, the clip coordinate (x_c,y_c) of model vertex can be considered as the texture coordinate of the vertex. The texture coordinate is usually defined in [0,1], if (x_c,y_c) is beyond [0,1], it should be transformed to[0,1].

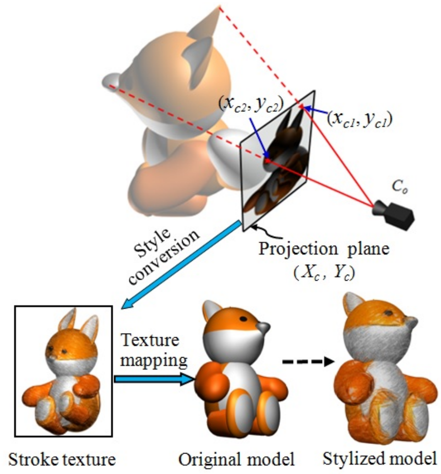

Figure 5: *Calculation of texture coordinates and texture mapping .*

After the artistic processing, then the stroke texture is mapped to the model surface and the model is attached with art style (Fig.5). Actually, texture mapping is the inverse process of the projection for reference image.

6. Results and discussion

Fig.6 shows the oil painting and color pencil style of stereoscopic images rendered by our method. Here, the stroke textures are obtained by style conversion of reference images through the painting plug-ins of PHOTOSHOP. Fig.7 shows other four styles of watercolor, sketch, pastel and car-toon. Fig.7-d use the model-based algorithm [LMHB00] to render stroke texture and object contour lines. According to the figure, we can see that stereoscopic images can be created with fine results for various artistic styles using our method.

Since binocular images are obtained by projecting stylized 3D model in two camera spaces, the consistency of binocular images will not be influenced by the randomness

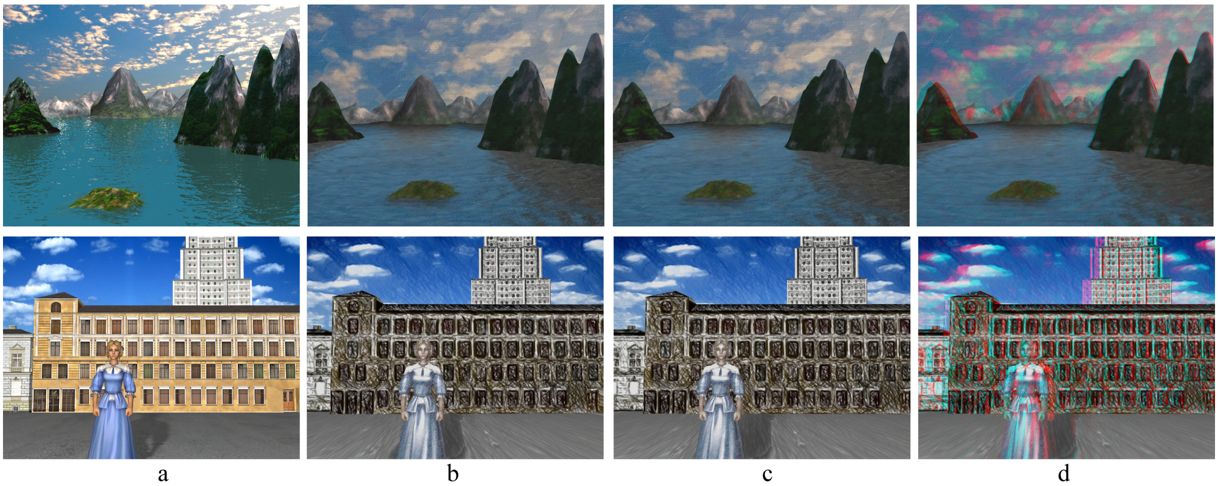

Figure 6: *Rendering results of oil-painting-style(top row) and color-pencil-style(bottom row) stereoscopic images. column a is the original 3D scene, b is left-eye image, c is right-eye image and d is stereoscopic image.*

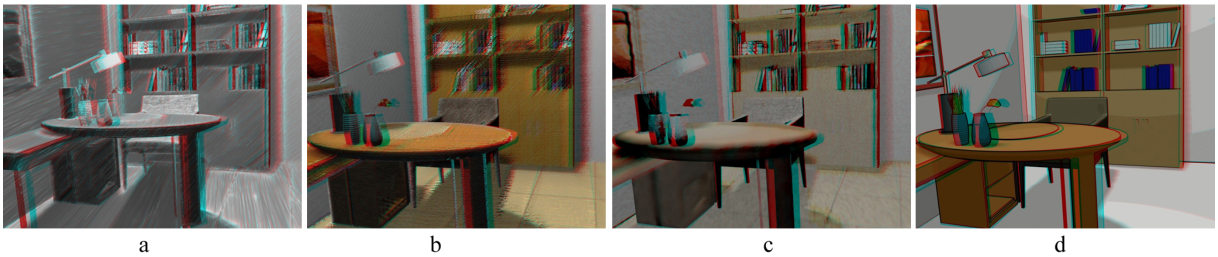

Figure 7: *Rendering results of several art styles. a:sketch style; b:pastel drawing style; c:watercolor style; d:cartoon style.*

of the strokes. The artistic rendering process is only used for generating stroke textures and it is an independent processing step. Therefore, most of the existing artistic rendering algorithms can be applied to the framework. The final artistic effects are dominated by specific artistic rendering algorithm.

To simplified the implement of rendering reference image, the auxiliary camera is set up based on the bounding cylinder of object. However, this is an approximate method, although it can capture the both views' visible areas as maximum as possible, there are still some small visible areas may be missed in reference image. especially for those complex models with many concave surfaces or small subdivisions. But interestingly, we found that there are no obvious visual artifacts in final results in spite of the loss of small visible areas. since the lost areas are usually small and narrow and the strokes have strong uncertainty in size and direction, the randomness of strokes exactly compensates this limitation. In practical applications, 3D models can be preprocessed to reduce the information missing when a 3D scene is created. If some parts of a model can be treated as independent objects, the parts should be detached from the model as individuals. For example, for a character with cap and other props, the cap and props can be modeled as separate objects. The main advantage of our method to render reference image is simple and efficient. Of cause, other more accurate method, such as [KJC*11], can also be applied in our framework.

In some abstract art style, painter may don't carefully describe the border regions of objects, so brush strokes may stretch across an object's boundary. In addition, some pigment has strong diffusion property and ink often diffuse outside the object boundary. In our method, the regions of ink diffusion and the strokes crossing the boundary are cut off along the model's boundary, which make the image look abrupt. Such as the mountains in Fig.6 and the desk in Fig.7-c. Abrupt boundary may deteriorate the artistic effects for some more abstract art style, such as Chinese ink-wash painting, This is the main limitation of our method.

7. Conclusion and future works

For virtual 3D scene, this paper has proposed a complete artistic stereoscopic image rendering framework. Texture mapping is adopted to attach brush strokes on 3D model surfaces. The binocular images are generated directly by model projecting. Therefore, the binocular images can be guaranteed with high consistency. Universal compatibility is another major merit of our method. Various artistic rendering algorithms can be applied to our rendering framework. The capture of the reference image (or stroke texture) with two views' visible areas is the most critical step. Based on characteristics of perspective projection, we gave a simple and intuitive method to capture the reference image. The method can capture the visible contents of dual cameras as maximally as possible.

As discussed in section 6, in view of stereoscopic image acquired by directly using artistic 3D model, for most art styles, fine effect can be got, but in view of some more abstract artistic style, abrupt boundary may deteriorate artistic effects. Hence, based on the study made in the paper, we will continue to investigate how to keep the completeness of the cross-border strokes and ink diffusion effect.

References

[BBS*13] BASSETT K., BARAN I., SCHMID J., GROSS M. H., SUMNER R. W.: Authoring and animating painterly characters. *ACM TOG 32*, 5 (2013), Article No.156. 2

[CAS*97] CURTIS C. J., ANDERSON S. E., SEIMS J. E., FLEISCHER K. W., SALESIN D. H.: Computer-generated watercolor. In *Proc. SIGGRAPH '97* (1997), pp. 421–430. 2

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. *ACM TOG 22*, 3 (July 2003), 848–855. 4

[GGSC98] GOOCH A., GOOCH A., SHIRLEY P., COHEN E.: A nonphotorealistic lighting model for automatic technical illustration. In *Proc. SIGGRAPH '98* (1998), pp. 447–452. 1

[Her98] HERTZMANN A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH '98* (1998), pp. 453–460. 1

[HGG*11] HEINZLE S., GREISEN P., GALLUP D., CHEN C., SANER D., SMOLIC A., BURG A., MATUSIK W., GROSS M.: Computational stereo camera system with programmable control loop. *ACM TOG 30*, 4 (2011), Article No.94. 2

[KJC*11] KIM Y., JUNG H., CHOI S., LEE J., NOH J.: A single image representation model for efficient stereoscopic image creation. *Computer Graphics Forum 30*, 7 (2011), 2067–2076. 3, 5

[KLKL13] KIM Y., LEE Y., KANG H., LEE S.: Stereoscopic 3d line drawing. *ACM TOG 32*, 4 (2013), Article No.57. 2, 4

[KNBH12] KALOGERAKIS E., NOWROUZEZAHRAI D., BRESLAV S., HERTZMANN A.: Learning hatching for pen-and-ink illustration of surfaces. *ACM TOG 31*, 1 (2012), Article No.1. 1, 2

[KT04] KOOI F. L., TOET A.: Visual comfort of binocular and 3-d displays. *Display 25*, 2 (2004), 99–108. 1

[LB14] LI D., BAI C.: An efficient rendering method of wash-ink strokes and its applications. *Journal of Computer-aided Design and Graphics 26*, 3 (2014), 356– 363. 2

[LHW*10] LANG M., HORNUNG A., WANG O., POULAKOS S., SMOLIC A., GROSS M.: Nonlinear disparity mapping for stereoscopic 3d. *ACM TOG 29*, 4 (2010), Article No.75. 2

[LMHB00] LAKE A., MARSHALL C., HARRIS M., BLACKSTEIN M.: Stylized rendering techniques for scalable real-time 3d animation. In *Proc. NPAR '2000* (2000), pp. 13–20. 2, 4

[NAK12] NORTHAM L., ASENTE P., KAPLAN C. S.: Consistent stylization and painterly rendering of stereoscopic 3d images. In *Proc. NPAR '2012* (2012), pp. 47–56. 2

[NAK13] NORTHAM L., ASENTE P., KAPLAN C. S.: Stereoscopic 3d image stylization. *Computer & Graphics 27*, 5 (2013), 389–402. 2, 3

[NM00] NORTHRUP J. D., MARKOSIAN L.: Artistic silhouettes: A hybrid approach. In *Proc. NPAR '2000* (2000), pp. 31–38. 4

[SG04] STAVRAKIS E., GELAUTZ M.: Image-based stereoscopic painterly rendering. In *Proc. EGSR' 2004,* (2004), pp. 53–60. 2

[SG05] STAVRAKIS E., GELAUTZ M.: Stereoscopic painting with varying levels of detail. In *Proc. SPIE 5664, Stereoscopic Displays and Virtual Reality Systems XII* (2005), pp. 450–459. 2

[SSJ*10] SÝKORA D., SEDLACEK D., JINCHAO S., DINGLIANA J., COLLINS S.: Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum 29*, 2 (2010), 615ÍC623. 2

[ZVK11] ZHANG L., VAZQUEZ C., KNORR S.: 3d-tv content creation: Automatic 2d-to-3d video conversion. *IEEE Trans. on Broadcasting 57*, 2 (2011), 372– 383. 2

[ZW05] ZHANG L., WA J. T.: Stereoscopic image generation based on depth images for 3d tv. *IEEE Trans. on Broadcasting 51*, 2 (2005), 191–199. 2

[ZZXZ09] ZENG K., ZHAO M., XIONG C., ZHU S.: From image parsing to painterly rendering. *ACM TOG 29*, 1 (2009), Article No.2. 1, 2