COMMUNICATIONS IN INFORMATION AND SYSTEMS
Volume 14, Number 4, 213–235, 2014

Multiview Conversion of 2D Cartoon Images

SHAO-PING LU, SIBO FENG, BEEREND CEULEMANS, MIAO WANG, RUI ZHONG, AND ADRIAN MUNTEANU*

Multiview images offer great potential for immersive autostereoscopic displays due to the multiple perspectives of a dynamic 3D scene that can be simultaneously presented to a viewer. Traditional 2D cartoons do not contain depth information, and their painting styles are usually quite different from those of real images captured from the real world. This makes existing 2D-to-3D conversion techniques inapplicable because of the difficulty on geometry recovery or lack of sufficient data. This paper introduces an interactive multiview conversion scheme from a single 2D cartoon image. The proposed approach mainly consists of depth assignment and view synthesis. An interactive depth assignment approach is proposed to treat a cartoon image as a composition of ordered depth layers, and the depth can be easily assigned to these layers. A depth smoothing procedure is introduced by solving a Laplace equation with boundary conditions and further depth refinement is performed in order to produce a complete version of the depth map. An interactive image inpainting method is finally proposed to perform multiview image synthesis. The experimental results demonstrate the effectiveness and efficiency of the proposed approach.

1. Introduction

With decades of rapid development of animation industry and research in the fields of computer graphics and image processing, nowadays, the presentation of cartoon becomes more and more diversified. People are no longer accustomed to the classical hand-painted animation. Recent cartoons, like "*Frozen*", "*Zootopia*" and "*Kung Fu Panda*", use highly detailed 3D-models, textures and lighting effects to render beautiful 3D animations. Such 3D-oriented cartoons, although asking for extremely heavy investments of artistic creation and computing workloads, can be easily transformed into stereoscopic or multiview styles for various immersive applications. In contrast to these modern cartoon production methodologies, conventional cartoons were created following a traditional 2D flat production style. In

* Research supported by the 3DLicornea project funded by the Brussels Region (Brussels Institute for Research and Innovation Innoviris).

this context, 2D-to-3D cartoon conversion becomes of crucial importance in order to enable the use of conventional 2D cartoons in immersive applications and 3D animation.

Multiview cartoon generation from 2D cartoon images aims at providing an immersive perception of depth by presenting the audience with multiple view points of the same scene. In this work, the main technical challenges are 2D image-based depth reconstruction and scene texture synthesis. Although the former has been extensively investigated in the last decades, accurate depth estimation from a single image is still an open problem. Furthermore, this problem is particularly challenging for cartoon images as the textures are often overly simplified. Moreover, the monocular cues in traditional cartoons are not as precise as those in images captured by cameras. Conventional 2D pictures are acquired under different lighting conditions, while in cartoons artists need different expression techniques to display various art effects.

Although several advanced learning-based techniques have shown significant capabilities in various image processing domains [5], applying machine learning methods in this context is prohibited by the lack of a sufficiently large training dataset. Such a dataset is difficult to be collected as a huge number of cartoons with consistent artistic styles is required.

A second challenge involves the rendering of multiple views (new perspective images) using rendering techniques based on the generated depth map/3D model and original 2D image. Due to the fact that new virtual images do not contain the information occluded in the original scene, inpainting (hole-filling) with consistent textures is also particularly difficult when using existing image completion methods [26]. Depth image based rendering (DIBR) methods [18] may also be used for 2D-to-3D conversion, but currently they are only suitable for binocular applications whose disparities between the original and virtual views are relatively small [12]; furthermore, these methods may incur additional artifacts when inpainting large holes in the generated virtual views.

In this paper, we propose an efficient semi-automatic method to generate 3D views from 2D cartoon images. In the proposed approach, both depth assignment and image inpainting work with user interactions. Instead of assigning absolute depth values to the original image, the proposed interactive depth assignment algorithm employs a novel processing paradigm whereby (*i*) the input cartoon image is treated as a composition of several depth layers which can be sorted by user interactions, and (*ii*) depth values are assigned to the corresponding layers according to human perception. An interactive inpainting method is also proposed to fill-in the disocclusion holes in the generated virtual views. Subsequently, the proposed approach can successfully propagate texture and structure to the missing

areas (holes) only with a few interactive user scribbles, providing enhanced 3D perception compared to the previous approaches.

In summary, the main contribution of this paper is that we introduce an entire framework for multiview conversion from a single 2D cartoon image. The proposed approach involves depth reconstructions and view synthesis, and requires only very simple user interactions. Our interactive solution can easily avoid various artifacts that are difficult to handle in fully automatic 2D-to-3D conversion methods.

2. Related Work

Our work includes three major technical components: image segmentation, depth assignment and multiview synthesis. We thus provide a brief overview of related work in each area. The state-of-the-art on 2D to 3D conversion will also be reviewed.

Image segmentation. Image segmentation is a fundamental research topic in computer vision, computer graphics and multimedia processing. Existing segmentation algorithms can be classified in terms of low-level grouping or high-level semantic segmentation, hard partitioning or soft matting, automatic or interactive segmentation, etc. [55]. Many *automatic segmentation* algorithms are based on graphs [19, 48] and gradient ascent modeling [14]. The Turbopixel-based approach [33, 36] partitions the image using geometric flows, depending on the local image gradients to distribute superpixels regularly on the image plane. *Interactive segmentation* can be seen as part of seeded region-growing family of algorithms [1], where connected pixels with similar colors are grouped to generate relevant areas for further user interaction. The binary partition tree algorithm [46] performs a hierarchical region segmentation for object-background segmentation. The well-known graph cut algorithm in [7] and its iterative improvement [45] solve the optimization problem using the max-flow/min-cut algorithm. Object-level segmentation for cartoon video tracking is also studied in [63]. With the recent advances in deep learning, segmentation based on convolutional neural networks (CNN) [5] proved their potential, although collection of training data in our application domains is difficult.

Depth assignment. Accurate depth reconstruction is still an open issue even with RGB-D sensors in indoor scenes [13]. Here we briefly review existing depth generation methods from a monocular image. There are various *automatic depth estimation* methods using learning-based or gestalt-based approaches. *Saxena et al.* [47] learn single static image's 3D scene structure and infer orientation using a Markov Random Field (MRF). The method presented in [35] infers the image depth from predicted semantic labels. In [62] trapped-ball segmentation is proposed to distinguish layers for cartoon video vectorization. *Hoiem et al.* [24] recover

an image's occlusion boundaries based on a Bayesian model which is trained on representative training data. In general, learning-based depth estimation strongly depends on the given dataset and segmentation technologies. One representative gestalt-based approach is given in [2], where the region boundaries obtained from segmentation are treated as visual cues for occlusion and are used to estimate the layers in the scene. The approaches in [9, 43] further combine image segmentation and hierarchical representations to perform monocular depth ordering.

Interactive Depth Estimation approaches like [58] allow the user to directly assign depth. These methods are tedious and inefficient even with the help of tailored editing tools[42] or geometric constraints[25]. Some approaches like [34, 54] can produce 3D models if some geometric constraints meet some specific requirements and objects in the image consist of planar surfaces. In [20], contours are used to generate *"artistic blobby objects"* with inaccurate absolute depth values. *Wang et al.* [57] allow the user to directly paint depth on an image through sparse scribbles, and the generated depth cues are processed as soft constraints to propagate to the rest of regions. The method in [60] provides a similar solution to estimate depth using transfusive image manipulation. *Sykora et al.* [51] propose a method to efficiently generate depth maps for a cartoon image by making use of an optimization framework that mimics the way a human reconstructs depth information from a single image. In [37] the authors create a similar framework by solving a Laplace equation. *Iizuka et al.* [28] further reduce the computational burden by simply applying superpixel segmentation for depth propagation.

Multiview synthesis. The view synthesis problem and related editing topics have been well-studied in last decades [26, 44, 49, 64]; interested readers are referred to the survey in [4]. Multiview synthesis takes a texture and depth images as input and generates the textures corresponding to different perspectives. The movement of the virtual camera position with respect to the single reference uncovers parts of the scene that are not present in the reference texture. To render a high-quality image, it is important to complete these missing texture regions – *disocclusions* – in a plausible manner. This is also referred to as Depth-Image-Based Rendering (DIBR). The state of the art in the area includes the method of *Daribo et. al.* [16] which uses depth information and adapts Criminisi's algorithm for disocclusion inpainting [15]. The well-known Patchmatch algorithm [3] has been extended to multiview inpainting [39], and further accelerated by taking depth information into account [38]. In [52], the authors propose to consistently compose and edit stereoscopic images. In [12], a MRF model of overlapping image patches is used to compute an inpainting result which minimizes the MRF energy. *Mu et al.* [41] further perform view completion and depth recovery simultaneously. Interested readers can refer to [40] on recent multiview synthesis and editing techniques.

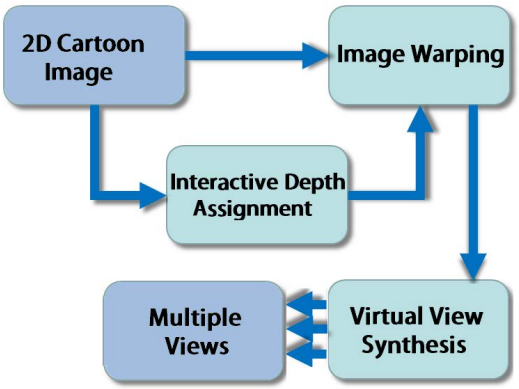

Figure 1: The proposed multiview conversion framework of a 2D cartoon image.

2D-to-3D Conversion. 3D content generation from 2D images has been extensively studied in the context of film-making using 3D scene modeling for single view (2D) or stereoscopic (3D) rendering. Some special applications such as sports live broadcasts, need a real-time 2D to 3D conversion, so *automatic 3D conversion* is preferable. One example can be seen in [10], where the authors exploit monocular depth-cues using colors, gradients and motion [31], which makes the algorithm heavily dependent on the scene complexity. *Kiana et al.* [8] generate stereoscopic 3D (S3D) soccer video based on a domain-specific dataset. Good depth estimation can be also achieved by applying semantic labelling during a learning procedure [35]. However, these methods are still far from practice. *Semi-automatic conversion* can generate better results than these automatic approaches by relying on user interactions; this, of course, makes it difficult to handle 2D videos in real-time.

Other approaches focus on interactively applying depth estimation for the key frames in a video. By doing so, *Varekamp et al.* [53] use bilateral filtering and motion compensation to propagate annotated depth to other frames in the video sequence. An object segmentation algorithm is used in [11] to the generate disparity maps for the key-frames. *Wu et al.*[59] propagate the depth by tracking the objects in non-key frames via a bi-directional Kanade-LucasTomashi (KLT) optical flow estimation algorithm. A more recent method [27] calculates depth based on a Bayesian framework and uses a Natural Scene Statistics (NSS) model to guide depth propagation. These luminance-based depth propagation methods are unsuitable for our 3D cartoon conversion, since traditional cartoons do not have photo-realistic luminance information as in a real environment.

Figure 2: Cartoon image pre-segmentation. Left: input image. Middle: user interaction with few scribbles. Right: segmented result.

3. Proposed solution

Fig. 1 shows the overall framework of the proposed multiview cartoon conversion method. Our approach consists of three parts: 1) interactive depth assignment, 2) view warping , and 3) view synthesis by interactive inpainting. Our system introduces an effective multiview generation workflow exploiting simple user interactions for both depth estimation and view synthesis. Unlike the approach in [37], the proposed method allows for interactive user inputs where the user can improve the synthesis result at every stage, getting visual feedback in real-time. The depth assignment process in [51] is improved and it becomes more applicable for multiview generation. The proposed interactive approach can easily propagate the expected texture and structure information to the missing areas (i.e. holes generated by view warping).

3.1. Interactive depth assignment

Most traditional 2D cartoons, such as *"Tom and Jerry"* and *"Mickey Mouse"*, are initially hand-made drawings which delineate objects and characters with black contour lines. Colors within a single entity are mostly similar, and textures of cartoons are much simpler than those of real-life photographs. In order to efficiently extract the objects and background regions, we build a depth map starting from a few scribbles which are propagated using a pre-segmentation and by solving a Laplace equation.

Pre-segmentation and coarse depth assignment. We first achieve multi-label pre-segmentation using [50], which uses graph-cuts [6] to find a consistent labeling based on initial user scribbles, pixel intensities and a Potts-model. This results in a coarse segmentation of the texture image. In order obtain a depth

map, the segments should be assigned depth values, respecting the existing foreground/background relations. We follow the method introduced in [30, 50] to solve this ordering problem. That is, each pre-segmented region in the image is treated as a node (vertex), and the directed edge between two nodes is defined by user interactions. The user draws lines to indicate the expected relative depth relationship between adjacent regions. After completing the interactions, all pre-segmented regions are sorted in a order using topological sorting [30]. If the graph which results from the user interactions is not acyclic, a topological sorting is not achievable. This situation would however also not be plausible and can easily be detected so it can be prompted to the user. Fig. 3 shows how the user specifies the relationship between layers. After that, the user can assign the specific depth values to the sorted layers. To do so, linear interpolation is used to assign desired depth values to those sorted layers [51]. We note that linear interpolation in [51] can be used for 2.5D cartoon pop-up, but it is unsuitable for multiview generation, because those pre-segmented layers are seldom positioned linearly in an image. As shown in Fig. 3, each defined line (white arrow) has a distance r_i where index i indicates the order of layers. Then the depth d_i for layer i is formulated as

$$d_i = D_{max} - \frac{D_{max} - D_{min}}{r_i} \cdot \sum_{i=0}^{N-1} \frac{1}{r_i}, \tag{1}$$

where D_{max} is the maximum depth value of all foreground layers that correspond to the first layer (object closest to audience), while D_{min} is the minimum depth value of the last layer (the furthest background). D_{max} and D_{min}, which are defined by the user, should satisfy the conditions: $1 \leq D_{max} \leq 255$, $1 \leq D_{min} \leq 255$ and $D_{min} \leq D_{max}$. N is the total number of directed edges which is equal to the total number of layers. It is easy to notice that depth assignment has a inverse relationship with the distance r. This approach is more appropriate than linear interpolation, since the depth can be assigned based on the relative position of objects defined by the user.

Depth refinement. Once the coarse depth map of the cartoon image is generated, more detailed depth transition between different layers should be considered. Similar to the approach introduced in [28, 51], we smooth transitions of depth whilst preserving the depth discontinuities. To achieve this, a contour-preserving smoothing method is proposed to refine the coarse depth map, and it can be formulated as the following quadratic problem:

$$I_{depth} = \arg\min_{\mathcal{I}_{tmp}} \sum_{p \in \mathcal{I}_{tmp}} \sum_{q \in \mathcal{N}_p} \Omega_{pq} (x_p - x_q)^2, \tag{2}$$

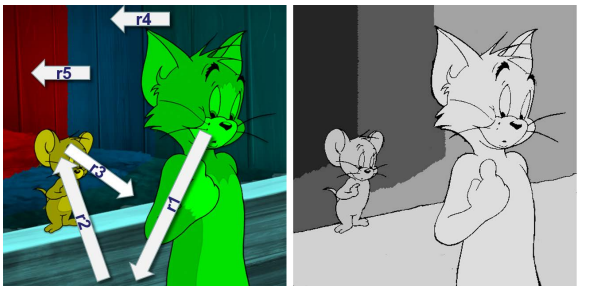

Figure 3: Layers sorting. Left: pre-segmented layers and sorting by user interaction. Right: the sorted result with desired depth information.

where x is the unknown smoothed depth value. $\mathcal{I}_{tmp}$ denotes the coarse depth map obtained from d_i in Eq. 1. x_p and x_q represent the pixels' depth values from $\mathcal{I}_{tmp}$. Pixel q belongs to the 4-connected neighboring pixels of pixel p. Ω_{pq} is the weighting function used to achieve depth transitions:

$$\Omega_{pq} = \begin{cases} \lambda e^{-\beta(\mathcal{I}_p - \mathcal{I}_q)^2} & \text{if } d_p \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The weights Ω_{pq} are computed from image gradients in the original texture image $\mathcal{I}$. The parameter λ is used to control the degree of smoothness. β is a constant ($\beta = 150$ by default) deciding how much the gradients will influence the depth transition. When $d_p = 0$, the weighting function $\Omega_{pq} = 0$, since the black contours in the coarse depth map will not participate in the smoothing procedure. For the regions where Ω_{pq} is non-zero, depth will be smoothed out based on image gradients from original image. The corresponding homogeneous regions in the original image will be better smoothed so that some geometrical relationship will be protected in the final depth map.

However, the gradient information from the original image is a double-edged sword, since it maintains the objects' geometries but prevents depth propagation and it introduces discontinuities in the depth maps. This problem will be addressed in the next step.

To minimize the energy function in Eq. 2, the following Laplace equation needs to be solved:

$$\nabla^2 x = 0 \tag{4}$$

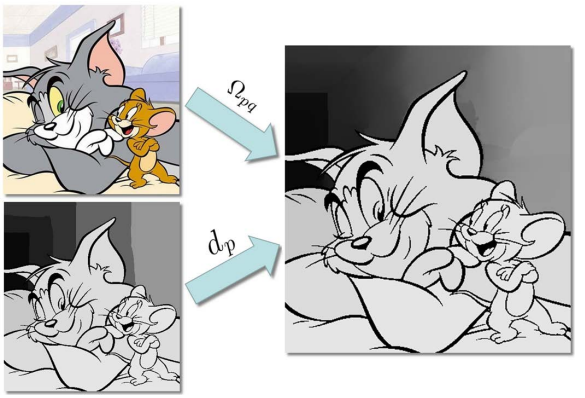

Figure 4: Coarse depth refinement with gradient information. In left column, the original image at the top provides gradient information and the coarse depth map at the bottom provides each layer's depth information. Image in the right column shows the smoothed result.

with the Dirichlet and Neumann boundary conditions, respectively:

$$\begin{aligned} Dirichlet: &\quad x_p = d_p \qquad if\ p \in \mathcal{I}_{tmp} \\ Neumann: &\quad x_p - x_q = 0 \quad if\ (d_p \neq d_q) \cap (d_p \neq 0). \end{aligned}$$

The Dirichlet boundary condition can be interpreted as the meaning that pixels aim to maintain their initial depth (the coarse depth map $\mathcal{I}_{tmp}$) as much as possible. The Neumann boundary is a higher order condition that makes depth values transit in the regions where the area does not contain contours ($d_p \neq 0$). Then the Laplace equation (4) becomes:

$$\mathrm{M} \cdot \mathrm{X} = \mu \cdot \mathrm{D} \tag{5}$$

where M is a $n \times n$ Laplace matrix and n is the number of pixels in original image. X is a unknown column vector indicating the smoothed depth values and D is the vector given by coarse depth map. μ is a constant used for controlling the smoothness of the depth map.

Up to now, the regions covered by contours are not taken into account as they were used to preserve depth discontinuities (see one example in Fig. 4). Thus, depth has to be expanded to the contours. Traditional methods perform depth expansion by eroding the regions based on the order of depth layers. However, this is only applicable for thin contours. An improved method [51] consecutively

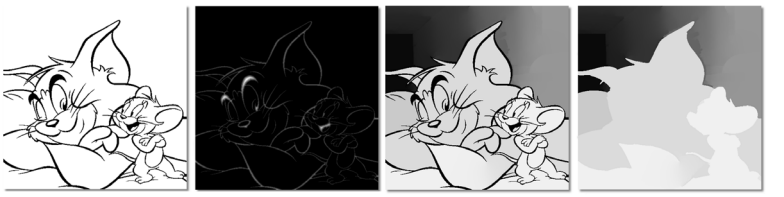

Figure 5: Depth expansion to contours. From left to right: contour mask; normalized distance map; depth map to be expanded to contours; expansion result.

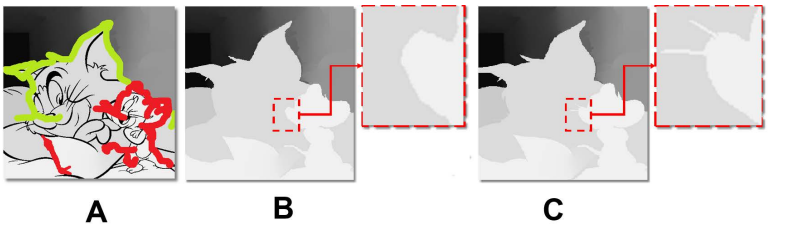

Figure 6: Interactive contour refinement. A: Strokes made by the user. B: Coarse depth map. C: refined depth map.

expands depth in a front-to-back order (obtained from sorting), which can successfully solve the expansion problem that is mentioned in the first case. But its solution is not efficient, since the depth expansion needs to solve another Laplace equation for each depth layer. Furthermore, the user needs to choose suitable thresholds for each depth expansion.

In this work, we introduce a simple interactive approach consisting of the following two phases: 1) coarse depth expansion and 2) contour refinement. First, a binary mask is created for contours and then the distance transform of a contour mask is computed based on

$$d_p = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}, \tag{6}$$

where d_p denotes the Euclidean distance between the pixels p and its nearest nonzero pixel q.

As shown in Fig. 5, once the distance map is obtained, the depth in contours are given by their closet pixels in the depth map. With this efficient approach, our depth expansion only can be instantly achieved even for a HD resolution image.

Finally, due to possible errors introduced in the coarse depth propagation, user interaction can be involved in the contour refinement. As shown in Fig. 6, the user can directly draw on the specified depth maps and the contours covered by strokes

will get the same depth value as the pixel where the stroke covers. Consequently, all these changes will be added to the coarse result to generate a complete depth map. Next, we are going to generate multiview images by making full use of the original image and its associated depth map.

3.2. Interactive depth-based view synthesis

The multiview generation introduced in this paper is applied to extrapolate multiple viewpoints from a single texture image [23]. As shown in the Fig. 7, numerous missing holes exist in the new image of the target viewpoint by view warping. One single discrete image cannot contain the complete 3D information to fully generate a new perspective view. Moreover, the depth map generated by the aforementioned depth assignment method may be inaccurate. Thus, inpainting is usually utilized to complete the virtual texture images.

As can be seen again in Fig. 7, the generated holes can be roughly classified into two categories: small cracks and disocclusion holes. Small cracks are usually several pixels wide, but disocclusion areas are caused by depth discontinuities whereby the involved pixels have different disparities. Small cracks can be easily removed by using median filtering followed by morphological filters: erosion and dilation. Erosion is used to fix the small holes and dilation is applied to compensate the eroded disocclusion areas.

Next we employ a depth-based pixel-level inpainting algorithm [39] to fill the remaining holes. Moreover, we further introduce interactive constraints to achieve a more robust inpainting result. Similar to exemplar-based approaches, the pixel-level inpainting fills the missing holes based on the known regions $\mathcal{I}_k$:

$$\mathcal{I}_k = \mathcal{I} - \mathcal{I}_h, \tag{7}$$

where $\mathcal{I}$ is the warped image and $\mathcal{I}_h$ is a mask denoting the hole areas. Then, the problem is how to find an optimal candidate from the known region to fill missing pixels. Firstly, the similarity between a patch from the known region and a patch centered on the hole boundary (pixel $m \in \partial\mathcal{I}_h$) is computed. The optimal candidate $\mathcal{P}_{n'}$ should satisfy:

$$\mathcal{P}_{n'} = \arg\min_{n \in \mathcal{I}_k} E(\mathcal{P}_m, \mathcal{P}_n), \tag{8}$$

where $\mathcal{P}_n$ is an optional patch from known regions and $\mathcal{P}_m$ is the patch on the hole boundary. E is a similarity metric; in our case the Sum of Squared Differences

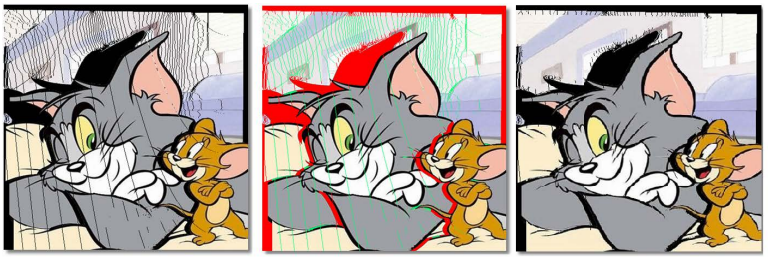

Figure 7: Different holes by view warping. Left: warped image. Middle: hole classes with different colors. Right: small crack filling result.

(SSD):

$$E(\mathcal{P}_m, \mathcal{P}_n) = \frac{1}{N_m} ||\mathcal{P}_m - \mathcal{P}_n||^2, \tag{9}$$

where N_m is the total number of known pixels within the patch $\mathcal{P}_m$.

Our dissoclussion inpainting solution is initialized by randomly selecting candidates from $\mathcal{I}_k$ to fill-in the pixels in missing holes $\mathcal{I}_h$; we then search the candidates (patches) throughout the known regions and calculate the corresponding similarity metric; finally, we iteratively update the optimal patch with minimum E to fill the hole's boundary. It is easy to imagine that the procedure of searching candidates throughout the known regions is very time-consuming. In order to find the optimal candidate more efficient and avoid trapping in the local minimum of similarity metric E, a logarithm search strategy is applied

$$\mathcal{C}_m = \bigcup_{(i,j),(i',j')} (x' + \alpha_i R_j, y' + \alpha_{i'} R_{j'}) \tag{10}$$

where $\mathcal{C}_m$ is the set of candidates for the hole at pixel m. (x', y') is the coordinate of center pixel for current optimal patch from $\mathcal{I}_k$. $R = [-1, 0, 1]$ represents the searching direction and $\alpha = [64, 32, 16, 8, 4, 2, 1, 0]$ decreasing exponentially indicate the searching radius.

This pixel-level inpainting algorithm can work iteratively since once all pixels on boundary $\partial\mathcal{I}_h$ are filled by their best candidates, their neighbor pixels belonging to $\mathcal{I}_h$ will be treated as the updated $\partial\mathcal{I}_h$ that will be put in the filling queue. Moreover, we further limit the search space of the disoculusion areas to the known background regions and all specified areas drawn by the user. Thus, the search space is defined as:

$$\mathcal{S} = \mathcal{S}_{\mathcal{I}_k} \bigcap \mathcal{S}_{\mathcal{D}}, \tag{11}$$

Figure 8: View synthesis by interactive inpainting. Red pixels are to be disoculuded areas; green lines are used to specify search space by the user.

where $\mathcal{S}_{\mathcal{I}_k}$ denotes that the search space in the known regions. $\mathcal{S}_{\mathcal{D}}$ is defined as

$$\mathcal{S}_{\mathcal{D}} = \bigcup_{m \in \mathcal{S}_{\mathcal{I}_k}} \{(\mathcal{D}(m) < u_{max}) \cap (\mathcal{D}(m) > u_{min})\}. \tag{12}$$

$\mathcal{D}(m)$ is the depth value at pixel m. u_{max} is the maximum depth of the regions covered by user strokes and u_{min} is the corresponding minimum depth.

Finally, as shown in the example in Fig. 8, the user can easily draw strokes to define the filling regions and specify some search areas for hole filling.

4. Experiments and Discussion

4.1. System implementation and parameters

The proposed multiview cartoon conversion solution is implemented in C++. Additionally, we configured the Laplace matrix solver by using the MATLAB graph analysis toolbox [22]. All the calculations are done in $CIELab$ color space. The whole procedure processing a 512×512 image takes less than 10 minutes for a skilled user on a laptop with 2.5GHz Quad-Core Intel i7-4710MQ CPU, 8GB memory and NVIDIA GTX860M GPU. In general, the proposed system allows the user to achieve real-time interaction for both depth assignment and virtual view synthesis on a HD resolution cartoon image.

In depth assignment phase, there are two main parameters, λ and μ, to control depth smoothing. If we fix λ and increase μ, the values of pixels x_p will be more likely to maintain the initial depth values of the coarse depth map and vice versa.

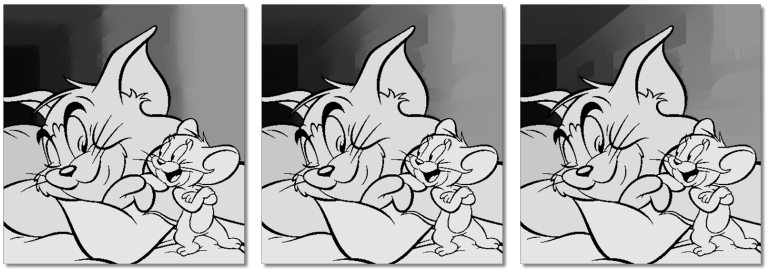

Figure 9: Three optimized depth maps with different λ while $\mu = 1$ and $\beta = 150$. Left: $\lambda = 150$. Middle: $\lambda = 1500$. Right: $\lambda = 15000$.

In our implementation, we set λ as a control parameter and set μ to a constant. By doing so, the smoothness of the depth maps can be manipulated by λ. As shown in Fig. 9, we find that the middle result is better than others, and it is critical to choose a good λ value to obtain a desired result. Because the computation of this step takes less than 5 seconds even for large images, our user-interface allows the user to change λ whenever this is desired; the user can also decide to stop this step based on the visual results.

In the virtual view synthesis processing, we empirically set the threshold $\mathcal{D}_k$ to 0.75, which is used to limit the search space for disocclusion inpainting.

4.2. Experimental results

We have produced many multiview conversion results for various cartoon images using the proposed approach. For example, Fig. 10 presents the whole multiview generation procedure for the "$Pokémon$" cartoon image. In this case, there are four $Pokémons$ with different positional relationships. In order to obtain the expected depth, the proposed approach first performs pre-segmentation of the input image. After that, the user specifies the expected depth relationships for the main segment layers. With further user interaction, the refined depth map is produced (see the second subfigure of the bottom row). Once the desired depth map is generated, the proposed approach warps both the color and depth images to the expected viewpoint designed by the user (see the third column of the same figure). Finally, on the last column we can see the view synthesis result for both the color and the depth images. One can also observe that the top of the chick's head that is closest to the audience has been spatially shifted to be away from other background objects.

Another example for the input image *Doraemon* is depicted in Fig. 11. The same processing workflow is followed as explained before. The proposed approach

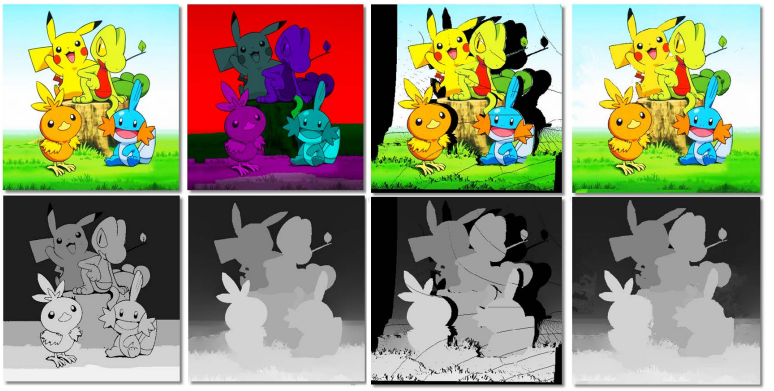

Figure 10: Multiview conversion for "*Pokémon*". Top row: original image, presegmentation, view warped color image, view synthesized color image. Bottom row: coarse depth maps, refined depth maps, view warped depth image, view synthesized depth image.

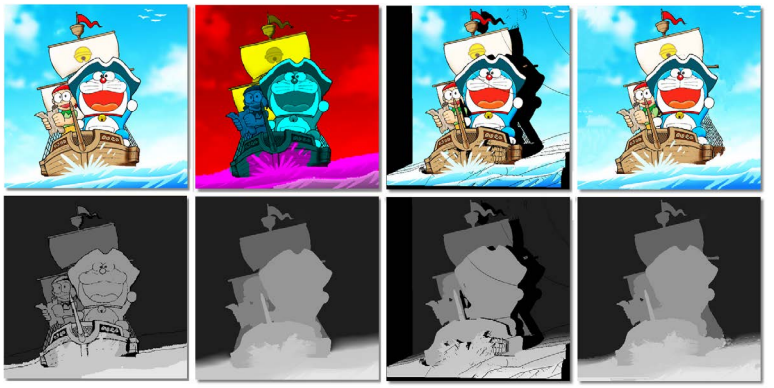

Figure 11: Multiview conversion for "*Doraemon*". Top row: original image, presegmentation, view warped color image, view synthesized color image. Bottom row: coarse depth maps, refined depth maps, view warped depth image, view synthesized depth image.

is able to produce plausible multiview conversion results, as illustrated in Fig. 11. In the generated new viewpoint, the cloud in the sky moves to the right. On the other side, the Doraemon also shifts to the middle of the flag. These phenomena are coherent with the fact that the camera moves from left to right.

In Fig. 12, multiple views are generated from a single 2D cartoon image using the proposed approach. Because our solution can generate a plausible depth map

Figure 12: Many different views generated by a single 2D cartoon image using the proposed approach.

for the input image, the trees and stone on the background are reasonably shifted to different imaging planes following the given camera model. These many views generated from a static 2D image can easily allow the viewer to navigate the scene from different viewpoints (see the supplemental video demo).

Fig. 13 shows the comparison with another interactive depth generation method introduced in [21]. Again, in this example our interactive solution can produce a good depth map. In comparison, the middle subfigure of the bottom row presents the depth generated by Random Walk-based solution [21], where the depth has poor transitions at the regions with strong image gradients. One can observe that with only a few simple user interactions, the depth map produced by our solution is more precise than that of [21]. Moreover, when performing the proposed view synthesis based on such depth maps, our solution generates much more plausible contents (see the reconstructed mice on the right column).

We further tested a popular depth map recovery method [61], where structure from motion (SfM) is used to estimate depth for video sequences in a fully automatic manner. The estimated depth maps are shown in Fig. 14. In order to compare the performance subjectively, we also synthesize a new virtual view using the corresponding depth maps generated by the above-mentioned methods. We can see that the depth generated by SfM-based method is far from accurate on this data. The view synthesis result using the depth produced by SfM is very noisy, and many pixels are wrongly mapped to the new view. There are several reasons that could explain these results. Firstly, the contents in these video frames have barely changed within a very limited amount of time which makes it hard to find the disparities of pixels between frames. Secondly, it is difficult to identify corresponding feature points between frames featuring the simple textures of cartoons. Therefore, although SfM techniques are more and more popular on various 3D reconstruction applications, they are not directly applicable for traditional 2D cartoon scenes.

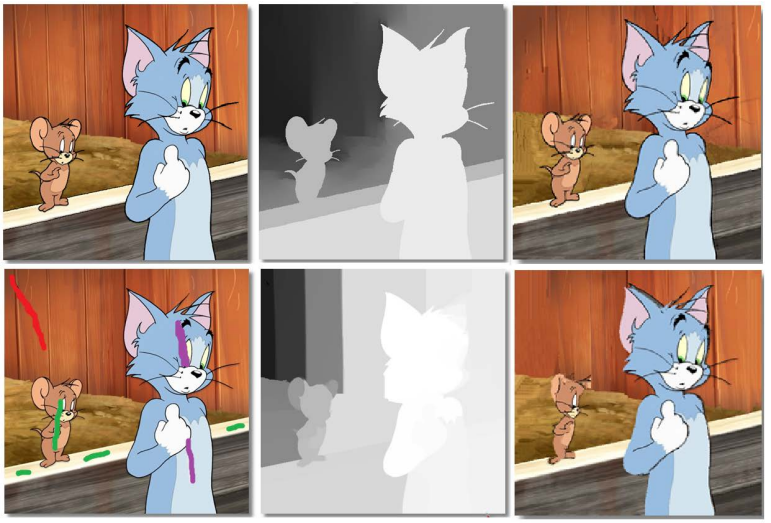

Figure 13: Multiview conversion with different depth generation methods. Top row: original image, depth map generated by our solution, and our view synthesis result, respectively. Bottom row: original image with user interactions, depth generated by a Random Walk algorithm [21] and corresponding view synthesis result, respectively.

4.3. Discussion and limitations

The view synthesis method in the proposed approach is based on [39]. By introducing user interactions, our solution can avoid various visual artifacts that are difficult to handle fully automatically. Currently, structure-oriented image completion and view synthesis (e.g. [26, 29]) are still very challenging. However, limited user interactions prove to effectively overcome this issue, which motivates the user interactivity built in the proposed method.

The proposed depth assignment method can be further improved using some more precise matting (e.g. the closed-form solution in [32]). The user can select the objects with few strokes by using image matting techniques and then depth will be precisely expanded to the selected areas. Although it provides an elegant method for depth expansion, such an approach is much more time-consuming compared to the proposed method - see the example obtained with image matting in Fig. 15.

Although high quality, pleasant multiview results are obtained with our method, visual artifacts cannot still be avoided in some cases. For example, weak contours where pixel intensities are very close to those of neighboring pixels are difficult to

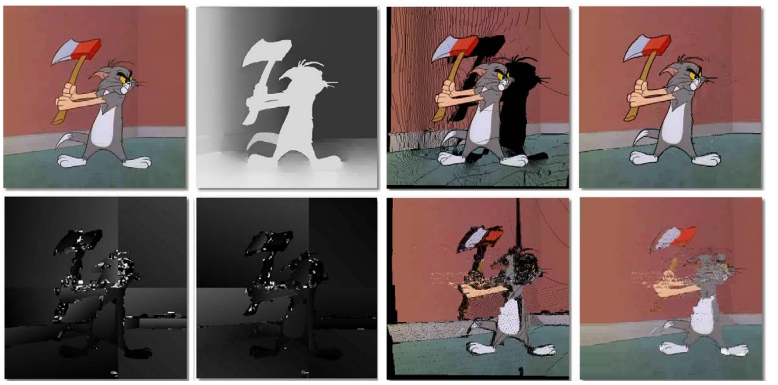

Figure 14: Comparison against the results obtained with the structure from motion method of [61]. Top row shows the original image, generated depth by our method, corresponding warping result and view synthesis, respectively. Bottom row shows two consecutive frames of depth maps generated by structure from motion [61], corresponding view warping and synthesized results, respectively.

extract in our solution. Conflicting constraints imposed by user interactions could also cause our method to synthesize undesired solutions.

5. Conclusions and Future Work

This paper proposes an interactive conversion method to generate multiple novel viewpoints starting from a single 2D cartoon image. Our work involves multi-label pre-segmentation and depth assignment for depth map generation from the input 2D cartoon image. Using this depth map, virtual view synthesis is employed to generate the desired viewpoints of the scene. The proposed solution provides stable and plausible results by making use of simple user interactions. Experimental results demonstrate the effectiveness and efficiency of the proposed approach for multiview cartoon conversion.

Further investigation enforcing temporal consistencies for the proposed multiview cartoon video conversion method is one of the potential avenues for future work. Moreover, although the proposed approach enables the user to specify different camera matrix parameters for virtual view synthesis, comfort-driven depth adjustment (e.g. [17, 56]) is an interesting research aspect to be considered in our future work.

Figure 15: Image matting could improve the depth assignment. Left: user interaction; middle: α matting result; right: refined depth maps (see the improvement on the red boundaries).

References

[1] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.

[2] M. R. Amer, R. Raich, and S. Todorovic, "Monocular extraction of 2.1 d sketch." in *Proc. Int. Conf. Image Proc.*, 2010, pp. 3437–3440.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24:1–10, 2009.

[4] C. Barnes and F.-L. Zhang, "A survey of the state-of-the-art in patch-based synthesis," *Computational Visual Media*, pp. 1–18, 2017.

[5] G. Bertasius, L. Torresani, S. X. Yu, and J. Shi, "Convolutional random walk networks for semantic image segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, July 2017.

[6] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[7] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proc. Int. Conf. Comput. Vis.*, vol. 1, 2001, pp. 105–112.

[8] K. Calagari, M. Elgharib, P. Didyk, A. Kaspar, W. Matusik, and M. Hefeeda, "Gradient-based 2D-to-3D conversion for soccer videos," in *Proc. ACM Conf. Multimedia*, 2015, pp. 331–340.

[9] F. Calderero and V. Caselles, "Recovering relative depth from low-level features without explicit t-junction detection and interpretation," *Int. J. Comput. Vision*, vol. 104, no. 1, pp. 38–68, 2013.

[10] X. Cao, A. C. Bovik, Y. Wang, and Q. Dai, "Converting 2D video to 3D: An efficient path to a 3D experience," *IEEE Multimedia*, vol. 18, no. 4, pp. 12–17, 2011.

[11] X. Cao, Z. Li, and Q. Dai, "Semi-automatic 2D-to-3D conversion using disparity propagation," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 491–499, 2011.

[12] B. Ceulemans, S.-P. Lu, G. Lafruit, P. Schelkens, and A. Munteanu, "Efficient mrf-based disocclusion inpainting in multiview video," in *Proc. Int. Conf. Multimedia and Expo.* IEEE, 2016, pp. 1–6.

[13] K. Chen, Y.-K. Lai, and S.-M. Hu, "3D indoor scene modeling from rgb-d data: a survey," *Computational Visual Media*, vol. 1, no. 4, pp. 267–278, 2015.

[14] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[15] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[16] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *Proc. IEEE MMSP*, 2010, pp. 167–170.

[17] S.-P. Du, B. Masia, S.-M. Hu, and D. Gutierrez, "A metric of visual comfort for stereoscopic motion," *ACM Trans. Graph.*, vol. 32, no. 6, p. 222, 2013.

[18] C. Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3D-TV," in *Electronic Imaging of International Society for Optics and Photonics*, 2004, pp. 93–104.

[19] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[20] Y. Gingold, T. Igarashi, and D. Zorin, "Structured annotations for 2D-to-3D modeling," *ACM Trans. Graph.*, vol. 28, p. 148, 2008.

[21] L. Grady, "Multilabel random walker image segmentation using prior models," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2005, pp. 763–770.

[22] L. Grady and E. Schwartz, "The graph analysis toolbox: Image processing on arbitrary graphs," *CAS CNS Technical Report Series*, no. 021, 2010.

[23] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[24] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.

[25] Y. Horry, K.-I. Anjyo, and K. Arai, "Tour into the picture: using a spidery mesh interface to make animation from a single image," in *Proc. SIGGRAPH*, 1997, pp. 225–232.

[26] J.-B. Huang, K. S. Bing, A. Narendra, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–10, 2014.

[27] W. Huang, X. Cao, K. Lu, Q. Dai, and A. C. Bovik, "Toward naturalistic 2D-to-3D conversion," *IEEE Trans. Image Process.*, vol. 24, no. 2, pp. 724–733, 2015.

[28] S. Iizuka, Y. Endo, Y. Kanamori, J. Mitani, and Y. Fukui, "Efficient depth propagation for constructing a layered depth image from a single image," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 279–288, 2014.

[29] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, p. 107, 2017.

[30] A. B. Kahn, "Topological sorting of large networks," *Commun. ACM*, vol. 5, no. 11, pp. 558–562, 1962.

[31] D. Kim, D. Min, and K. Sohn, "A stereoscopic video generation method using stereoscopic display characterization and motion analysis," *IEEE Trans. Broadcast.*, vol. 54, no. 2, pp. 188–197, 2008.

[32] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, 2008.

[33] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 12, pp. 2290–2297, 2009.

[34] H. Lipson and M. Shpitalni, "Optimization-based reconstruction of a 3D object from a single freehand line drawing," in *ACM SIGGRAPH courses*, 2007, p. 45.

[35] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1253–1260.

[36] Y. J. Liu, M. Yu, B. J. Li, and Y. He, "Intrinsic manifold slic: A simple and efficient method for computing content-sensitive superpixels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2017.

[37] A. Lopez, E. Garces, and D. Gutierrez, "Depth from a single image through user interaction," *Proc. CEIG*, pp. 1–10, 2014.

[38] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, "Performance optimizations for patchmatch-based pixel-level multiview inpainting," in *Proc. IC3D.* IEEE, 2013, pp. 1–7.

[39] S. Lu, J. Hanca, A. Munteanu, and P. Schelkens, "Depth-based view synthesis using pixel-level image inpainting," in *Proc. Int. Conf. Digital Signal Process.*, 2013, pp. 1–6.

[40] S. Lu, T. Mu, and S. Zhang, "A survey on multiview video synthesis and editing," *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 678–695, 2016.

[41] T.-J. Mu, J.-H. Wang, S.-P. Du, and S.-M. Hu, "Stereoscopic image completion and depth recovery," *The Vis. Comput.*, vol. 30, no. 6-8, pp. 833–843, 2014.

[42] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," in *Proc. SIGGRAPH*, 2001, pp. 433–442.

[43] G. Palou and P. Salembier, "Monocular depth ordering using t-junctions and convexity occlusion cues," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1926–1939, 2013.

[44] P. Prez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, pp. 313–318, 2003.

[45] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[46] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 561–576, 2000.

[47] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[48] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[49] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, 2005.

[50] D. Sykora, J. Dingliana, and S. Collins, "Lazybrush: Flexible painting tool for hand-drawn cartoons," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 599–608, 2009.

[51] D. Sykora, D. Sedlacek, S. Jinchao, J. Dingliana, and S. Collins, "Adding depth to cartoons using sparse depth (in) equalities," *Comput. Graph. Forum*, vol. 29, pp. 615–623, 2010.

[52] R.-f. Tong, Y. Zhang, and K.-L. Cheng, "Stereopasting: interactive composition in stereoscopic images," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 8, pp. 1375–1385, 2013.

[53] C. Varekamp and B. Barenbrug, "Improved depth propagation for 2D to 3D video conversion using key-frames," in *Proc. IETCVMP.*, 2007, pp. 1–7.

[54] P. A. Varley and R. R. Martin, "Estimating depth from line drawing," in *Proc. ACM symposium on Solid modeling and applications*, 2002, pp. 180–191.

[55] J. Wang, "Foreground segmentation in images and video: Methods, systems and applications," Ph.D. dissertation, University of Washington, 2007.

[56] M. Wang, X.-J. Zhang, J.-B. Liang, S.-H. Zhang, and R. R. Martin, "Comfort-driven disparity adjustment for stereoscopic video," *Computational Visual Media*, vol. 2, no. 1, pp. 3–17, 2016.

[57] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross, "Stereobrush: interactive 2D to 3D conversion using discontinuous warps," in *Proc. Eurographics Symposium on Sketch-Based Interfaces and Modeling*, 2011, pp. 47–54.

[58] L. Williams, "3D paint," in *Proc. SIGGRAPH*, 1990, pp. 225–233.

[59] C. Wu, G. Er, X. Xie, T. Li, X. Cao, and Q. Dai, "A novel method for semi-automatic 2D to 3D video conversion," in *Proc. 3DTV-CON*, 2008, pp. 65–68.

[60] K. Yücer, A. Sorkine-Hornung, and O. Sorkine-Hornung, "Transfusive weights for content-aware image manipulation." in *Proc. VMV*, 2013, pp. 57–64.

[61] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, 2009.

[62] S.-H. Zhang, T. Chen, Y.-F. Zhang, S.-M. Hu, and R. R. Martin, "Vectorizing cartoon animations," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 4, pp. 618–629, 2009.

[63] H. Zhu, X. Liu, T.-T. Wong, and P.-A. Heng, "Globally optimal toon tracking," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–10, 2016.

[64] Z. Zhu, H.-Z. Huang, Z.-P. Tan, K. Xu, and S.-M. Hu, "Faithful completion of images of scenic landmarks using internet images," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 8, pp. 1945–1958, 2016.

DEPARTMENT OF ELECTRONICS AND INFORMATICS (ETRO), VRIJE UNIVERSITEIT BRUSSEL (VUB)
PLEINLAAN 2, BRUSSELS B-1050, BELGIUM
E-mail address: slu@etrovub.be

DEPARTMENT OF ELECTRONICS AND INFORMATICS (ETRO), VRIJE UNIVERSITEIT BRUSSEL (VUB)
PLEINLAAN 2, BRUSSELS B-1050, BELGIUM
E-mail address: sfeng@etrovub.be

DEPARTMENT OF ELECTRONICS AND INFORMATICS (ETRO), VRIJE UNIVERSITEIT BRUSSEL (VUB)
PLEINLAAN 2, BRUSSELS B-1050, BELGIUM
E-mail address: bceulema@ETROVUB.be

DEPARTMENT OF COMPUTER SCIENCE, TSINGHUA UNIVERSITY
FIT 3-523, BEIJING 100084, P. R. CHINA
E-mail address: wangmiaothu@mail.tsinghua.edu.cn

DEPARTMENT OF ELECTRONICS AND INFORMATICS (ETRO), VRIJE UNIVERSITEIT BRUSSEL (VUB)
PLEINLAAN 2, BRUSSELS B-1050, BELGIUM
E-mail address: rzhong@ETROVUB.be

DEPARTMENT OF ELECTRONICS AND INFORMATICS (ETRO), VRIJE UNIVERSITEIT BRUSSEL (VUB)
PLEINLAAN 2, BRUSSELS B-1050, BELGIUM
E-mail address: acmuntea@etrovub.be

RECEIVED SEPTEMBER 26, 2014
ACCEPTED MARCH 11, 2015