Stereoscopizing Cel Animations

Xueting Liu Xiangyu Mao Xuan Yang Linling Zhang Tien-Tsin Wong

The Chinese University of Hong Kong*

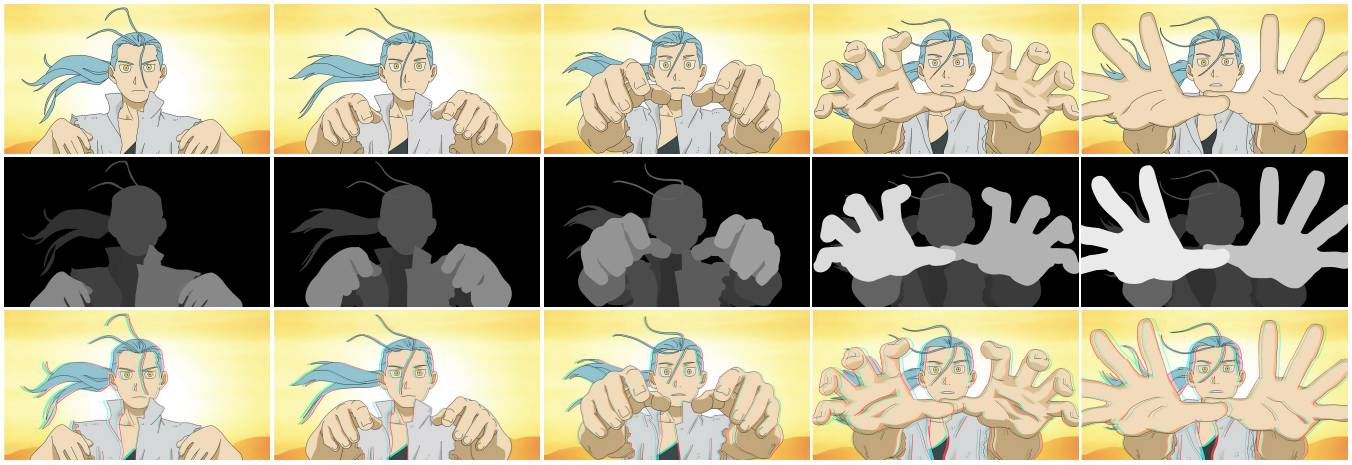

Figure 1: *Stereoscopization of a cel animation. Our method takes an ordinary 2D cel animation (top row) as input, infers the temporal-consistent ordering, and synthesizes the per-frame depth maps (middle row), in order to generate a stereoscopic cel animation (bottom row, presented in the form of anaglyphs). This sequence has 12 frames (*1920×1080*). The frame containing the maximal number of regions has 82 regions. In our experiment, depth ordering takes 12 minutes, and depth synthesis takes 9.6 minutes.*

Abstract

While hand-drawn cel animation is a world-wide popular form of art and entertainment, introducing stereoscopic effect into it remains difficult and costly, due to the lack of physical clues. In this paper, we propose a method to synthesize convincing stereoscopic cel animations from ordinary 2D inputs, without labor-intensive manual depth assignment nor 3D geometry reconstruction. It is mainly automatic due to the need of producing lengthy animation sequences, but with the option of allowing users to adjust or constrain all intermediate results. The system fits nicely into the existing production flow of cel animation. By utilizing the T-junction cue available in cartoons, we first infer the initial, but not reliable, ordering of regions. One of our major contributions is to resolve the temporal inconsistency of ordering by formulating it as a graph-cut problem. However, the resultant ordering remains insufficient for generating convincing stereoscopic effect, as ordering cannot be directly used for depth assignment due to its discontinuous nature. We further propose to synthesize the depth through an optimization process with the ordering formulated as constraints. This is our second major contribution. The optimized result is the spatio-temporally smooth depth for synthesizing stereoscopic effect. Our method has been evaluated on a wide range of cel animations and convincing stereoscopic effect is obtained in all cases.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth cues; J.5 [Computer Application]: Arts and Humanities—Fine arts;

Keywords: Stereopsis, 3D cartoon, and T-junction.

*e-mail:{xtliu,xymao,xyang,llzhang,ttwong}@cse.cuhk.edu.hk

Links: DL PDF

1 Introduction

Traditional cel animation is produced with each frame being drawn manually on celluloids or via computer tablets, and remains a widely used approach (Fig. 2). Unfortunately, it is extremely difficult to introduce stereoscopic effect into cel animations. To our best knowledge, there is only a scarce number of stereoscopic cel animations produced so far. Unlike live-action movies that can be captured with a stereo camera and 3D computer animations that can be computer rendered (e.g. Toy Story and Cyborg 009 [Production I.G. et al. 2012]), hand-drawn cartoons contain no physically correct depth to estimate nor 3D geometrical information to exploit. In fact, frames drawn by cel animators usually contain physically incorrect objects or shapes to maintain aesthetics and style [Rademacher 1999]. Training animators to manually draw stereoscopic pairs of frames is almost infeasible.

As cels may be physically incorrect, 3D geometry reconstruction of the hand-drawn scenes becomes infeasible. Besides, the transition between adjacent frames is typically much larger than that of the live-action videos, thus existing pose estimation and feature track-

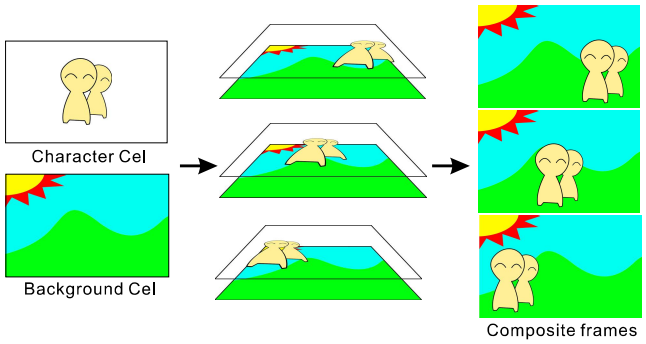

Figure 2: *Traditional production of cel animation. Motion characters are drawn on physical celluloids (or compositing layers in modern digital production) and then composed to produce the animation.*

ing may not be applicable. One may suggest utilizing cels as ordered layers in the animation production. However, such cel layers may not correspond to the physical depth we need. As illustrated in Fig. 2, the two characters may be collapsed into a single cel during the production. No ordering information between them is available in the original cel. Such collapsing treatment is decided by the animator and is common in the real production. Although one can *manually* introduce depth information to still cartoons by labeling [Sỳkora et al. 2010] or modeling geometry [Production I.G. et al. 2012], these approaches are labor-intensive. The situation is even worsened when extended to long animation sequences.

In this paper, we propose a novel method to "*stereoscopize*" (introduce stereoscopic effect to) traditional cel animations by *inferring the pseudo-depth*, that looks convincing and pleasant. While our method is automatic, it provides the option of full user control via a direct adjustment and constraints on all intermediate results. It fits naturally into the existing production flow of the 2D cel animation. Animators do not have to model 3D geometry nor to train themselves for hand-drawing stereo frames. While cartoons may not contain physically correct information, there remain important cues for human audiences to perceive the depth. One important cue is the *T-junction* (Fig. 5) corresponding to the *occlusion*, and has long been aware by psychologists and computer scientists [Bruce et al. 2003; Guzmán 1968] in visual perception. Hence, we propose to first infer the ordering of the layers by exploiting the T-junction cue. Here, the "layers" refer to the regions we extracted and may not be equivalent to the cel layers. While the T-junction cue may not be obvious to detect in natural images, it is especially distinctive in cartoon drawing as regions are mostly enclosed by clear edges. However, the T-junction cue may still be noisy and lead to ordering inconsistency within a frame and/or among the frames. To suppress the noise while allowing the change of ordering due to the actual motion, we formulate the relation between each pair of layers as a graph-cut problem to maintain temporal ordering consistency.

Even with the ordered layers, layers separated by equal distance (due to the lack of inter-layer distance information) and discontinuous depth change across consecutive frames cannot produce visually appealing stereoscopic effect. Instead, layers belonging to the same object should have similar depth values, and the depth should change smoothly across frames. To compute the pseudo-depth with these required properties, we formulate it as an optimization problem, with the computed layer ordering as inequality constraints and the user control as higher-priority constraints. Once the pseudo-depth is computed (Fig. 1, middle row), we can then synthesize a stereo pair for each frame by rendering the layers from novel viewpoints and inpainting the missing pixels due to disocclusion. Our major contribution lies on two aspects. The first one is a novel graph-cut formulation for achieving temporal consistency of ordering. The second is the depth synthesis by minimizing the energy over similar motion and temporal smoothness. On the bottom row of Fig. 1, visually pleasant and convincing stereoscopic effect for the cel animation is obtained, even the character is originally drawn on a single cel layer and provides no physical clues.

2 Related Work

The key to stereoscopize movies or animations is to obtain a depth value for each pixel in each frame, so that the disparity can be computed. The depth is determined by obtaining either image-based depth maps or geometry-based models. Existing methods can be roughly classified into depth inferencing and depth creation.

Depth Inferencing Recovering depth and/or geometry from natural images is a classical problem in computer vision. Existing approaches utilize various photographic depth cues for recovery, including shading [Horn 1990; Wu et al. 2008], texture [Super and Bovik 1995; Forsyth 2001], focus [Nayar and Nakagawa 1990; Assa and Wolf 2007], and even haze [He et al. 2009]. With a live-action video sequence as the input, camera trajectory can be estimated and temporal coherence can be considered [Kang and Szeliski 2004; Zhang et al. 2008; Lang et al. 2010]. Unfortunately, these methods are only applicable for natural images/videos, because hand-drawn cartoons are lack of cues exploited above. For instance, cartoon shading is usually crude and not guaranteed to be physically correct. More seriously, the movement between consecutive frames is usually much larger than that of the live-action videos. Together with the insufficiency of textures in cartoons, feature tracking and pose estimation become very difficult. Hence, depth or geometry recovery from hand-drawn cel animations is infeasible with the above methods.

While most depth cues are not exploitable in cel animations, there remains one common cue available in both natural images and cartoons. It is the T-junction. Its notion has long been aware by researchers in visual perception [Metzger 1936; Guzmán 1968; Bruce et al. 2003]. However, only small number of work is available [Apostoloff and Fitzgibbon 2005; Dimiccoli and Salembier 2009a; Dimiccoli and Salembier 2009b; Amer et al. 2010; Jia et al. 2012] and all of them only focus on natural images. In this paper, we make the first attempt to exploit the T-junction cue within a cartoon frame and the temporal consistency of T-junctions over the whole cel animation to infer the depth. Note that, unlike in natural images where T-junctions are less obvious to detect, the T-junction cue is especially suitable for cartoons due to the availability of clear enclosing edges.

Depth Creation Attempts have also been made to construct 3D geometry from line drawings by making assumptions on both the input drawing and the shape being constructed. Taking CAD or architectural drawings as input and making the parallel-line assumption, methods have been developed to construct objects or architectural structures [Lipson and Shpitalni 1996; Varley and Martin 2002; Lee et al. 2008; Ward et al. 2011]. Due to the strong assumptions, they are not applicable to arbitrary drawings, such as cartoons of "organic" characters or even physically incorrect but stylish drawings. Another stream of work focuses on sketch-based modeling [Igarashi et al. 1999; Gingold et al. 2009; Goldberg 2009; Karpenko and Hughes 2006; Nealen et al. 2007; Joshi and Carr 2008; Kim et al. 2013] that can construct more "organic" objects by making another set of assumptions. Unlike these comprehensive shape construction approaches, we only construct a 2.5D layer representation from the cartoon and make no assumption on the input.

The most straightforward approach to create depth is to allow users

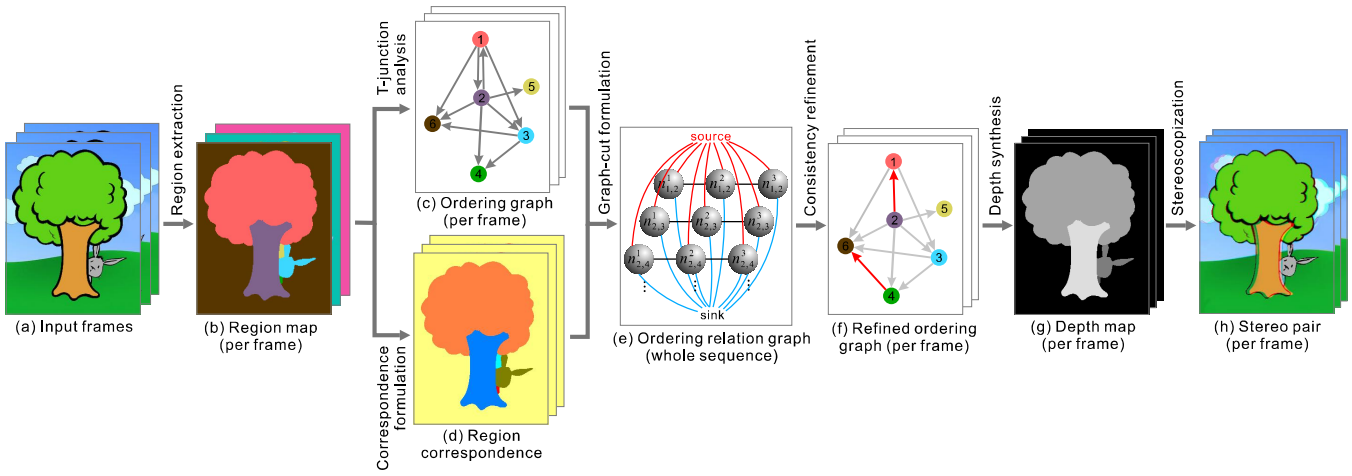

Figure 3: *System overview.*

to assign depth values to pixels directly [Ventura et al. 2009; Schar et al. 2008; Wang et al. 2011]. The amount of user intervention can be reduced by specifying equalities and inequalities [Zhang et al. 2002; Assa and Wolf 2007; Sỳkora et al. 2010]. Nevertheless, existing methods are mainly applied on still cartoons and seldom consider temporal consistency. Due to the amount of manual input, practical application of these methods to lengthy cel animation is questionable. In contrast, our depth-inferencing method is mainly automatic with optional user control, and capable to synthesize temporally coherent depth for the whole animation sequence.

If the original cel layers are available, one can also create a stereoscopic animation by manually assigning depth to each cel layer [Tokyo Movie Shinsha 1977; Production I.G. 2011; Rivers et al. 2010] as practiced in the current film industry. However, content within the same cel is therefore flattened unless it is modeled separately with geometry. In our work, we compute the depth for the regions extracted from a cel, contents in the same cel can also be stereoscopized as demonstrated in Fig. 1.

3 Overview

Our system is overviewed in Fig. 3. Given a sequence of animation frames as input, we first extract the regions from each frame. Our goal is to determine a depth value for each region in order to synthesize stereo frames. Here, each area enclosed by edges forms a region (Fig. 4(e)). Unlike typical segmentations, we do not segment purely based on color, as separated regions caused by shading (e.g. shaded regions in Fig. 4(a)) are inappropriate in our application. Instead, we first identify all edge pixels and store them in an edge map. Note that edges in cartoons are not necessarily black in color. It can be any color but locally darker than that of the neighboring regions. As the luminance channel contains the most visually sensitive content, we convert the RGB frames to Lab color space and process only the L channel. We preprocess the L image by applying the adaptive histogram equalization to make edges more distinctive, followed by the median filtering. The difference between the before and after median-filtered images is the edge map. The median filter can effectively differentiate the explicitly hand-drawn edges (trough-shaped profile Fig. 4(c)) from color discontinuity due to shading (stair-shaped profile in Fig. 4(d)). Fig. 4(b) shows the edge map extracted with Fig. 4(a) as the input. We further extract edge-enclosing regions (Fig. 4(e)) by applying a rolling-ball [Zhang et al. 2009] on this image-based edge map. Note that simple flood-filling may fail as stylish drawings not always form closed regions. Optionally, users are allowed to modify the extracted regions (merge over-segmented or split under-segmented regions) via an interactive

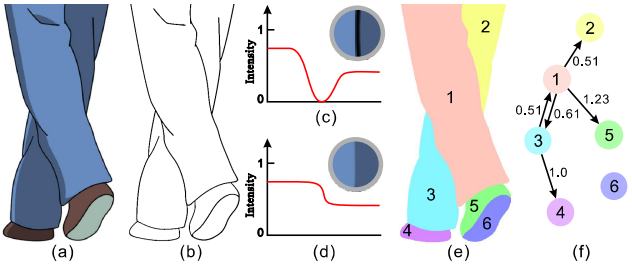

Figure 4: *(a) Input frame. (b) Edge map. (c) Profile of intensity across an explicit edge in the small top-right balloon. (d) Intensity profile across the edge due to shading. (e) Region map. (f) Ordering graph.*

tool.

As one of our key contributions, we then determine the depth-ordering of the extracted regions. We utilize the T-junction cue to resolve the ordering. A T-junction indicates the appearance of occlusion and suggests an ordering relation that one region occludes the other two (e.g. Fig. 5(b)). With the T-junction cue, we construct an ordering graph for each frame independently (Fig. 3(c)). By topologically sorting [Kahn 1962] (Section 4) this ordering graph, we can already determine an ordering. However, sometimes a T-junction is vague or even incorrect (Fig. 5(c)-(e)), thus relying on T-junction cues from a single frame is noisy and insufficient. By exploiting T-junctions from multiple frames, we can suppress noises and maintain temporal consistency. In particular, we formulate this problem as a graph-cut problem [Boykov et al. 2001]. A single ordering relation graph is constructed for the whole animation sequence (Fig. 3(e)) where the ordering relations between every pair of regions are the nodes and inter-frame correspondences among regions (Fig. 3(d)) are the edges. Then temporal-consistent ordering relations can be obtained by solving an optimal cut (Section 5). A nice feature of this graph-cut formulation is that it also allows sharp changes of the orderings due to actual motions. The result of the graph-cut is used to remove inconsistent and/or insert missing orderings in each ordering graph (Fig. 3(f)). The final ordering of regions for each frame is determined using the topological sorting.

To produce convincing stereoscopic effect without discontinuous depth change, depth ordering alone is not sufficient. We need to further compute the *pseudo-depth* (Fig. 3(g)) so that the depth of a region changes *smoothly* over time (Section 6). We formulate it as an optimization problem with an objective to minimize two types of inter-region depth distances. They are the depth distances

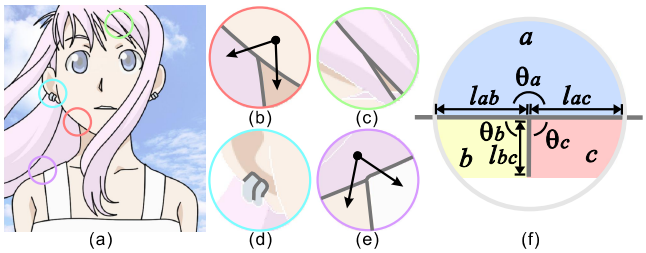

Figure 5: *An example frame in (a) contains T-junctions having correct suggestion (b), vague suggestion (c), as well as incorrect suggestions (d) & (e). (f) Notations used in our formulation.*

between temporally neighboring regions over consecutive frames, and the depth distances between spatially neighboring regions with similar motions (regions with similar motions are more likely to be connected, and so as their depths). Previously estimated ordering information is formulated as the inequality constraints so that the ordering is not violated during energy minimization.

Finally, with the optimized image-based depth maps (partial geometry) and the original color frames, we can synthesize a stereo pair for each frame (Fig. 3(h)) by re-rendering each frame as viewed from two novel eye positions (Section 7). Gaps due to disocclusion in the re-rendered views are inpainted. The reason, that we do not reuse the input frame as one view of the stereo pair, is to avoid large gaps generated by large eye disparity which may complicate the subsequent inpainting.

4 T-Junction for Ordering

When a boundary (the vertical line in letter "T") is blocked by another boundary (the horizontal line in "T"), a T-junction is formed and it suggests the appearance of occlusion (Fig. 5(f)). While the T-junction suggests no ordering information between the two regions sharing the blocked boundary (regions b & c in Fig. 5(f)), it suggests a high belief of the third region (region a in Fig. 5(f)) occluding the other two (b & c). We define a T-junction as a 3-valence junction point in the edge map. An ideal T-junction is formed by three sufficiently long boundaries in which two boundaries are colinear and the third boundary is perpendicular to them. Obviously, short boundaries (Fig. 5(d)) as well as vague T-junction (Fig. 5(c)) may mislead the ordering. Hence, we compute a belief of the occlusion suggestion for each detected T-junction. Consider a T-junction t, we model its belief of suggesting region a blocking regions b and c (denoted as $a \rightarrow b$ and $a \rightarrow c$) as

$$B^t_{a\rightarrow b} = B^t_{a\rightarrow c} = kl\min(\theta_a, 2\pi - \theta_a), \tag{1}$$

where $l = \min(l_{ab}, l_{ac}, l_{bc}, l_o)$, and $k = 1/(\pi l_o)$.

Here, region a is the region with the maximally subtended angle θ_a at the junction. Regions b and c are the two remaining regions. Fig. 5(f) explains the notation. Notation l_{ab} denotes the arc length of the boundary shared by regions a and b. l_{ac} and l_{bc} are defined similarly. l_o is the radius of the circular neighborhood centered at the junction and it is pre-defined (15 pixels in all our experiments, and should be associated with the resolution). It bounds the contribution of the boundary length to the above belief as it is meaningless to consider the whole boundary. Constant k normalizes the belief value to the range of [0,1]. The belief defined above is maximized when the blocking boundary is straight and all boundaries are sufficiently long. When there exists two equal maximally subtended angles at the same junction (Fig. 5(c)), both ordering suggestions are valid and their beliefs are computed separately.

Our current junction identification method is rather straightforward. More sophisticated approach can be found in [Noris et al. 2013].

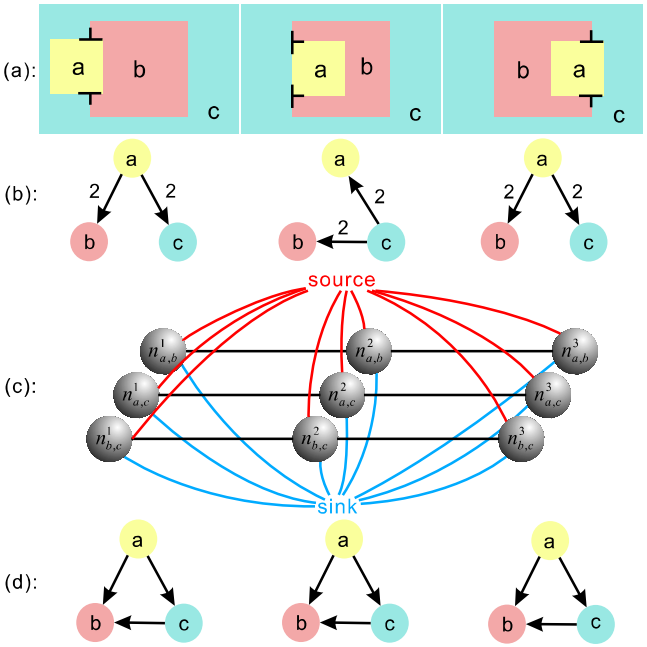

Figure 6: *(a) An input sequence of three frames. (b) The corresponding ordering graphs constructed based on the T-junctions in a single frame. (c) The ordering relation graph for the whole sequence. (d) Refined ordering graphs based on the graph-cut result.*

However, no matter how sophisticated the method is, missing or incorrect T-junctions may still be unavoidable. Instead of relying on the sophistication of T-junction identification and its belief model design, we rely on the aggregate effect of the large number of T-junction suggestions from multiple frames, to suppress the noisiness, compensate the missing T-junctions, and resolve the inconsistency among the ordering suggestions in the following section.

With the ordering suggestions and their beliefs, we construct an ordering graph for each frame (Fig. 4(f)). Each region corresponds to a node in the graph and an ordering suggestion $a \rightarrow b$ corresponds to a directed edge from node a to node b. Each edge carries a weight corresponding to the belief. When there are multiple T-junctions between regions a and b, the beliefs of all T-junctions suggesting the same ordering direction are summed and assigned as the weight of the directed edge from a to b,

$$w^f_{a\rightarrow b} = \sum_{t\in T} B^t_{a\rightarrow b} \tag{2}$$

where T is the set of T-junctions suggesting the same ordering $a \rightarrow b$ in the frame f. It is possible that different T-junctions suggest opposite ordering directions, i.e. both $a \rightarrow b$ and $b \rightarrow a$ exist. These opposite directions correspond to opposite directed edges in the graph (the cycle in Fig. 4(f)). In that case, weights of opposite ordering directions are summed separately.

5 Temporal-Consistent Ordering

It seems that the ordering of all regions in each frame already can be obtained by topologically sorting each ordering graph independently. However, T-junctions in a single frame may be unreliable, contradictory, or temporally inconsistent. Fig. 6(a) shows one unstable scenario in which a yellow square moves from left to right and occludes the pink square behind. Even though the ordering relationship among the three color regions does not change over time, the T-junctions captured are inconsistent with each other (Fig. 6(b)). Hence, we utilize the aggregate effect of a large number of T-junctions from multiple frames in order to maintain the temporal

consistency. While an ordering relation tends to remain unchanged in a period of time, it may also change at certain points due to actual motions. Hence, we formulate this problem as a graph-cut problem which can maintain the temporal consistency while simultaneously allow sharp and persistent change of ordering.

We construct an *ordering relation graph* (Fig. 6(c)) for the whole animation sequence, based on the per-frame ordering graphs constructed previously. In this graph, each node, denoted as $n^f_{a,b}$, stands for an ordering relation between a pair of regions a and b in frame f. Each non-terminal node is connected to the two terminal nodes, source and sink. If the graph-cut result labels a non-terminal node $n^f_{a,b}$ to source, it means region a occludes region b ($a \rightarrow b$) in frame f. Otherwise, region b occludes region a ($b \rightarrow a$).

Data Cost The data cost measures how likely an ordering relation between regions a and b in a frame f is $a \rightarrow b$ (source) or $b \rightarrow a$ (sink). Each node is connected to both the source and the sink, and the costs associating with the correspondingly edges are defined as

$$D(n^f_{a,b}, \text{source}) = w^f_{a\rightarrow b}/(w^f_{a\rightarrow b} + w^f_{b\rightarrow a}) \tag{3}$$

and

$$D(n^f_{a,b}, \text{sink}) = w^f_{b\rightarrow a}/(w^f_{a\rightarrow b} + w^f_{b\rightarrow a}) \tag{4}$$

respectively.

Region Correspondence and Smoothness Cost The smoothness cost measures how likely an ordering relation remains unchanged in two consecutive frames, and is modeled as the similarity of the corresponding regions. To measure the similarity, first we need to determine the correspondence of regions. Unlike live-action videos, content of cel animation usually changes much more rapidly and abruptly. So existing motion tracking methods are generally not applicable. We propose to determine the region correspondence based on the similarity of regions. In particular, the similarity of two regions, a and b, is measured in terms of their differences in color, position, size, and shape as follow,

$$s_{a,b} = \mathcal{J}_{a,b} \max\left(\frac{o_{a,b}}{\min(r_a, r_b)}, \exp\left(-\frac{|r_a - r_b|}{\min(r_a, r_b)} - \frac{|h_a - h_b|}{\min(h_a, h_b)}\right)\right) \tag{5}$$

where

$$\mathcal{J}_{a,b} = H[\mathcal{T}_C - \mathcal{C}_{a,b}]H[\mathcal{T}_\mathcal{N} - \mathcal{N}_{a,b}], \tag{6}$$

H is a Heaviside step function

$$H[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0, \end{cases} \tag{7}$$

and $\mathcal{C}_{a,b} = \| \mathbf{q_a} - \mathbf{q_b} \|$ measures the color difference of a and b by calculating the Euclidean distance between their corresponding color histogram vectors $\mathbf{q_a}$ and $\mathbf{q_b}$. The color histogram vector is constructed in RGB color space with each channel quantized into 16 bins. $\mathcal{N}_{a,b}$ is the smallest Euclidean distance between regions a and b and is normalized by image resolution. $\mathcal{T}_C$ and $\mathcal{T}_\mathcal{N}$ are user-defined thresholds and are set to 0.3 and 0.1 respectively in our experiments. $o_{a,b}$ is the size of the overlapping area of a and b, r_a is the size of a, and h_a is a very crude shape descriptor defined as $h_a = r_a/y_a$ where y_a is the perimeter of a.

For each region in a frame, we search for the most similar region in the previous and subsequent frames respectively and they are referred as the corresponding regions. Note that it is possible that a region has no corresponding region previously or/and subsequently. We design the similarity by taking two forms of region changes into account. On one hand, a region may be divided into multiple sub-regions in neighboring frames due to occlusion. These sub-regions can be very different in shape and size with the original region, but their positions are less likely to change too much. Thus, measuring the overlapping area gives high tolerance for this complication. On the other hand, a region can translate a lot over two consecutive frames, but the shape of the region is less likely to change. In this case, measuring the difference of area size and shape gives higher tolerance. It is less likely to have both severe occlusion and large movement simultaneously, as this may hurt the "readability" of the animation and seems to be avoided by cel animators.

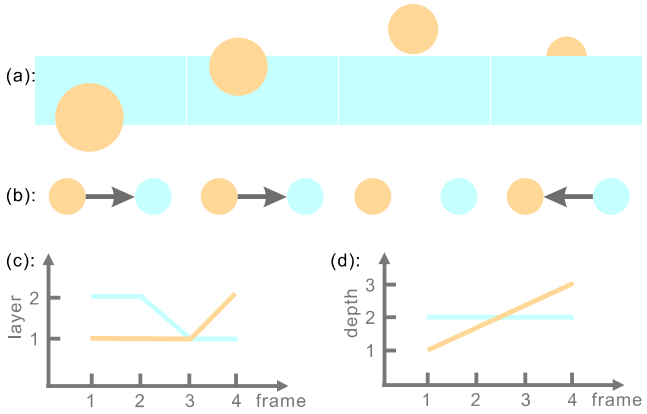

Figure 7: *Direct usage of the ordering for depth assignment. (a) Four input frames. Corresponding regions are labeled with the same colors. (b) Refined ordering graphs. (c) Assigning the layer number as the depth leads to discontinuous motion over time. The colors of the curves correspond to that of the regions. (d) Plot of our optimized depth values over time.*

With the correspondence information, we create an edge to link each two corresponding *relations* (not regions, i.e. nodes in Fig. 6(c)) of regions a and b in consecutive frames f and $f+1$. The associated smoothness cost is modeled as a function of the similarity of a in frames f and $(f+1)$, $s_a^{f,f+1}$, and the similarity of b in frames f and $(f+1)$, $s_b^{f,f+1}$,

$$V(n^f_{a,b}, n^{f+1}_{a,b}) = \beta \min(s_a^{f,f+1}, s_b^{f,f+1}) \tag{8}$$

where β is the user-defined scaling factor. We set it to 0.8 in all our experiments.

Optimization and Consistency Refinement The overall energy function is

$$\sum_{f,u} D(n^f_{a,b}, u)\Phi(u^f_{a,b}, u) + \sum_f V(n^f_{a,b}, n^{f+1}_{a,b})\Phi(u^f_{a,b}, u^{f+1}_{a,b}) \tag{9}$$

where $u \in \{\text{source, sink}\}$ is the label, $u^f_{a,b}$ is the label of $n^f_{a,b}$, and $\Phi(u, v)$ returns 1 if u is different from v and 0 otherwise. Noise-suppressed and temporal-consistent ordering can then be obtained by finding the minimum cut of the graph that minimizes the above energy function. With the graph-cut result, we can then refine the per-frame ordering graphs by adding missing edges and removing inconsistent edges (Fig. 6(d)). Followed by the topological sorting, we obtain the ordering of all regions in all frames.

6 Depth Synthesis

Given the ordered regions as layers, the simplest way to create depth is to assign each layer with a distinct depth value, with closer layers having smaller depth values. A natural assumption is that the inter-layer distance is constant (Fig. 8(c)). But obviously, such depth assignment cannot produce convincing and temporal-coherent stereoscopic effect. Fig. 7 explains why such simple approach cannot maintain temporal coherence. Consider a tan-colored ball is thrown over a fixed blue wall (Fig. 7(a)). The ordering graph of each frame after refinement is shown in Fig. 7(b). By assigning the layer ordering as the depth without considering temporal coherence, it is possible to obtain depth values for the tan and blue layers over the four frames as plotted in Fig. 7(c). Fig. 12(b)

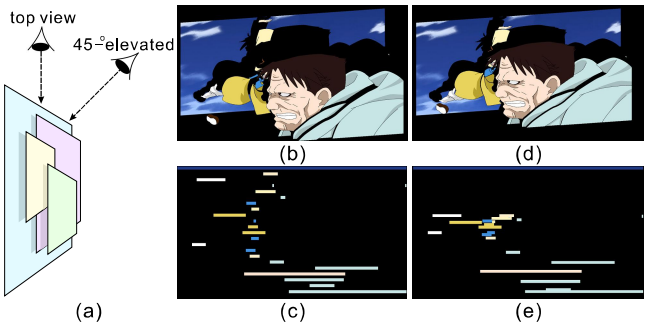

Figure 8: *Depth of a real example created by assigning layer number as depth ((b) & (c)) and our depth optimization ((d) & (e)). (a) Configuration of the two cameras used for rendering the layers. (b) & (d) are rendered from the 45°-elevated camera. (c) & (e) are rendered from the top-view camera. Note the difference in the distribution of layers along the z direction.*

shows a real example in which the depth of old man changes abruptly over the frames.

Temporal-Coherent Depth Synthesis We formulate the depth synthesis as an optimization. Before describing the objective function in detail, we first identify the requirements for a smooth depth motion. Firstly, the same regions should not change in depth abruptly over time. That is, the depth distances of temporally neighboring regions should be minimized. Secondly, regions belonging to the same object should be close to each other in depth. However, determining whether regions belonging to a single object requires a sophisticated semantic analysis. Instead, by assuming that regions belonging to the same object are more likely to move synchronously, we can simplify the problem to minimizing the depth distances between regions with similar motions. In addition, the previously computed depth ordering should be preserved in the form of constraints. Thus, we formulate the depth synthesis as a minimization of the following energy function,

$$E_s + E_t \tag{10}$$

where the term E_s corresponds to the similar motion requirement, and E_t corresponds to the temporal smoothness. We define the similar motion term E_s as,

$$E_s = \sum_{f,a,b} \exp\left(-\delta_s \parallel \mathbf{m_a^f} - \mathbf{m_b^f} \parallel\right)(d_a^f - d_b^f)^2 \tag{11}$$

which minimizes depth differences between spatially neighboring regions with similar motions. Here, d_a^f denotes the depth of region a in frame f, and it is the value to be determined during the minimization. $\mathbf{m_a^f}$ is the motion vector of region a. It is a vector (normalized to the canvas size) from the centroid of a in frame f to the centroid of a in frame $f+1$. δ_s is a scaling factor and set to 0.1 in all our experiments.

The temporal smoothness term is defined as

$$E_t = \sum_{f,a} s_a^{f,f+1}(d_a^f - d_a^{f+1})^2 + \lambda \sum_{f,a} e_\zeta(d_a^f, \zeta)^2 \tag{12}$$

It minimizes depth differences between corresponding regions between consecutive frames. The left part controls the tolerance of depth change of a region. $s_a^{f,f+1}$ is the similarity of a defined in Section 5. Our rationale is that a region having a smaller change in shape is less likely to have large depth change. Hence, the blue wall in Fig. 7 is less likely to change in depth while the moving ball (with a change of scale) is more acceptable to have a larger change in depth. The right part in Eq.(12) aims at obtaining a smooth depth change over time guided by a fitted curve. That is, we hope the depth d_a^f of region a over time f can be represented by a fitted quadratic curve ζ. Its fitting error $e_\zeta(d_a^f, \zeta)$ is what we are minimizing. λ is the weight and is set to 100 in all our experiments.

The previously obtained depth ordering is formulated as the following linear constraints

$$d_b^f - d_a^f \geq g_{a\to b}^f, \qquad \forall w_{a\to b}^f > 0 \tag{13}$$

The above constraint is only applied to any two regions with an ordering relation. Function $g_{a\to b}^f$ controls the minimal depth difference between regions a and b in frame f. It tries to pull regions apart in the depth domain and provide the variety of inter-layer depth distance. If a T-junction has a higher belief value, it is more confident that the corresponding regions differ in depth. If the ordering relation between two regions are more persistent over a period of time, the depth difference should be more observable. Moreover, regions with larger sizes contribute more to the overall visual experience. This gives rise to the following design,

$$g_{a\to b}^f = \min(r_a^f, r_b^f) o_{a\to b}^f w_{a\to b}^f \tag{14}$$

where

$$o_{a\to b}^f = \min_{f'} |f' - f| \;\; s.t. \;\; w_{b\to a}^{f'} > 0 \tag{15}$$

measures the number of frames to the nearest swapping point (when $a \to b$ becomes $b \to a$), r_a^f is the size of a in frame f, and $w_{a\to b}^f$ is the belief of the T-junction.

This optimization can be solved using standard methods such as active set [Gill et al. 1984]. It terminates when the energy converges. After minimization, each region is assigned with an optimized depth value. Fig. 12(c) shows the per-frame depth maps obtained with this approach. Our result effectively suppresses the abrupt depth change in Fig. 12(b). Note that the optimized depth values are relative, and can be scaled as needed.

User Constraint We allow users to specify constraints to the depth synthesis by adding new ordering suggestions or directly assign depth values to the regions. Whenever the user introduces a new ordering suggestion, we add an additional inequality constraint to Eq. 13 and remove any contradicting constraints immediately. Whenever the user directly assigns the depth value of a region a in frame f, we add a new constraint to the optimization process as

$$d_0 - \epsilon \leq d_a^f \leq d_0 + \epsilon \tag{16}$$

where d_0 is the user-assigned depth value, and ϵ is a small tolerance.

Background Region Special treatment is needed to deal with the background region. Even with T-junctions, the ordering between the background and other regions is not informative. Currently, our system identifies the background region by heuristics, e.g. regions that are large in size and regions having frequent contact to the frame boundary are regarded as background. Of course, users can also refine the identification of the background via a simple interactive tool. Once a region is identified as the background, it is assigned with an infinite depth.

7 Stereoscopization

With the per-frame depth maps (partial geometry), we can synthesize a stereo pair of images for each frame by rendering the corresponding depth map (textured with the input color frame) from novel viewpoints. Instead of regarding the original frame as one of the two views, we render two novel views by equally translating the viewpoint to the left and to the right. This strategy reduces the amount of disoccluded pixels to fill. Fig. 9(a) and 10(a) are two blow-ups of the re-rendered images with disoccluded pixels colored in green.

Figure 9: *(a) A blow-up of a left view image from the sequence in Fig. 13. (b) Optimized depth map with gaps. (c) Region map with gaps. (d) Regions are extended to fill the disocclusion pixels. (e) The final inpainted result.*

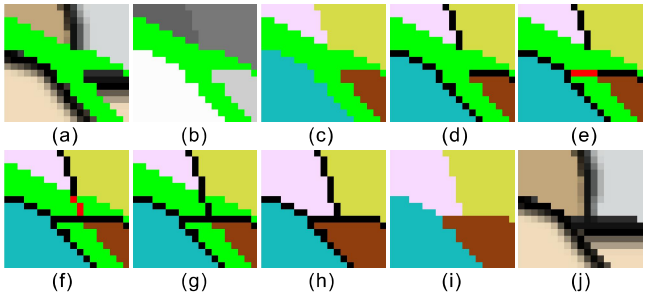

Figure 10: *(a) A blow-up of a right view image from the sequence in Fig. 1. (b) Optimized depth map. (c) Region map with gaps. (d) Region map dressed with boundaries. (e) Boundary of the closer region is first extended. (f) Followed by the farther region. (g) The final extended boundaries. (h) Extended regions with boundaries. (i) Extended regions only. (j) The final inpainted result.*

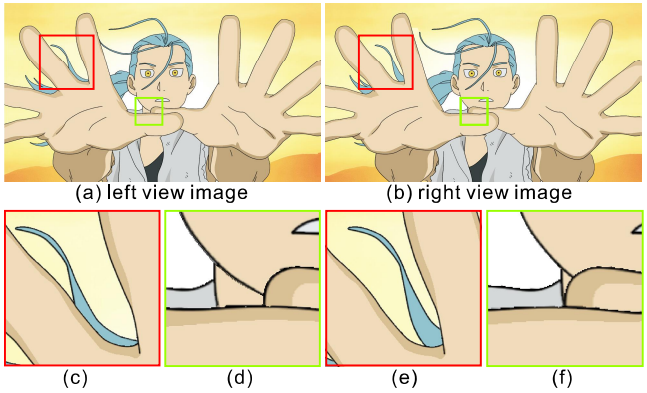

Figure 11: *(a) & (b) are the final inpainted stereo pair. (c) & (d) are blow-ups of (a). (e) & (f) are blow-ups of (b).*

We then inpaint the disoccluded pixels by extending the regions being occluded. Fig. 9 demonstrates a simple scenario in which only one region is disoccluded. Note that there can be multiple regions being disoccluded (Fig. 10). Hence, before the actual inpainting, we need to first identify which region a disoccluded pixel belongs to. The basic idea is to extend each region in a front-to-back order. With the previously synthesized depth map, we first pick the closest region with the boundary broken by the disoccluded pixels (Fig. 10(e)), and then extend it along its tangent near its end, until the extension is blocked by another region. Then the next closest region is selected and performed with the boundary extension similarly (Fig. 10(f)). The process continues until all broken boundaries are extended (Fig. 10(h)). The result is an extended region map (Fig. 10(i)). Finally, for each region, we inpaint its disoccluded pixels using texture synthesis method [Ashikhmin 2001]. More sophisticated inpainting techniques [Criminisi et al. 2004; Sun et al. 2005], such as structure-based inpainting, may also be employed for large disocclusion. Fig. 9(e) and 10(j) show the inpainted results. Fig. 11(a) & (b) show a stereo pair of one frame. Note how the regions are properly occluded or disoccluded in the blow-ups of the two views (Fig. 11(c)-(f)).

8 Results and Discussion

To validate the effectiveness of our method, we stereoscopize a wide variety of cel animations, ranging from Japanese to Western styles of drawing, and from single-character to hundreds of characters animated sequences. Fig. 1 , 12 and 15 are Japanese-style animations while the ones in Fig. 13 and 14 are in more Western style. Readers are referred to the supplementary video for visualizing the stereoscopic effect of the examples shown in this paper.

Fig. 1 shows a character stretching out his hands. In traditional cel animation production, his hands, head, and body are very likely to be collapsed into a single cel. Existing approaches to create stereoscopic effect have to manually separate this single character into multiple layers in order to manually assign depth to each layer. Obviously, this is tedious. In contrast, our method automatically generates multiple regions and synthesizes the depth for stereoscopization. Fig. 13 further demonstrates the strength of our method. There are hundreds of regions in this example, making the manual depth assignment and the maintenance of temporal coherence very labor-intensive. Instead, we can conveniently stereoscopize the animation with temporal consistency. As our method purely relies on the T-junction cue of edges, color information is not required during stereoscopization. Fig. 14 shows one interesting example, in which the input animation contains only line drawings. Even with this, we can introduce stereoscopic effect into the animation. Fig. 15 demonstrates the effectiveness of our graph-cut based depth ordering estimation in handling the sudden change of ordering in the sequence. Here, the girl does a crossover. With the synthesized depth maps, we can further introduce out-of-focus effect into the sequence (Fig. 12(e)). In our above experiments, the manual intervention is minimal. For those frames requiring adjustment, each frame only requires less than one minute of user intervention. The same set of parameter values are applied in all our experiments.

Reliability of T-junctions To validate how effective T-junctions are in suggesting ordering, we compare the ordering estimated by our method to the ground truth ordering. We first prepared ground truth orderings by manually labeling the ordering of every pair of regions in each frame. So that we can compute the correct ratio. On average, each frame contains about 213 T-junctions (maximum 454). Table 1 shows the correct ratio statistics at different stages of our system. Even if the ordering is estimated only based on individual T-junctions, the correct ratio is already around 68%-83%. With the simple belief computation (Eq.(2)), the correct ratio is significantly improved. After the graph-cut based temporal consistent ordering computation, the ratio is further raised to 85%-98%.

Correct Ratio	Fig. 1	Fig. 12	Fig. 13	Fig. 14	Fig. 15
T-junction alone	77.18%	74.70%	82.88%	70.73%	68.83%
With intra-frame information	90.17%	90.36%	94.09%	80.57%	81.75%
With temporal coherence	96.81%	94.14%	97.29%	85.60%	96.55%

Table 1: *Reliability of T-junctions in each stage of our ordering determination.*

Timing statistics All our experiments are conducted on PC with 3GHz CPU, 4 GB system memory. The total computational time for each sequence is reported in the corresponding caption. Currently, the whole system is implemented with Matlab. No GPU is used. We believe GPU implementation can significantly boost the system performance.

Limitation One of our limitations lies in the cardboard-like representation where regions are basically assumed to be flat. Hence, we

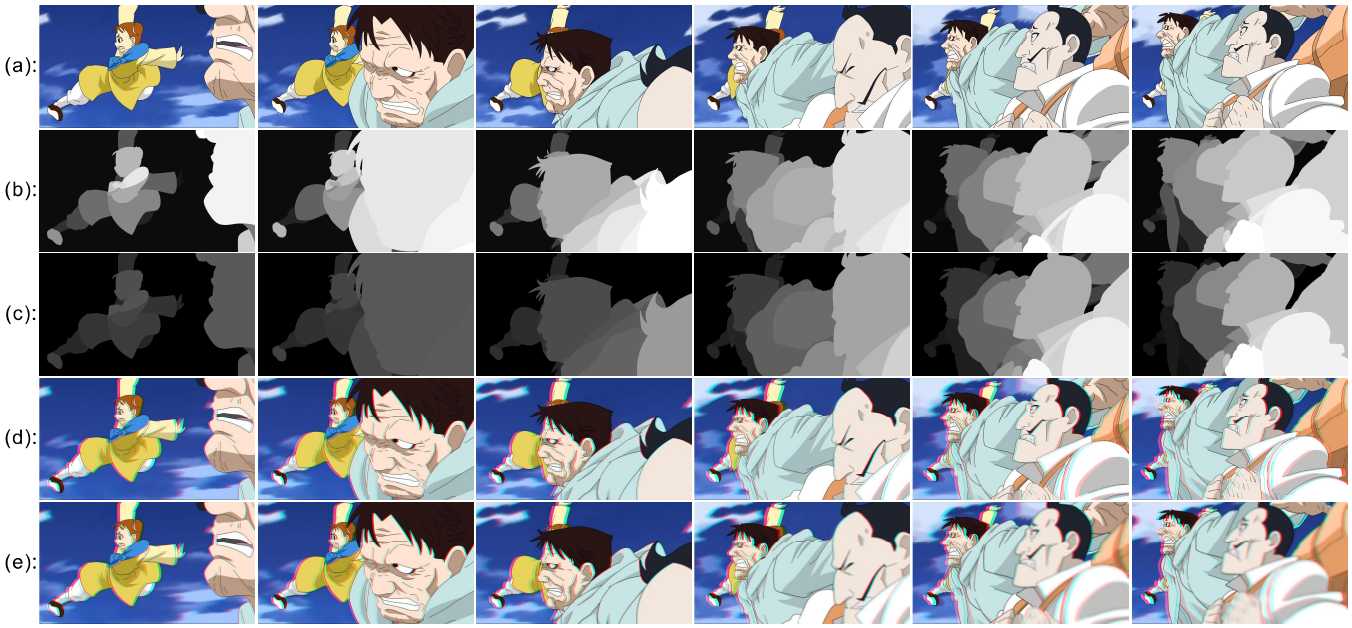

Figure 12: *"Running." (a) Input frames. (b) Topologically sorted ordering graph visualized as intensity. (c) Depth maps. (d) Stereo result. (e) Stereo result with out-of-focus effect. This sequence has 12 frames* (1920×1080)*. The frame containing the maximal number of regions has 72 regions. Depth ordering takes 13.5 minutes, and depth synthesis takes 10.9 minutes.*

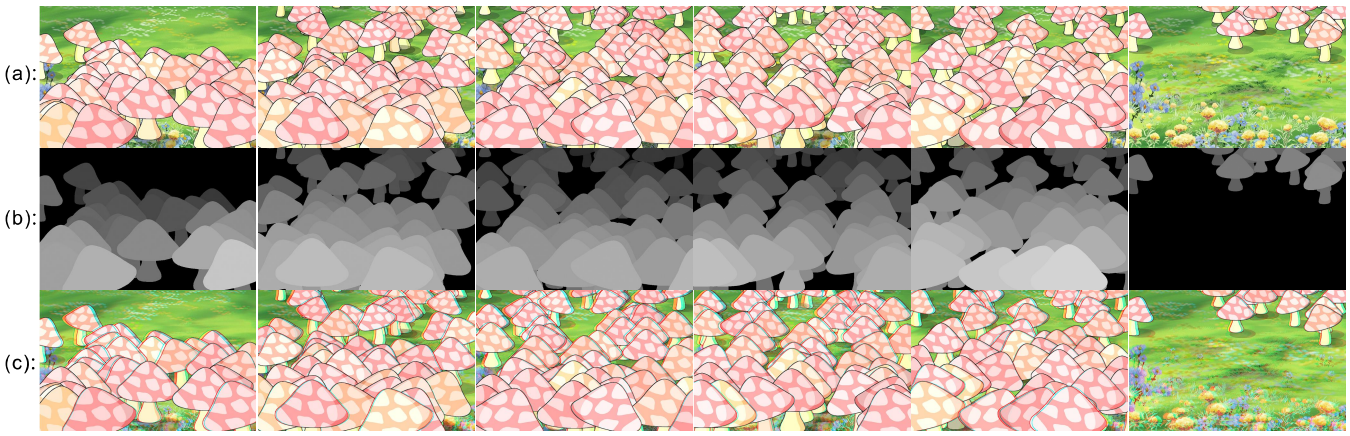

Figure 13: *"The mushrooms." (a) Input frames. (b) Depth maps. (c) Stereo result. This sequence has 80 frames* (994×728)*. The frame containing the maximal number of regions has 124 regions. Depth ordering takes 39.1 minutes, and depth synthesis takes 65.3 minutes.*

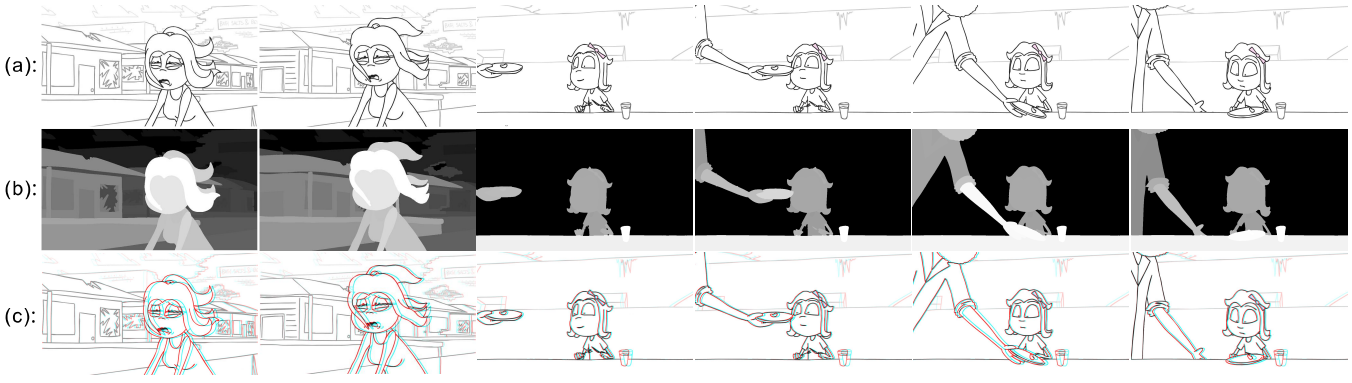

Figure 14: *"Raised by Zombies." (a) Input frames. (b) Depth maps. (c) Stereo result. This sequence has 52 frames* (1280×720) *presented in the form of line drawing. The frame containing the maximal number of regions has 188 regions. Depth ordering takes 23 minutes, and depth synthesis takes 40.2 minutes.* © *Guy Collins.*

Figure 15: *"The basketball girl." (a) Input frames. (b) Depth maps. (c) Stereo result. This sequence has 15 frames* (1104×622)*. The frame containing the maximal number of regions has 92 regions. Depth ordering takes 5.4 minutes, and depth synthesis takes 5.1 minutes.*

cannot synthesize curved regions (curving towards the viewpoint) with gradually changing depth values. It is possible to "inflate" the regions to create pseudo-3D meshes so that regions are connected with each other in the boundary. Besides, in some cases, a region a can be partly occluding another region b and simultaneously a is partly occluded by b (see the head and collar of the running old man in the third frame in Fig. 12). Currently, we cannot compute the depth correctly. Furthermore, our assumption of similar motion suggesting the same object may fail. Unless with sophisticated semantic analysis, such problem has to be resolved by user intervention.

As our method highly relies on the proper identification of regions, cartoons without clear boundary lines (blurry images, images with smoke or explosive effects) may not generate correct result. Another limitation is that our result may fail to resolve the ordering if T-junctions in the sequence consistently suggest an incorrect ordering. Currently, we can only correct this by hand.

9 Conclusions

In this paper, we present a novel method to stereoscopize 2D cel animations. The proposed method relies only on the T-junction cue to resolve the ordering of regions. It fits naturally into the existing production of cel animations. The cel animation can be produced as usual, with an additional last step of our stereoscopization. Our high degree of automation frees users from the labor-intensive segmentation and depth assignment.

Our first key contribution is to maintain the temporal consistency of ordering relationship across the frames, via a graph-cut formulation. Our second contribution is the temporal-coherent depth synthesis via a novel optimization formulation. Convincing stereoscopic effect is created in all our examples.

While our current method only relies on T-junctions, other depth cues like crude shading could also be exploited in the future. We may also "inflate" regions to avoid the cardboarding gaps. Besides, we shall further investigate the feasibility in deducing the grouping of regions in a more semantic fashion.

Acknowledgements

This project is supported by Hong Kong RGC General Research Fund (Project No. CUHK417411), NSFC 2012 (Project No. 61272293) , Basic Research for 2012 Shenzhen Municipal Science and Technology Programme (Project No. JCYJ20120619152326448), and CUHK SHIAE Fund (Project No: 8115034).

References

AMER, M., RAICH, R., AND TODOROVIC, S. 2010. Monocular extraction of 2.1 d sketch. In *Proc. of the International Conference on Image Processing*.

APOSTOLOFF, N., AND FITZGIBBON, A. 2005. Learning spatiotemporal T-junctions for occlusion detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, IEEE, 553–559.

ASHIKHMIN, M. 2001. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM, 217–226.

ASSA, J., AND WOLF, L. 2007. Diorama construction from a single image. In *Computer Graphics Forum*, vol. 26, Wiley Online Library, 599–608.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 11, 1222–1239.

BRUCE, V., GREEN, P. R., AND GEORGESON, M. A. 2003. *Visual Perception: Physiology, Psychology, & Ecology*. Psychology Press.

CRIMINISI, A., PÉREZ, P., AND TOYAMA, K. 2004. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on 13*, 9, 1200–1212.

DIMICCOLI, M., AND SALEMBIER, P. 2009. Exploiting t-junctions for depth segregation in single images. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, IEEE, 1229–1232.

DIMICCOLI, M., AND SALEMBIER, P. 2009. Hierarchical region-based representation for segmentation and filtering with depth in single images. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, IEEE, 3533–3536.

FORSYTH, D. 2001. Shape from texture and integrability. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, IEEE, 447–452.

GILL, P., MURRAY, W., SAUNDERS, M., AND WRIGHT, M. 1984. Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Transactions on Mathematical Software (TOMS) 10*, 3, 282–298.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2d-to-3d modeling. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 148.

GOLDBERG, E. 2009. Medial axis techniques for stereoscopic extraction. In *SIGGRAPH 2009: Talks*, ACM, 74.

GUZMÁN, A. 1968. Decomposition of a visual scene into three-dimensional bodies. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), 291–304.

HE, K., SUN, J., AND TANG, X. 2009. Single image haze removal using dark channel prior. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 1956–1963.

HORN, B. 1990. Height and gradient from shading. *International journal of computer vision 5*, 1, 37–75.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 409–416.

JIA, Z., GALLAGHER, A., CHANG, Y., AND CHEN, T. 2012. A learning-based framework for depth ordering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 294–301.

JOSHI, P., AND CARR, N. 2008. Repoussé: Automatic inflation of 2D artwork. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, The Eurographics Association, 49–55.

KAHN, A. 1962. Topological sorting of large networks. *Communications of the ACM 5*, 11, 558–562.

KANG, S., AND SZELISKI, R. 2004. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision 58*, 2, 139–163.

KARPENKO, O., AND HUGHES, J. 2006. SmoothSketch: 3D free-form shapes from complex sketches. In *ACM Transactions on Graphics (TOG)*, vol. 25, ACM, 589–598.

KIM, Y., WINNEMÖLLER, H., AND LEE, S. 2013. Wysiwyg stereo painting. In *Proceedings of ACM Symposium on Interactive 3D Graphics and Games 2013*.

LANG, M., HORNUNG, A., WANG, O., POULAKOS, S., SMOLIC, A., AND GROSS, M. 2010. Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG) 29*, 4, 75.

LEE, S., FENG, D., AND GOOCH, B. 2008. Automatic construction of 3d models from architectural line drawings. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, ACM, 123–130.

LIPSON, H., AND SHPITALNI, M. 1996. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design 28*, 8, 651–663.

METZGER, W. 1936. *Gesetze des sehens*. W. Kramer.

NAYAR, S., AND NAKAGAWA, Y. 1990. Shape from focus: An effective approach for rough surfaces. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, IEEE, 218–225.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: designing freeform surfaces with 3D curves. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 41.

NORIS, G., HORNUNG, A., SUMNER, R. W., SIMMONS, M., AND GROSS, M. 2013. Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics (TOG) 32*, 1, 4.

PRODUCTION I.G., SANZIGEN ANIMATION STUDIO, AND ISHIMORI PRODUCTIONS, 2012. Cyborg 009.

PRODUCTION I.G., 2011. Ghost in the shell: Stand alone complex: Solid state society.

RADEMACHER, P. 1999. View-dependent geometry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 439–446.

RIVERS, A., IGARASHI, T., AND DURAND, F. 2010. 2.5D cartoon models. *ACM Transactions on Graphics (TOG) 29*, 4, 59.

SCHAR, S., BIERI, H., KILLER, T., AND JIANG, X. 2008. Introducing stereo effects into cel animations. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008*, IEEE, 353–356.

SUN, J., YUAN, L., JIA, J., AND SHUM, H. 2005. Image completion with structure propagation. *ACM Transactions on Graphics (ToG) 24*, 3, 861–868.

SUPER, B., AND BOVIK, A. 1995. Shape from texture using local spectral moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 17*, 4, 333–343.

SỲKORA, D., SEDLACEK, D., JINCHAO, S., DINGLIANA, J., AND COLLINS, S. 2010. Adding depth to cartoons using sparse depth (in) equalities. In *Computer Graphics Forum*, vol. 29, Wiley Online Library, 615–623.

TOKYO MOVIE SHINSHA, 1977. Ie naki ko.

VARLEY, P., AND MARTIN, R. 2002. Estimating depth from line drawing. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, ACM, 180–191.

VENTURA, J., DIVERDI, S., AND HÖLLERER, T. 2009. A sketch-based interface for photo pop-up. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ACM, 21–28.

WANG, O., LANG, M., FREI, M., HORNUNG, A., SMOLIC, A., AND GROSS, M. 2011. Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ACM, 47–54.

WARD, B., KANG, S. B., AND BENNETT, E. P. 2011. Depth director: A system for adding depth to movies. *Computer Graphics and Applications, IEEE 31*, 1, 36–48.

WU, T., SUN, J., TANG, C., AND SHUM, H. 2008. Interactive normal reconstruction from a single image. *ACM Transactions on Graphics (TOG) 27*, 5, 119.

ZHANG, L., DUGAS-PHOCION, G., SAMSON, J., AND SEITZ, S. 2002. Single-view modeling of free-form scenes. *The Journal of Visualization and Computer Animation 13*, 4, 225–235.

ZHANG, G., JIA, J., WONG, T., AND BAO, H. 2008. Recovering consistent video depth maps via bundle optimization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 1–8.

ZHANG, S., CHEN, T., ZHANG, Y., HU, S., AND MARTIN, R. 2009. Vectorizing cartoon animations. *Visualization and Computer Graphics, IEEE Transactions on 15*, 4, 618–629.