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.

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-
ing may not be applicable. One may suggest utilizing cels as or-

dation. Animators do not have to model 3D geometry nor to train

The key to stereoscopicize movies or animations is to obtain a depth

2 Related Work

s non-exploitable in cel animations, there remains one common cue available in both natural images and car-
toons. It is the T-junction. Its notion has long been aware by re-
searchers in visual perception [Metzger 1936; Guzmán 1968; 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 discontinu-
ous depth change across consecutive frames cannot produce visu-
ally 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 prob-
lem, 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 view-
points and inpainting the missing pixels due to disocclusion. Our
major contribution lies on two aspects. The first one is a novel

Figure 2: Traditional production of cel animation. Motion char-
acters are drawn on physical celluloids (or compositing layers in
modern digital production) and then composited to produce the ani-
mation. Character Cel Background Cel Composite frames

graph-cut formulation for achieving temporal consistency of order-
ing. 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.

Depth Inferencing

The most straightforward approach to create depth is to allow users
to freely specify the depth of each pixel by drawing or specifying
the pseudo-depth value. Another approach is to infer or create
depth from images using a variety of cues and methods. In this paper,
we propose a novel method to exploit the T-junction cue.

Depth Creation

Attempts have also been made to create 3D geometry from line drawings by making assumptions on both the
input drawing and the shape being constructed. Taking CAD or ar-
chitectural drawings as input and making the parallel-line assump-
tions, 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 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 en-
closing edges.
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; Sýkora 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(a)). 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 stylistic 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 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...
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 → b and a → c) as

$$B_{a ightarrow b}^t = B_{a ightarrow c}^t = kl \min(\theta_a, 2\pi - \theta_a),$$

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

Here, region a is the region with the maximally subtended angle \(\theta_a\) 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. \(l_a\) 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].
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 \( n_{a,b}^f \), 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 \( n_{a,b}^f \) 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

\[
D(n_{a,b}^f, \text{source}) = w_{a\rightarrow b}^f / (w_{a\rightarrow b}^f + w_{b\rightarrow a}^f)
\]

and

\[
D(n_{a,b}^f, \text{sink}) = w_{b\rightarrow a}^f / (w_{a\rightarrow b}^f + w_{b\rightarrow a}^f)
\]

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,

\[
s_{a,b} = \mathcal{J}_{a,b} \max \left( \frac{a_{a,b}}{\min(r_a, r_b)}, \exp \left( -\frac{|r_a - r_b|}{\min(r_a, r_b)} - \frac{|h_a - h_b|}{\min(h_a, h_b)} \right) \right)
\]

where

\[
\mathcal{J}_{a,b} = H[\mathcal{T}_C - C_{a,b}]H[\mathcal{T}_N - N_{a,b}],
\]

\( H \) is a Heaviside step function

\[
H[x] = \left\{ \begin{array}{ll} 0, & n < 0 \\ 1, & n \geq 0 \end{array} \right.
\]

and \( C_{a,b} = || q_a - q_b || \) measures the color difference of \( a \) and \( b \) by calculating the Euclidean distance between their corresponding color histogram vectors \( q_a \) and \( q_b \). The color histogram vector is constructed in RGB color space with each channel quantized into 16 bins. \( N_{a,b} \) is the smallest Euclidean distance between regions \( a \) and \( b \) and is normalized by image resolution. \( \mathcal{T}_C \) and \( \mathcal{T}_N \) are user-defined thresholds and are set to 0.3 and 0.1 respectively in our experiments. \( a_{a,b} \) 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.

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 \( f+1 \), \( s_{a,b}^{f,f+1} \), and the similarity of \( b \) in frames \( f \) and \( f+1 \), \( s_{b,b}^{f,f+1} \),

\[
V(n_{a,b}^f, n_{a,b}^{f+1}) = \beta \min(s_{a,b}^{f,f+1}, s_{b,b}^{f,f+1})
\]

where \( \beta \) 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

\[
\sum_{f,u} D(n_{a,b}^f, u)\Phi(u_{a,b}^f, u) + \sum_{f} V(n_{a,b}^f, n_{a,b}^{f+1})\Phi(u_{a,b}^f, u_{a,b}^{f+1})
\]

where \( u \in \{ \text{source, sink} \} \) is the label, \( u_{a,b}^f \) is the label of \( n_{a,b}^f \), 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 tangelo-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)
It minimizes depth differences between corresponding regions between all our experiments.

the centroid of depth change of a region. (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 over the frames.

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 in frame f + 1, we add a new constraint to the optimization process as

$$d_{a} - d_{b} \leq d_{a} - d_{b} \leq d_{a} + \epsilon$$

where $d_0$ is the user-assigned depth value, and $\epsilon$ 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 Stereocapization

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.
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 stereoscopicization. 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 stereoscopicization. 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%.

<table>
<thead>
<tr>
<th>Correct Ratio</th>
<th>Fig. 1</th>
<th>Fig. 12</th>
<th>Fig. 13</th>
<th>Fig. 14</th>
<th>Fig. 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-junction alone</td>
<td>77.18%</td>
<td>74.70%</td>
<td>82.88%</td>
<td>70.73%</td>
<td>68.83%</td>
</tr>
<tr>
<td>With intra-frame information</td>
<td>90.17%</td>
<td>90.36%</td>
<td>94.09%</td>
<td>80.57%</td>
<td>81.75%</td>
</tr>
<tr>
<td>With temporal coherence</td>
<td>96.81%</td>
<td>94.14%</td>
<td>97.29%</td>
<td>88.60%</td>
<td>96.55%</td>
</tr>
</tbody>
</table>

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.
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 cel animation can be produced 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 stereocozipation. 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


