High Dynamic Environment Sequence Sampling

Ying Yang[1], Changguo Yu[2], Liang Wan[2,*], Wei Feng[1]
[1]School of Computer Science & Technology, Tianjin University
[2] School of Computer Software, Tianjin University

Abstract

In this paper, we propose a novel method for high dynamic environment sequence sampling. Without future frame information, our approach achieves temporal coherence by mapping sampling information from previous frame to current frame based on temporal superpixels. However, temporal superpixels have unbalanced importance for rendering, although light regions in environment sequences can be tracked across frames. To solve this issue, we develop an adaptive merge-and-split scheme to adjust sample segments to achieve a more importance-balanced sample distribution. Compared with several state-of-the-art methods, our approach gets consistently improved rendering quality across environment sequences. Experiment results also show decent temporal coherence of our approach in sequence rendering.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Environment sequence sampling, temporal superpixels, adaptive merge-and-split

1 Introduction

To enhance rendering realism of 3D virtual scenes, a widely used way is applying HDR environment maps as light sources. When we have rendering computation, the costly integral of lighting, visibility and BRDF functions need to be faced. Using high-resolution environment maps imposes an urgent need to simplify this process. Environment sampling, which approximates real-world illumination with a few directional lights, is a powerful technique to decrease computational burden. Moreover, it can save storage, making dynamic environment sequence illumination practicable.

Previously, different environment sampling techniques [Debevec 2006] [Agarwal et al. 2003][Clarberg et al. 2005] have been proposed for single environment map or dynamic environment sequence. Many static sampling methods [Debevec 2006] [Agarwal et al. 2003] decompose environment maps according to some importance metrics. Directly applying static sampling methods to each frame of environment sequences may lead to low coherence among sample patterns and generate flickering artifacts. In environment sequence sampling, the temporal coherence is a key point to consider. For instance, [Wan et al. 2011] deals with the environment sequence as a whole to generate samples. Because of considering all frames simultaneously, the generated samples maintain good temporal consistency within a sequence. However since this method generates light samples with regular shapes, the rendering results may not always hold high quality. Additionally, it is not applicable to the sampling of real-time captured environment video.

[*]Corresponding author, e-mail: lwan@tju.edu.cn

In this paper, we propose a novel sampling method, just taking current frame and past frame in account to assign coherent light samples. Our method exploits the temporal slic [Chang et al. 2013], which is a real-time video segmentation method, to obtain temporally coherent superpixels of the current frame related with the past frame. However, temporal superpixels have unbalanced importance for rendering, although light regions in environment sequences can be tracked across frames. We then develop an adaptive merge-and-split scheme to adjust segments for a more importance-balanced sample distribution. Experimental results demonstrate that our approach achieves comparable temporal consistency to existing environment sampling methods, while yielding consistently better rendering quality.

2 Related work

2.1 Environment Map Sampling

To compute reflected radiance on an object model in a 3D scene, we need to compute an integration of incident lighting, the visibility function and the BRDF function of a model's surface [Rubinstein 2009]. A high-resolution environment map or an environment sequence undoubtedly brings a large integration computation for rendering. Environment map sampling techniques [Gibson and Murta 2000][Kollig and Keller 2003] have been proposed to greatly reduce computational burden by approximating an environment map with a limited set of directional light sources, most of which are distributed in important light areas.

Existing environment sampling methods are principally designed for sampling single environment map or an environment sequence. Several static sampling methods decompose an environment map into a set of regions according to some importance metrics and place one light sample in each region. Decomposing processing, including regular [Debevec 2006][Wan et al. 2005][Annen et al. 2008] and irregular decomposition [Agarwal et al. 2003], produces a set of small regions with somehow balanced importance. Consequently, more samples will be allocated in areas with high importance (such as highlights). There are some methods proposed to generate samples by warping algorithms based on wavelet [Clarberg et al. 2005][Clarberg and Akenine-Möller 2008] or spherical harmonics [Jarosz et al. 2009].

High dynamic environment sequence makes sampling more complicated. To keep temporal coherence in rendered results, consecutive frames need to be considered in order to suppress large differences in sample patterns across frames. [Ghosh et al. 2006] presents a sequential Monte Carlo (SMC) algorithm to draw samples from the product of illumination and BRDF, and propagates them in time through sequential importance sampling. [Hašan et al. 2008] treats the sequence rendering as a tensor formulation, where each tensor element represents the contribution of one light to one pixel in one frame. It samples rows, columns of the tensor and clusters efficiently a few lights to approximate the animation. [Wan et al. 2011] treats an environment sequence as a spatio-temporal volume, and decomposes the volume by adaptively constructing a Kd-tree.

In this work, we are focused on processing on-line captured environment sequences. Unlike those offline methods [Hašan et al. 2008][Wan et al. 2011], we only utilize current frame and one past frame to propagate and generate temporally coherent light samples.

2.2 Video Segmentation

Image segmentation aims to collect perceptually similar or homogeneous pixels into regions based on texture or color. Video segmentation is more complicated by considering additional motion information. Some image segmentation methods [Winnemöller et al. 2006] [Chen et al. 2007] are directly applied to video frames. However, due to lack of temporal information, the generated segmentation results may present jitters across time. Most video analysis methods utilize temporal correspondences across frames, besides proper feature of each frame.

Spatio-temporal video segmentation methods can be divided into two groups. One is without knowledge about future frames during each frame processing, while the other is considering long-range relation into past frames. [Brendel and Todorovic 2009] presents an unsupervised video object segmentation with respect to region contours in all frames. In [Grundmann et al. 2010], a hierarchical graph-based algorithm is used for long video sequences by iteratively refining a tree of spatio-temporal segmentations after whole volume oversegmentation. [Paris 2008] only relies on past frames to maintain temporal coherence and derived a mixed kernel of temporal exponential decay function and Gaussian function in mean-shift segmentation [Debevec and Malik 1997][Georgescu et al. 2003]. [Chang et al. 2013] conducts temporal superpixels (TSPs) using only current frame and one past frame. The same TSPs across the frames belong to the same objects.

3 Our Algorithm

The basic idea of our method is dividing each frame of environment sequence into strata of close importance, with an emphasis on coherence across frames. Our method relies on temporal superpixels (TSPs) [Chang et al. 2013], which can effectively detect objects' boundaries and track the same part of an object across frames. Specifically, we first deal with the first two consecutive frames by generating their TSPs, which imply the correspondence relationship between them. Then we construct small strata with balanced importance via an adaptive merge-and-split scheme on the first frame. Based on strata of the first frame, we map merged parts of strata to the second frame and apply the adaptive merge-and-split scheme on the second frame. Each light sample corresponds to a superpixel on one frame. By this way, we are able to generate temporally coherent sample patterns in the entire sequence. The pseudo codes of this process are shown in Algorithm 1. In the following, we start briefly introducing the process of temporal superpixels construction.

Algorithm 1 Online Environment Sequence Sampling Based On TSPs

```
1:  Input: $frame\ set\ F$
2:      $compute\ superpixels\ of\ first\ frame\ S_{F_1}$
3:      $adaptive\ merge-and-split\ S_{F_1}$
4:      $sampleframe\ F_1\ to\ get\ Sa_1$
5:    for $f = 2...n$
6:        $compute\ temporal\ superpixels\ of\ S_{F_f}$
7:        $map\ merged\ parts\ from\ frame\ F_{f-1}$
8:        $adaptive\ merge-and-split\ S_{F_f}$
9:        $sample\ frame\ F_f\ to\ get\ Sa_f$
10: return $Sa$
```

3.1 Temporal Superpixels

The method proposed in [Chang et al. 2013] extends the SLIC algorithm to get temporal superpixels in an online fashion. SLIC [Achanta et al. 2010] performs a local clustering of pixels in a 5-dimensional space composed of x-, y-coordinates and Lab color values. The SLIC algorithm can be summarized in two steps: (1) initialize k cluster centers by random sampling, (2) iteratively allocate the best matching pixels around cluster centers according to distance measure and update cluster centers to produce superpixels. [Chang et al. 2013] utilizes digital topology concepts to formulate a generative model to mimic SLIC. To handle temporal dimension, it models flow between frames with a bilateral Gaussian process and uses this information to propagate superpixels between two consecutive frames. Since we simply apply this method, readers are referred to the paper [Chang et al. 2013] for detail.

Temporal superpixels results of a HDR environment sequence are demonstrated in Figure 1. The corresponding regions on two frames are labelled with same colors. As we can see, TSPs can trace objects across time and achieve good correspondences.

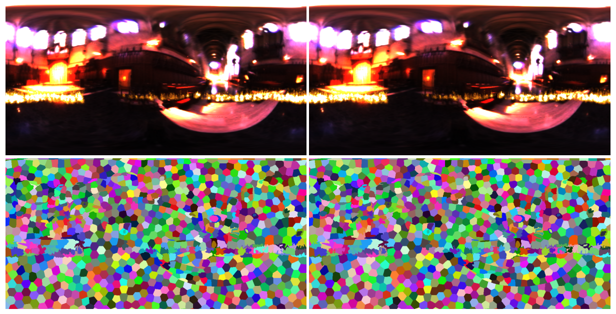

Figure 1: *Temporal superpixels on 107 and 108 frames of grace_flame sequence.*

3.2 Segments Construction via Adaptive Merge-and-Split Scheme

Given a single frame in an environment sequence, the shapes of light regions can be well captured by temporal superpixels, in each of which pixels have nearly uniform intensities. However, the importance of superpixels may be non-uniform over the frame. For instance, we observe that highlight regions may occupy relatively large areas. On the contrary, visible light regions may possess small areas, each of which owns lower imporatnce.

To construct a set of strata with balanced importance, we propose an adaptive merge-and-split scheme on existing temporal superpixels. Here, we adopt an importance metric of light regions presented in [Agarwal et al. 2003]. The metric mixes light intensity and region area as follow.

$$\Gamma\left(L, \Delta\omega\right) = L^a \Delta\omega^b, \tag{1}$$

where $\Delta\omega$ refers to the solid angle of light regions, L is the integrated illumination, and $a = 1$, $b = 1/4$ due to a visibility-based analysis in [Agarwal et al. 2003].

Adaptive merge-and-split scheme is implemented on a single frame as an iterative process (see Algorithm 2). Each iteration consists of a merging operation and a splitting operation. For merging operation, the average importance among segments is first computed. We then detect to-be-merged regions which have relatively lower importance values. After the merging operation, we recompute the average importance value for the splitting process. Again, to-be-split regions which have rather high importance are determined and split into small strata. This process iterates until the average importance value is nearly stable. To avoid unnecessary repetition, we demand no intersection exist between the set of merged regions and the set of split regions. In other words, merged regions can be never split during the processing.

Algorithm 2 Adaptive Merge-and-Split Scheme

```
1:  Input: $Region\ set\ R,\ adjacency\ matrix\ \mathcal{A}.$
2:  do
3:     $compute\ \Gamma_a\ of\ all\ regions$
4:     $determine\ the\ merged\ region\ set\ \{i^*\}$
5:     for $each\ i^*$
6:        $find\ neighbor\ N_{i^*}\ from\ \mathcal{A}$
7:        for $j \in N_{i^*}$
8:           $find\ region\ j^*\ due\ to\ Eqn\ (4)$
9:           $merge\ region\ i^*\ and\ region\ j^*$
10:          $update\ R\ and\ \mathcal{A}$
11:    $recompute\ \Gamma_a\ of\ all\ regions$
12:    $determine\ the\ split\ region\ set\ \{k^*\}$
13:    for $each\ k^*$
14:       $split\ region\ k^*$
15:       $update\ R\ and\ \mathcal{A}$
16:    $compute\ the\ update\ average\ importance\ \Gamma_{an}$
17: while $|\Gamma_a - \Gamma_{an}| > \varepsilon$
18: return $R$
```

3.2.1 Merging Operation

Merging operation aims to group small regions with lower importance values. Toward this end, we compute an average importance metric Γ_a of all regions on one frame, given by

$$\Gamma_a = \frac{\sum_{i=1}^{N} \Gamma(L_i, \Delta\omega_i)}{N}, \tag{2}$$

where N is the amount of segments. Then, we find out all the regions, whose importance values are lower than the minimum value of Γ_a^{t-1} and Γ_a^t, i.e.

$$\{i^* \mid \Gamma^t(L_{i^*}, \Delta\omega_{i^*}) < \alpha \min(\Gamma_a^{t-1}, \Gamma_a^t)\}, \tag{3}$$

where i^* is region index, t is frame index and α is empirically set to $1/2$. To keep temporal coherence, we expect to have Γ_a^t of the current frame close to Γ_a^{t-1} of the last frame. Hence we choose the minimum value of them in the merging condition to avoid producing large rendering shaking. For the first frame, Γ_a^{t-1} is set as the same as Γ_a^1.

Given a to-be-merged region, one of its neighbors needs to be chosen for the merging, so that the importance value of the combination is the minimum among all possible choices. This is formulated as,

$$j^* = \arg\min_{j \in \eta(i^*)} \Gamma^t(L_{i^*} + L_j, \Delta\omega_{i^*} + \Delta\omega_j), \tag{4}$$

where,

$$\left| I_{avg}^t(i^*) - I_{avg}^t(j^*) \right| < \varepsilon. \tag{5}$$

Here $\eta(i^*)$ denotes the neighbors of region i^*, and $I_{avg}^t(i^*)$ is the region's average intensity value. Equation 5 ensures that the average intensity difference between i^* and its chosen neighbor j^* is lower than a small threshold ε. The merging process can be effectively accomplished via an adjacency matrix $\mathcal{A}$, which records the neighborhood information and whether regions have been merged. The upper row in Figure 2 illustrates the merging result of the first frame.

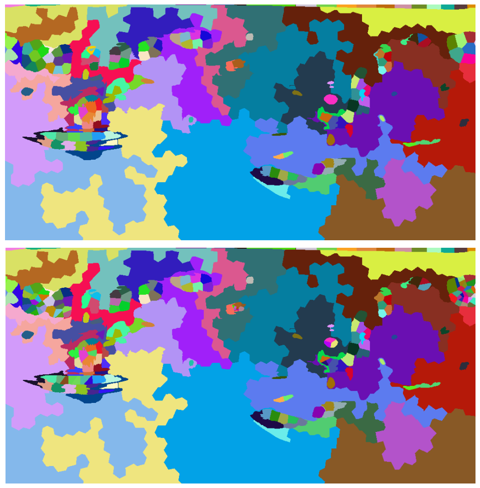
Figure 2: *From the top down shows merging and splitting results on the first frame of "grace_flame" sequence.*

3.2.2 Splitting Operation

We tend to split highlight regions with high importance into smaller areas. To be specific, we firstly seek out those regions with large importance values, i.e.

$$\{k^* \mid \Gamma^t(L_{k^*}, \Delta\omega_{k^*}) > \beta\Gamma_a\}, \tag{6}$$

where k^* is the region index, and parameter β is empirically set as $\beta = 2$. Then we figure out the number of newly generated small regions as,

$$N_{k^*}^t = \min(\left\lfloor \frac{\Gamma^t(L_{k^*}, \Delta\omega_{k^*})}{\Gamma_a^t} \right\rfloor, M_{k^*}), \tag{7}$$

where M_{k^*} is the number of pixels in region k^*. Since the region is intensity uniform, we split the region by using Hochbaum-Shmoys algorithm [Hochbaum and Shmoys 1985].

The lower row in Figure 2 shows the final result by adaptive merge-and-split on the first frame. We can see that many visible light regions are merged to large area regions, and that highlight regions are decomposed into smaller areas. Up to now, we have decomposed a single frame into a set of disjointed regions with balanced importance.

3.3 Segment Mapping Between Consecutive Frames

Given the environment sequence, we want to propagate the sample information of the previous frame to the current frame to keep temporal coherence. This can be achieved by utilizing the correspondences embedded in generated temporal superpixels. It is noted that the superpixel labels in merged regions can be traced across time, but newly generated segments in split regions have no temporal correspondences between frames. Therefore, we will only map merged regions in the previous frame to the current frame. To be specific, for each of temporal superpixels in the current frame, we find its corresponding superpixels in the previous frame. If a set of temporal superpixels have been merged in the previous frame, we will apply the merging operation to the superpixels with the same labels in the current frame. By this way, we can directly map merged parts of the previous frame with balanced importance to the current frame. This mapping process is equivalent to propagating samples across consecutive frames.

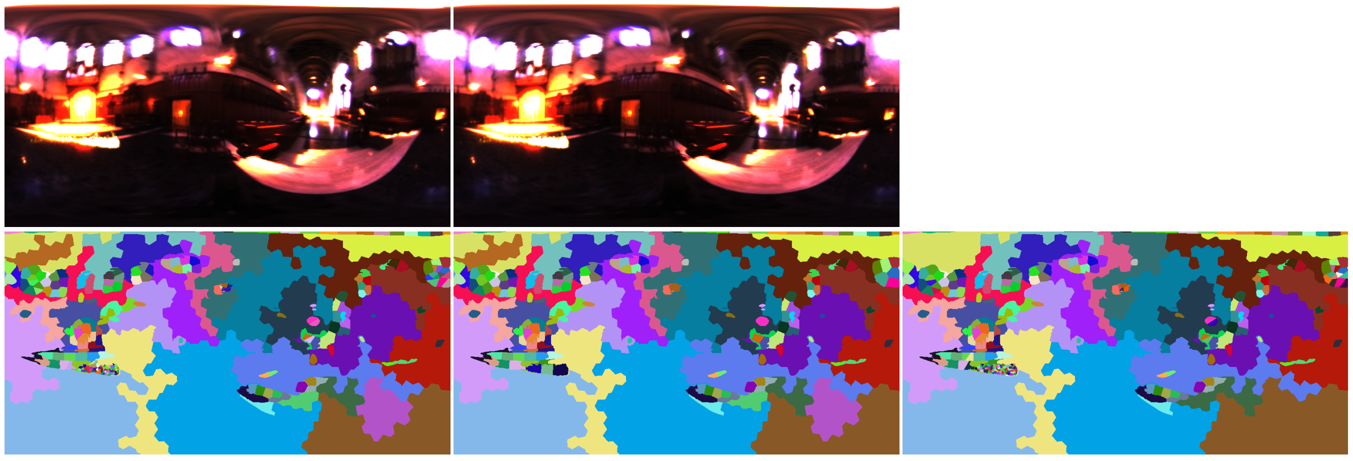

Figure 3: *Mapping results from 30-th frame to 31-th frame of "grace_flame" sequence: (a) the 30-th environment frame and its sampling strata; (b) the 31-th environment frame and the mapped results from the 30-th environment frame; (c) the sampling strata of the 31-th environment frame.*

During the mapping process, we observe that some highlights, expected to be split, may appear in some places corresponding to merged regions on the previous frame. The above mapping scheme may wrongly merge highlights with surrounding superpixels. We then set a constraint on intensity changes for avoiding missing highlight regions in the merging, which requires that a merged region candidate R on frame t can be merged, if its importance is equal or less than the average importance metric of frame $t-1$, that is

$$\Gamma_R^t \leq \Gamma_a^{t-1}. \tag{8}$$

Figure 3 shows a mapping process. After adaptive merge-and-split process, we get 30-th frame's state as Figure 3(a). Then we map merged regions of 30-th frame to 31-th frame based on TSPs (Figure 3(b)). Lastly, we still employ adaptive merge-and-split scheme to obtain 31-th frame's strata, shown in Figure 3(c).

4 Experimental Results

We compare our approach with four state-of-the-art methods, including two static sampling methods, namely structured importance sampling (structured) [Agarwal et al. 2003] and hierarchical sample warping (HSW) [Clarberg et al. 2005]; and two sequence sampling methods, namely spherical q^2-tree [Wan et al. 2005] and spatio-temporal volume sampling (STS) [Wan et al. 2011]. Also note that spherical q^2-tree utilizes temporal coherence between frames mainly for acceleration. It achieves temporal coherent sampling thanks to its regular decomposition over spatial domain. STS, on the other hand, exploits both the temporal and spatial coherence of environment sequences, by adaptively stratifying the whole sequence as a volume.

In the experiments, we use two HDR environment map sequences as light sources [Wan et al. 2011]. The "reading_room" environment sequence serves for illuminating a set of table and chair models, while the "grace_flame" environment sequence for a scene of girl, and both contain 120 frames. The ground truth results are rendered by uniformly sampling each environment map with 100,000 light samples. Since our method can automatically determine the strata number, we use this number to create samples for other methods. Here, the average number of samples per frame is 260 for "grace_flame" sequence and 97 for "readingroom" sequence. We evaluate the rendering quality with three image quality metrics, namely SSIM, HDRVDP2 [Mantiuk et al. 2011] and RMSE.

4.1 Visual Comparison

We first show rendered results of different sampling methods for 5 frames (20, 40, 60, 80, 100) of "grace_flame" sequence in Figure 4. We can see that for the first four frames, our results suffer from less errors in shadows on the background curtain. When the fire is flaming, other methods produce lots of hard shadows for the girl model. For instance, structured method and HSW method generate hard shadows on the curtain behind the girl; spherical q^2-tree and STS generate hard shadows on the curtain beneath the girl. We think that hard shadows are resulted mainly due to insufficient light samples in some important light regions. We further visualize the SSIM error maps for these frames in Figure 5. In SSIM maps, pure red color denotes a low SSIM value representing large difference, while blue color is the reversal. Obviously, our results have higher SSIM values for the results of these five frames.

For "readingroom" sequence, our results (shown in Figure 6) present softer shadows than those from structured method, spherical q^2-tree and STS, especially near the shadows of the left chair. Although our results look similar to those from HSW in this example, our method exhibits more consistent visual experience in the rendered sequence. Readers are referred to the supplementary video for a close comparison. In summary, our method is able to achieve visually pleasing rendering results for the two testing environment sequences, each illuminating different 3D scenes.

4.2 Image Quality Metric Evaluation

In the second experiment, we quantitatively evaluate the rendering quality by using three image quality metrics, SSIM, RMSE and HDRVDP2. As demonstrated in [Čadík et al. 2012], SSIM is a robust perceptual metric for image quality. RMSE is classically used to measure image differences. HDRVDP2 [Mantiuk et al. 2011] is a calibrated visual metric for visibility and image quality. For SSIM metric, a value close to 1 represents better similarity to the groundtruth, while a high quality rendered result is also expected to have a close-to-zero RMSE value and a close-to-100 HDRVDP2 MOS value.

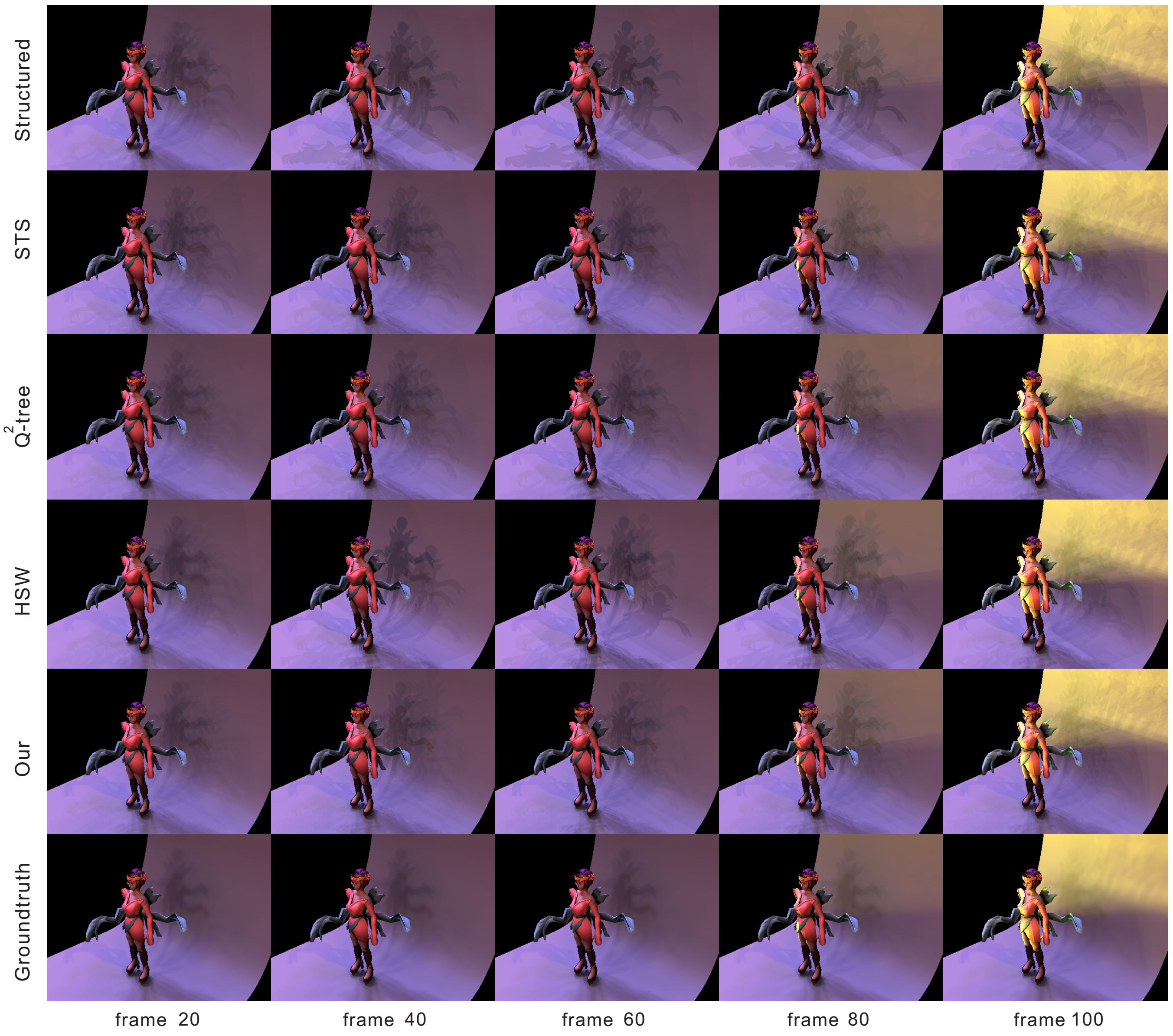

Figure 4: *Visual comparison of five rendered frames by using "grace_flame" sequence.*

In overall speaking, we generally have comparable or better rendering results as compared to state-of-the-art sampling methods. As shown in Figure 7(a), our method gets higher SSIM values for "grace_flame" sequence than other methods, and only lower than HSW near the end. In terms of RMSE metric (Figure 7(b)), our method reports lower or comparable errors to STS method, better than spherical q^2-tree as well as the other two static sampling methods. When evaluating results using HDRVDP2 metric, we achieve high MOS values in most frames, only a bit low during 80-100 frames in Figure 7(c). With the three figures together, we can say that our method yields relatively consistent and better quantitative metric values than those methods in comparison.

For "reading_room" sequence, we get similar performance. In more detail, our method produces SSIM values only lower than those from HSW, and HDRVDP2 values comparable to HSW results, which are consistently better than other results. We only get relatively poor RMSE values for "reading_room" sequence.

4.3 Rendering Coherence Evaluation

In this section, we employ the temporal inconsistency metric proposed in [Wan et al. 2011] to evaluate rendering coherence by different methods. The metric computes temporal differences of rendered results from a sampling method and groundtruth results, given by

$$E(t) = \frac{1}{N}\sum_{i}\omega_i \left|\Delta X_i(t) - \Delta \hat{X}_i(t)\right|, \quad (9)$$

where $X_i(t)$ is the intensity of pixel i on t-th rendered frame of a sampling method and $\hat{X}_i(t)$ is counterpart of groundtruth. $\Delta X_i(t)$ is defined as the pixel difference between two rendered frames, i.e. $\Delta X_i(t) = X_i(t) - X_i(t-1)$. The weighting factor is computed as $\omega_i = \left|X_i(t) - \hat{X}_i(t)\right| + 1$. A close-to-zero metric value indicates that a sampling method possesses high temporal coherence.

Figures 8 and 9 show the inconsistency metric values of differen-

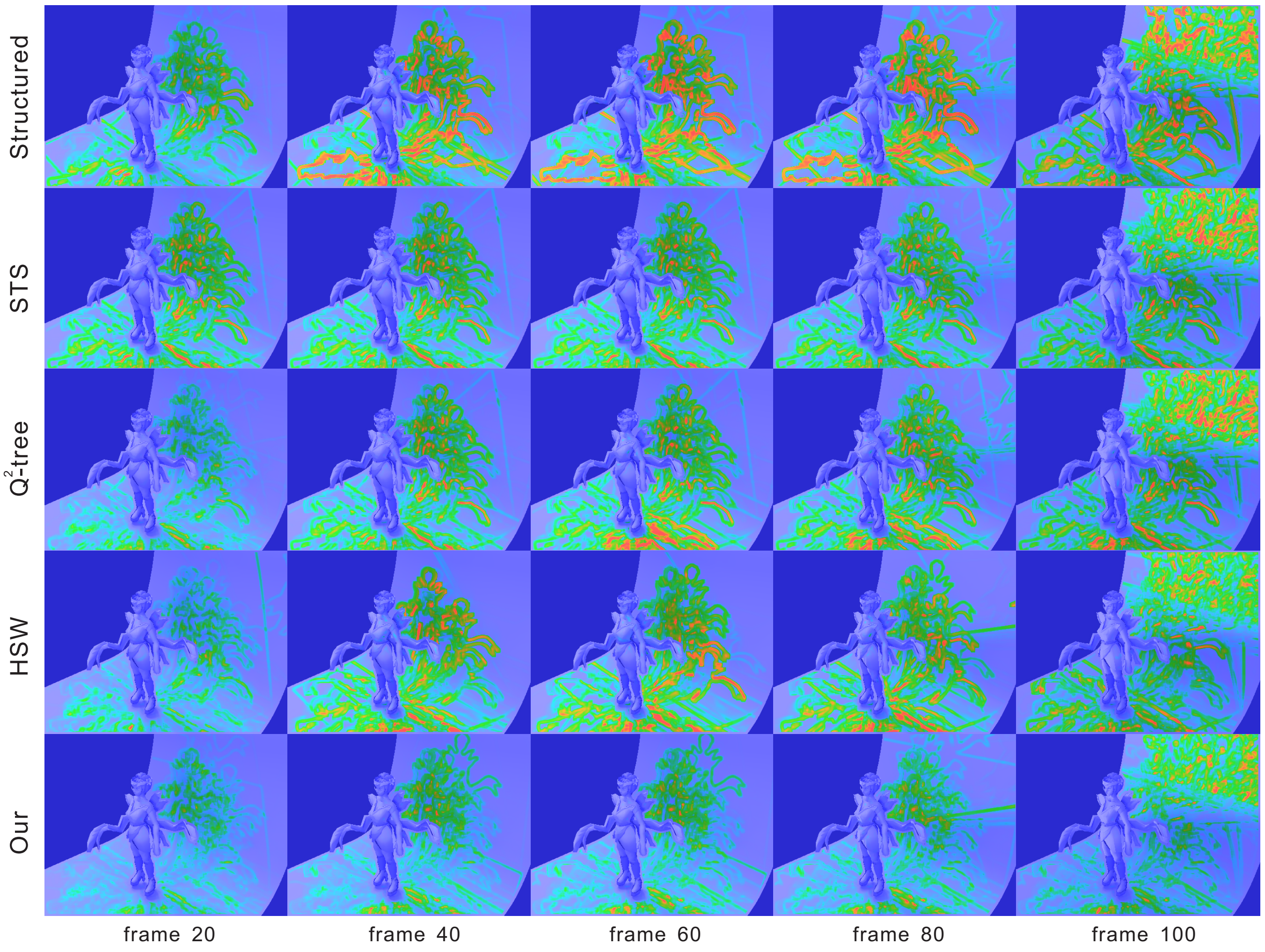

Figure 5: *SSIM maps of different methods for the five rendered frames by using "grace_flame" sequence.*

t methods in two scenes. Compared with the two static sampling methods, i.e. structured and HSW, our inconsistency values are closer to zero and keep better temporal coherence as a whole. On the other hand, we achieve comparable results to spherical q^2-tree, while a bit lower than STS method. However, STS considers all the frames simultaneously, but we only use two frames at each time instance. In addition, our method acquires consistently better rendering quality than spherical q^2-tree as shown in the previous section.

4.4 Timing Performance Evaluation

We now report the timing performance of different sampling methods. Our experiments are conducted on an Intel Core i5 3.0GHz desktop with four cores and 8G memories. Among the comparative methods, STS and spherical q^2-tree are implemented in C/C++ code, and structured method and HSW are provided in Matlab code. For our approach, the implementation can be split into three parts, optical flow computation which is the prerequisite for TSPs generation, TSPs generation, and our sequential sampling. We use the authors' codes to do the first two parts [Chang et al. 2013]. Table 1 shows the time for processing a whole environment map sequence. We find that our unoptimized implementation spends longer time than STS and spherical q^2-tree, and that the most costly operation is the computation of optical flow and TSPs.

Table 1: *Processing Time of Different Methods.*

Methods		Processing Time (s)	
		grace_flame	readingroom
Structured (M)		1,279	1,837
STS (C)		43	41
q^2-tree (C)		27	21
HSW (M)		1,328	1,261
Our	Optical flow (M)	4,223	3,391
	TSP (M)	2,028	1,750
	Sampling (C)	132	110

4.5 Discussion

Since the proposed sampling method relies on the correspondence information from TSPs, the erroneous correspondences between consecutive frames if there is may affect the merge-and-split process. However, this situation is quite seldom in our experiments. Besides, we currently adopt the authors' Matlab code for optical flow and TSPs generation [Chang et al. 2013], which is rather time consuming as demonstrated in Table 1. We would like to accelerate our sampling code as well as the optical flow and TSPs generation

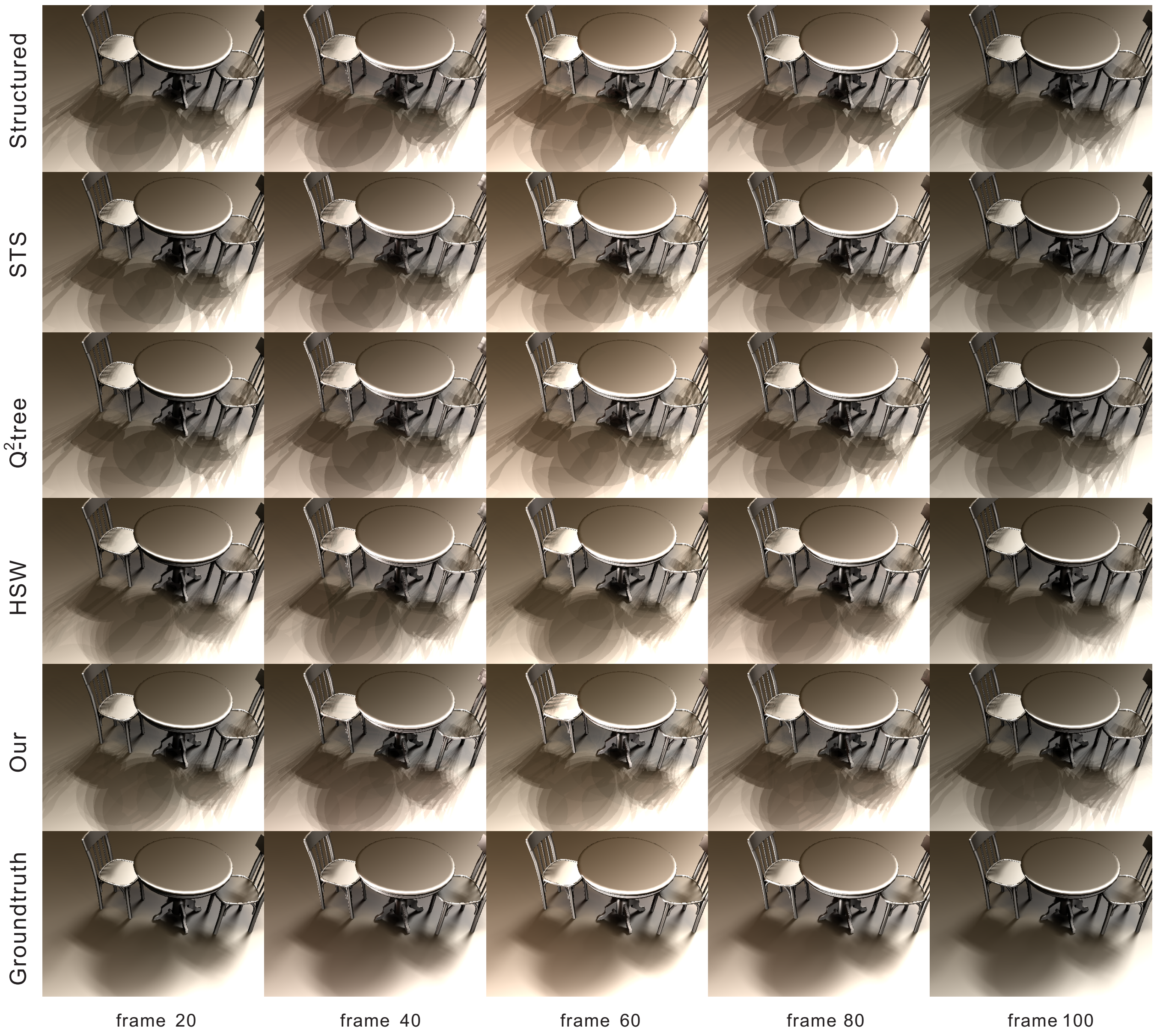

Figure 6: *Visual comparison of five rendered frames by using "reading_room" sequence.*

code in order to facilitate real-time performance.

5 Conclusion

In this paper, we have presented a new online environment sequence sampling method. Our method relies on the powerful temporal superpixels to get correspondences between two consecutive frames. Only considering one previous frame, we can achieve high temporal coherence in on-line fashion. An adaptive merge-and-split scheme is developed to generate more balanced importances regions based on temporal superpixels. The effectiveness of the proposed method is validated through both visual comparison and quantitative evaluation. We found that our method achieves better rendering quality in terms of three typical image quality metrics, and acquires comparable rendering coherence as compared to environment sequence sampling methods.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (61100122 and 61100121), and Tianjin Science Foundation for Youth (12JCQNJC00100).

References

ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SÜSSTRUNK, S. 2010. Slic superpixels. Tech. rep.

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. 2003. Structured importance sampling of environment maps. *ACM Transactions on Graphics (TOG) 22*, 3, 605–612.

ANNEN, T., DONG, Z., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2008. Real-time, all-frequency shadows

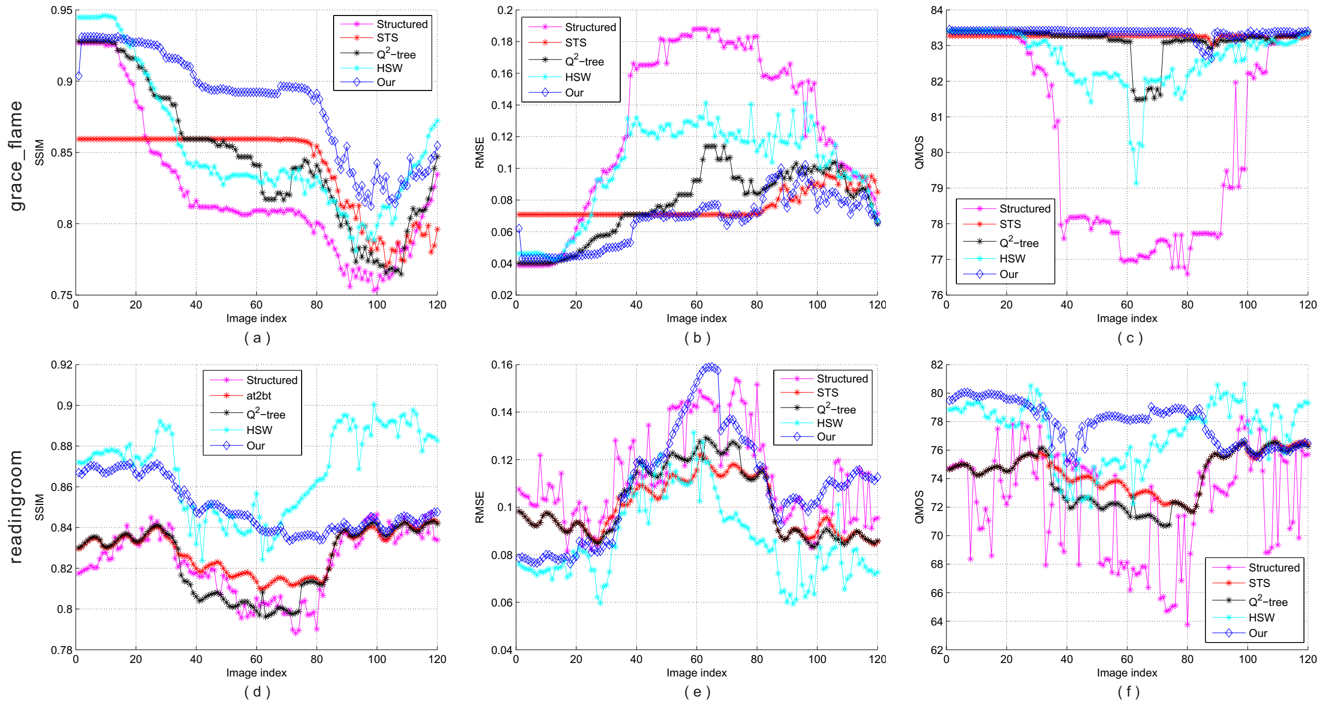

Figure 7: *Image quality metric evaluation for different methods on two environment sequences.*

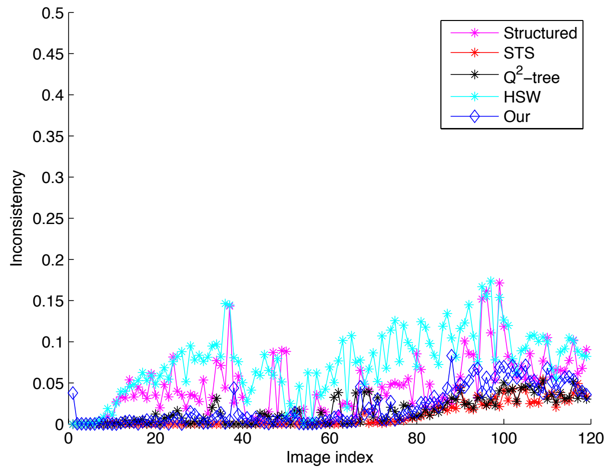

Figure 8: *Rendering inconsistency evaluation for "grace_flame" sequence.*

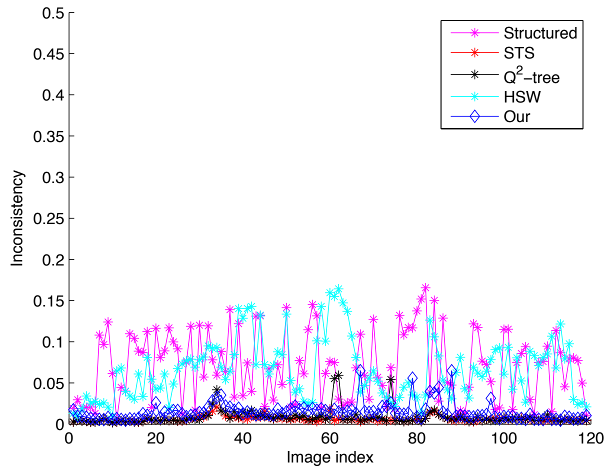

Figure 9: *Rendering inconsistency evaluation for "reading_room" sequence.*

in dynamic scenes. *ACM Transactions on Graphics (TOG) 27*, 3, 34.

BRENDEL, W., AND TODOROVIC, S. 2009. Video object segmentation by tracking regions. In *IEEE 12th International Conference on Computer Vision*, IEEE, 833–840.

ČADÍK, M., HERZOG, R., MANTIUK, R., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2012. New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Transactions on Graphics 31*, 1–10.

CHANG, J., WEI, D., AND FISHER III, J. W. 2013. A video representation using temporal superpixels. In *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2051–2058.

CHEN, J., PARIS, S., AND DURAND, F. 2007. Real-time edge-aware image processing with the bilateral grid. *ACM SIGGRAPH 2007 Papers*.

CLARBERG, P., AND AKENINE-MÖLLER, T. 2008. Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proceedings of Eurographics 2008) 27*, 2, 681–690.

CLARBERG, P., JAROSZ, W., AKENINE-MÖLLER, T., AND JENSEN, H. W. 2005. Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Transactions on Graphics (TOG) 24*, 3, 1166–1175.

DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Siggraph*, 369–378.

DEBEVEC, P. 2006. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2006 Courses*, ACM, 6.

GEORGESCU, B., SHIMSHONI, I., AND MEER, P. 2003. Mean shift based clustering in high dimensions: a texture classification example. In *ICCV*, 456–463.

GHOSH, A., DOUCET, A., AND HEIDRICH, W. 2006. Sequential sampling for dynamic environment map illumination. In *Proceedings of the 17th Eurographics conference on Rendering Techniques*, Eurographics Association, 115–126.

GIBSON, S., AND MURTA, A. 2000. *Interactive rendering with real-world illumination*. Citeseer.

GRUNDMANN, M., KWATRA, V., HAN, M., AND ESSA, I. 2010. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2141–2148.

HAŠAN, M., VELÁZQUEZ-ARMENDÁRIZ, E., PELLACINI, F., AND BALA, K. 2008. Tensor clustering for rendering many-light animations. *Computer Graphics Forum 27*, 4, 1105–1114.

HOCHBAUM, D. S., AND SHMOYS, D. B. 1985. A best possible heuristic for the k-center problem. *Mathematics of operations research 10*, 2, 180–184.

JAROSZ, W., CARR, N. A., AND JENSEN, H. W. 2009. Importance sampling spherical harmonics. *Computer Graphics Forum 28*, 2, 577–586.

KOLLIG, T., AND KELLER, A. 2003. Efficient illumination by high dynamic range images. In *Proceedings of the 14th Eurographics workshop on Rendering*, Eurographics Association, 45–50.

MANTIUK, R., KIM, K. J., REMPEL, A. G., AND HEIDRICH, W. 2011. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics 30*, 4 (July), 40:1–40:14.

PARIS, S. 2008. Edge-preserving smoothing and mean-shift segmentation of video streams. In *ECCV*. Springer, 460–473.

RUBINSTEIN, R. Y. 2009. *Simulation and the Monte Carlo method*, vol. 190. Wiley-interscience.

WAN, L., WONG, T.-T., AND LEUNG, C.-S. 2005. Spherical q 2-tree for sampling dynamic environment sequences. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, Eurographics Association, 21–30.

WAN, L., MAK, S.-K., WONG, T.-T., AND LEUNG, C.-S. 2011. Spatiotemporal sampling of dynamic environment sequences. *IEEE Transactions on Visualization and Computer Graphics 17*, 10, 1499–1509.

WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM SIGGRAPH 2006 Papers*, 1221–1226.