EG UK Theory and Practice of Computer Graphics (2012)
Hamish Carr and Silvester Czanner (Editors)

Light clustering for Dynamic Image Based Lighting

S. Staton, K. Debattista, T. Bashford-Rogers & A. Chalmers

Abstract

High Dynamic Range (HDR) imagery has made it possible to relight virtual objects accurately with the captured lighting. This technique, called Image Based Lighting (IBL), is a commonly used to render scenes using real-world illumination. IBL has mostly been limited to static scenes due to limitations of HDR capture. However, recently there has been progress on developing devices which can capture HDR video sequences. These can be also be used to light virtual environments dynamically. If existing IBL algorithms are applied to this dynamic problem, temporal artifacts viewed as flickering can often arise due to samples being selected from different parts of the environment in consecutive frames. In this paper we present a method for efficiently rendering virtual scenarios with such captured sequences based on spatial and temporal clustering. Our proposed Dynamic IBL (DIBL) method improves temporal quality by suppressing flickering, and we demonstrate the application to fast previews of scenes lit by video environment maps.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Real world lighting is often an essential component of many rendering systems. This lighting is commonly captured from all directions on a (hemi)sphere at a single point using HDR capture which is capable of acquiring the entire lighting of an environment. The resulting Environment Map (EM) is then used in rendering systems to accurately relight virtual scenes. This finds use in a large range of applications; from video games to architectural design [DBB02]. Many of the algorithms used in these scenarios have been specifically designed to deal with static lighting environments encoding distant real-world lighting at a single time point. However, recently, the ability to capture HDR video sequences [CBB*09] has made it possible to compute animations lit by dynamic environments. In this more general case of lighting from Video Environment Maps (VEM), the algorithms which are designed for static lighting are less optimal, and often lead to a significant drop in temporal quality over an animation sequence. This is especially noticeable in interactive rendering algorithms which approximate the lighting environment by a small set of Virtual Directional Light (VDL) sources. If conventional sampling strategies are used there may be large jumps in the directions of the VDLs across consecutive frames which lead to noticeable flickering in an image sequence rendered with DIBL.

In this paper, we present a novel approach to maintain temporal quality. Our approach relies on an observation about DIBL sequences: there are large regions of similar luminance throughout most sequences, interspersed with infrequent jumps in luminance. We therefore exploit this natural coherence in these sequences through a two stage process; firstly generating a set of VDLs for all the frames of a sequence, then, secondly, clustering them to minimise flickering. As we pre-determine the total number of clusters, our method also has the additional advantage of automatically placing more clusters in high frequency regions in the 4D space of directions, luminance and time.

The main contribution of this paper is a clustering algorithm over unbiased samples taken from the VEM which reduces temporal artefacts. This method works by generating a 3D volume of samples over all the frames in a VEM, and performing clustering over their 4D properties. Using this method, we obtain a significant improvement in temporal quality for DIBL.

This paper is structured as follows; Section 2 details some of the relevant related work, Section 3 details our clustering method for reducing flickering, Section 4 shows results for our algorithm. Finally, conclusions and future works are presented in Section 5.

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG12/017-024

2. Related Work

Accurately calculating the lighting in rendering systems is a costly task. This is only made more complicated with the addition of complex real world lighting from HDR EMs. This section looks at existing methods of handling both IBL and DIBL. In particular, we focus on just the methods that sample directly from the environment rather than a combination of the EMs and surface reflectance.

2.1. Image Based Lighting

This concept was introduced by Blinn and Newell [BN76], then extended by Debevec [Deb98] to use HDR images to represent real world lighting. Extensive previous work in the field has focused on improving the sampling techniques of EMs. In parricular, importance sampling [PH10a] has been used to effectively generate samples for diffuse scenes. Agarwal et al. [ARBJ03] developed a method called Structured Importance sampling that stratifies the EM into regular strata based on luminance and predicted occlusion from the scene which allows for speedup in occlusion calculations. Debevec [Deb05] applied a median cut algorithm to EMs that separated it into rectangular regions based on luminance and placed light sources at the brightest points.

Bidirectional importance sampling [BGH05] is a technique that samples from the product of the EM and the Bidirectional Reflectance Distribution Function (BRDF), based on rejection sampling. However rejection sampling can be very computationally expensive with no guaranteed run time. Resampled importance sampling [TCE05] generates samples by firstly generating a large set of samples and then resampling from that set. Unlike the other methods the samples drawn are approximately distributed according to the desired probability density function. This method can be costly as a very large initial set of samples must be generated to achieve good results.

2.2. Dynamic Image Based Lighting

Liang et al. [WWL05] built a spherical q^2-tree from the EM and adapted it frame by frame based on changes in the luminance. This method only repositions light sources in areas of the EM where the luminance has changed significantly, therefore reducing flicking. This work was then extended [WMWL11] into a spatiotemporal sampling method that pre processes the VEM into spatiotemporal volumes. These are used to create VPLs that are shared across frames, further reducing flickering and improving rendering quality. However this method can lead to a waste of samples in low-energy regions because of how it stratifies the VEM into rectangular-subvolumes.

Havran et al. [HSK*05] extended their previous work [HDS03] on mapping the unit square to a uniform hemisphere into a DIBL method. They used this to create a set of representative directional light sources which are filtered over time to reduce flickering. The luminance of the lights are redistributed to create a smoother animation, however this redistribution introduces bias and the resulting images will not conform to ground truth.

Ghosh et al. [GDH06] introduced an algorithm that uses bidirectional importance sampling for the first frame of the VEM and then propagates the samples for further frames. It uses a Markov Chain Monte Carlo (MCMC) transition kernel to redistribute the samples.

3. Light clustering

Image based lighting generates images lit from an EM via solving the rendering equation [Kaj86] at a specific point x in the scene:

$$L_o(x,\omega,t) \;=\; \int_{\Omega} f_r(x,\omega,\omega')L(\omega',t)\cos\Theta d\omega' \qquad (1)$$

where L_o is the outpoint radiance in the direction ω, t is the time at which the image is rendered, i.e. the frame in the VEM, Ω is the positive hemisphere above x, f_r is the bidirectional reflectance distribution function BRDF, and $L(\omega',t)$ is the incoming radiance from direction ω'. This integral is commonly solved through approximating the EM as N VDLs. Therefore, Equation 1 can be approximated as:

$$L_o(x,\omega,t) \;\approx\; \frac{1}{N}\sum_{i=0}^{N}\frac{f_r(x,\omega,\omega_i')L(\omega_i',t)\cos\Theta}{p(\omega_i',t)} \qquad (2)$$

where $p(\omega_i',t)$ is the probability density function for generating the i'th VDL. This procedure of generating samples is known as importance sampling. If $p(\omega_i',t)$ is proportional to $L(\omega_i',t)$, variance in the image from sampling the lighting environment will be minimised. However, when the application is for fast, or interactive image previews, N will be small. When this is the case, samples may randomly jump around the EM (even if quasi-random sampling [Nie92] is used). Therefore, to keep the benefits of sampling proportional to the lighting, but to minimise flickering, we propose a clustering based method.

The method has two major parts, all of which are carried out as preprocessing steps:

- The first is the initial sampling of the VEM.
- The second is the clustering of these samples and creating light sources from the clusters.

The following section describes this process in more detail.

3.1. Initial Sampling of the VEM

The method starts by sampling each of the individual frames of the VEM, creating a large number of samples to then

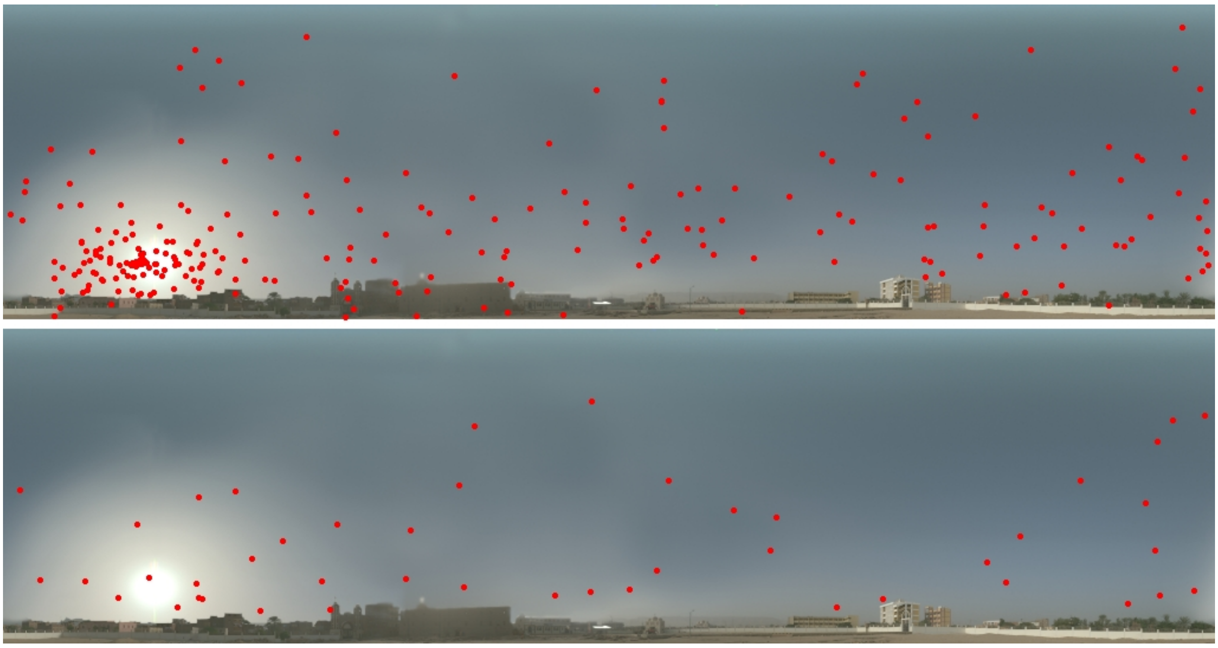

Figure 1: *Top: Importance sampling of frame with 256 samples. Bottom: Light positions after clustering of samples using 56 clusters.*

be used by the clustering algorithm. Importance sampling [PH10a] was selected as it generates samples only from the EM, rather than a combination of the EM and the BRDF, meaning that the samples are independent of the scene geometry and therefore the cluster will not need to be recreated for different scenes. It will also generate more samples in high frequency areas of the EM naturally, allowing them to be clustered more effectively and to more accurately represent, even with N discrete samples, the lighting information present in the EM.

3.2. Clustering

Once the samples have been generated they can be clustered into representative light sources that are shared across many frames of the animation. To take advantage of the temporal coherence, the clustering will not just be performed on the positions of the samples, but also based on their luminance and by time. Samples are being clustered by time to take advantage of temporal coherence and to encourage the creation of clusters with samples in consecutive frames. The luminance of samples will also be taken into account as it will reduce the chance of high frequency samples being lost due to being clustered with low frequency samples.

K-means [For65] clustering was the clustering algorithm used. It was chosen over other clustering algorithms as it is fast ($O(n \log n)$ where n is the number of samples), and is centroid based. This means that it will cluster samples based on their distance from the central point of the cluster, rather than based on a distribution or density of samples. Another benefit of this method is that it will assign every sample to a cluster, where other methods would classify some samples as noise and then ignore them.

The outline of the K-means algorithm is given below:

1. Select k random samples to be the initial centres of the k clusters.
2. Assign each sample to its closest cluster.
3. Recalculate the centre point of each cluster.
4. Repeat step 2 and 3 until no samples are reassigned to different clusters.

To use this algorithm a metric is needed to decide the distance between each of the samples that takes into account position, luminance and time. A metric has been designed for each of these and they can be combined to give the distance, $D_{u,v}$, between two samples, u and v, so that they can be clustered. This will be defined as:

$$D_{u,v} = \alpha P_{u,v} + \beta L_{u,v} + \gamma T_{u,v} \tag{3}$$

where $P_{u,v}$, $L_{u,v}$ and $T_{u,v}$ are the position, luminance and time metrics respectively and α, β and γ are the weightings for each metric. Each of the metrics produce a value between 0 and 1 so that the weightings for each metric can be adjusted for different scenes and to create clusters with different properties.

3.2.1. Position Metric

There are two possible spaces that can be used to represent the positions of the samples, texel space or as points on a unit sphere. It is important for the clustering algorithm that the second representation is used. This is because, in texel space, the clusters will not be able to wrap around the edge of the EM. This means that the position of sample will be a unit vector representing a point on the surface of a unit sphere. Then the angle between two samples will give the difference in position of the samples and the metric can be defined as:

$$P_{u,v} = \frac{\arccos(\overline{u} \cdot \overline{v})}{\pi} \tag{4}$$

where $\overline{u}$ and $\overline{v}$ are the unit vectors representing the positions of the samples u and v. In the above equation arccos is used as it will give the angle between the two positions, the result of which is divided by π in order to limit the results to the range $[0,1]$.

3.2.2. Luminance Metric

The luminance metric can be defined by the difference in the luminance between two samples. This can then be limited to the range $[0,1]$ by normalising. The metric is given by:

$$L_{u,v} = \frac{|Lum_u - Lum_v|}{\max_{\omega \in \Omega}(Lum_\omega)} \tag{5}$$

where Lum_u is the luminance of the sample u and Ω is the set of all samples.

3.2.3. Time Metric

Like the luminance metric the time metric is computed by taking the difference in the index of the frame that contains the samples and then normalising. The metric is given by:

$$T_{u,v} = \frac{|fr_u - fr_v|}{\Sigma fr} \tag{6}$$

where fr_u is the index of the frame that contains sample u and Σfr is the total number of frames in the VEM.

3.3. Light Source Generation

Once the clusters have been generated, representative light sources can be created. For each of the k clusters one light source is created that has the mean position of the cluster. Then each cluster, $k \in K$, the position, $P(k)$, can be computed as:

$$P(k) = \frac{\sum_{u \in k} \overline{u}}{|k|} \tag{7}$$

where $\overline{u}$ is the unit vector representing the position of sample u and $|k|$ is the number of samples in the cluster k.

As well as the position of the light source the cluster must be given a time span based on the samples that the it contains. So the light source for cluster k will appear in the frames with indices in the range $[f_{start}(k), f_{end}(k)]$, where $f_{start}(k)$ is $\min_{u \in k}(fr_u)$ and $f_{end}(k)$ is $\max_{u \in k}(fr_u)$.

Finally the radiance of the light source for each cluster is computed. It is important to use the radiance from each sample, so that each one is taken into account. The radiance, $R(k)$ of each light source is given by:

$$R(k) = \frac{\sum_{u \in k} c_u}{f_{end}(k) - f_{start}(k) + 1} \tag{8}$$

where c_u is the radiance of the sample u. The "$+1$" is in the denominator as it is possible for a cluster to contain samples from only one frame and without it the result could be undefined.

Calculating the radiance for the light sources in this way will change the total light in independent frames of the resulting animation, but the total light will remain the same. This will introduce bias to some frames, but reduces temporal flickering significantly, resulting in a smoother animation.

4. Results

Our method compares results with importance sampling. Two methods have been used for comparison:

1. A temporal inconsistency metric that computes the inconsistency of an animation by finding the difference between the rendered result and the ground truth.
2. A direct comparison of the positions of the clusters and the samples generated by importance sampling over time.

The scene that has been used for the comparison uses a VEM of the sun moving through the sky to light the Kiti church model, as shown in Figure 4. For the clustering method 512 clusters have been generated from 1024 initial samples per frame. For the weightings of the clustering metrics we used $\alpha = 3$, $\beta = 3$ and $\gamma = 1$. We compare this to importance sampling with n samples per pixel, where n is the average number of clusters that contribute to each frame. For this scene, $n = 200$. The ground truth is also computed using importance sampling with 1024 samples per pixel. All of the results are generated on a PC with an Intel(R) Core(TM) i7-2760QM quad core 2.40GHz processor using an unoptimised CPU physically-based renderer [PH10b]. Each image is 512 by 512 pixels.

4.1. Temporal Inconsistency Metric

To measure the temporal inconsistency of our method we use the metric outlined by Wan et al. [WMWL11]. The metric

Figure 2: *Top: Example frames from the rendered results of our clustering method.*

measures how similar each images are by using the mean absolute error between the rendered results and the ground truth. This is given by:

$$E(t) = \frac{1}{N}\sum_i \omega_i |\Delta X_i(t) - \Delta \hat{X}_i(t)| \quad (9)$$

where X_i is the intensity of the i in the tth rendered frame from a sampling method; $\hat{X}_i(t)$ is the counterpart of $X_i(t)$ in the ground truth; and N is the total number of pixels in the rendered frame. The operator $\Delta(.)$ is the temporal difference between two consecutive frames, so $\Delta X_i(t) = X_i(t) - X_i(t+1)$. Finally the weighting factor, ω_i, that weights the results by the difference between the rendered result and the ground truth is given by $\omega_i = |X_i(t) - \hat{X}_i(t)| + 1$. As this weighting factor directly compares illumination of a sampling method to the ground truth, this metric will also indicate the accuracy of the resulting illumination. Using this metric a rendering method can be said to be more temporally consistent if its inconsistency values are closer to zero.

Figure 3, shows the inconsistency of each frame of the two methods. It is clear that the inconsistency for our clustering method is closer to zero for all frames in the animation. As well as the inconsistency of each frame, the average inconsistency of the whole animation has been computed. The average inconsistency of the importance sampling technique is 0.00262 while the average for the clustering method is 0.00125.

4.2. Light Positions Over Time

Figure 4 shows the positions of clusters that contributed in five consecutive frames of the VEM and figure 5 the positions samples generated by importance sampling for the same frames (right column). The clustering method will produce results with less temporal flickering between frames, as the positions of the clusters do not move around the EM, while with the importance sampling method there is significant movement of samples between the frames.

5. Conclusion and Further Work

In this paper we presented an approach to improve the temporal quality of DIBL for rendering realistic animations. Our method is based on two main passes; unbiased sampling of directional lights from the VEM frames and then a clustering step to reduce flickering artifacts. Our results demonstrate a significant improvement in performance over the more straightforward method.

In the future, we plan to extend this work by experimenting with different clustering algorithms, and using more sophisticated sampling techniques [TCE05]. We intend to experiment with considering other possible dimensions and the possibility of adapting the dimensions separately to further improve the quality where it may be most required. Our method only samples the EM, however for certain lighting scenarios it would be more advantages to consider other sampling distributions, for example the BRDF. In future work we also intend to investigate multidimensional sampling methods for such challenging environments. Finally, our method is ideal for animations, and can be adapted for real-time scenarios in the future.

6. Acknowledgments

This work has been partially supported by EPSRC grant EP/I038780/1.

References

[ARBJ03] Agarwal S., Ramamoorthi R., Belongie S., Jensen H. W.: Structured importance sampling of environment maps. *ACM Trans. Graph. 22*, 3 (July 2003), 605–612. 2

[BGH05] Burke D., Ghosh A., Heidrich W.: Bidirectional Importance Sampling for Direct Illumination. *Computer Graphics Forum (Proceedings of Eurographics)* (2005). 2

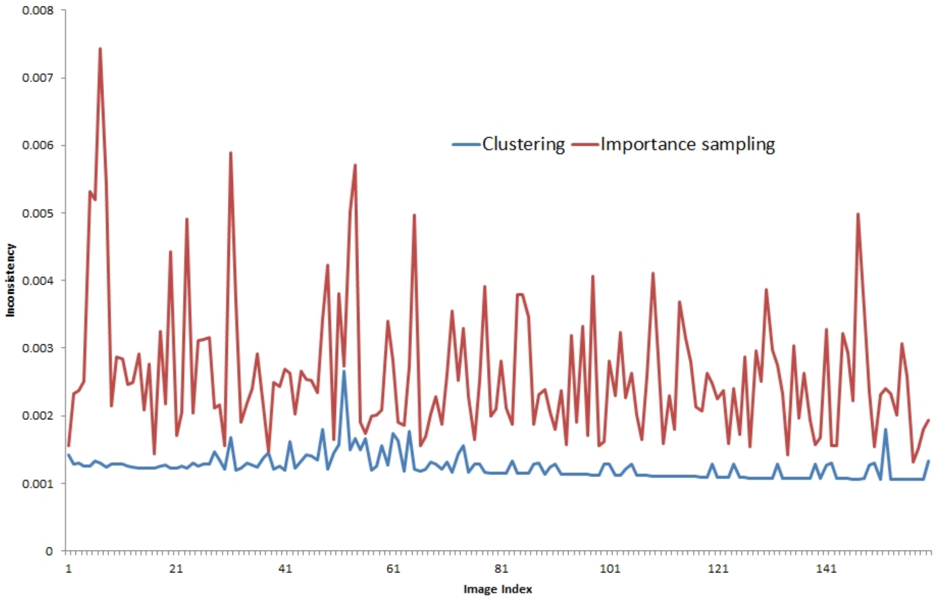

Figure 3: *For the Kiti scene, the temporal inconsistency measured compared to the ground truth.*

[BN76] BLINN J. F., NEWELL M. E.: Texture and reflection in computer generated images. *Commun. ACM 19*, 10 (Oct. 1976), 542–547. 2

[CBB*09] CHALMERS A., BONNET G., BANTERLE F., DUBLA P., DEBATTISTA K., ARTUSI A., MOIR C.: High-dynamic-range video solution. In *SIGGRAPH ASIA Art Gallery & Emerging Technologies* (2009), p. 71. 1

[DBB02] DUTRE P., BALA K., BEKAERT P.: *Advanced Global Illumination*. A. K. Peters, Ltd., Natick, MA, USA, 2002. 1

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 189–198. 2

[Deb05] DEBEVEC P.: A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2005 Posters* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 2

[For65] FORGY E.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics 21* (1965), 768–780. 3

[GDH06] GHOSH A., DOUCET A., HEIDRICH W.: Sequential sampling for dynamic environment map illumination. In *Proc. Eurographics Symposium on Rendering* (2006), pp. 115–126. 2

[HDS03] HAVRAN V., DMITRIEV K., SEIDEL H.-P.: Goniometric Diagram Mapping for Hemisphere. Short Presentations (Eurographics 2003), 2003. 2

[HSK*05] HAVRAN V., SMYK M., KRAWCZYK G., MYSZKOWSKI K., SEIDEL H.-P.: Importance Sampling for Video Environment Maps. In *Eurographics Symposium on Rendering 2005* (Konstanz, Germany, 2005), Bala K., Dutré P., (Eds.), ACM SIGGRAPH, pp. 31–42,311. 2

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph. 20*, 4 (Aug. 1986), 143–150. 2

[Nie92] NIEDERREITER H.: *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadephia, PA, 1992. 2

[PH10a] PHARR M., HUMPHREYS G.: *Physically Based Rendering from Theory to Implementation (Second Edition)*. Morgan-Kaufmann, 2010. 2, 3

[PH10b] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 4

[TCE05] TALBOT J. F., CLINE D., EGBERT P. K.: Importance resampling for global illumination. In *Rendering Techniques 2005 Eurographics Symposium on Rendering* (Aire-la-Ville, Switzerland, 2005), Bala K., Dutré P., (Eds.), Eurographics Association, pp. 139–146. 2, 5

[WMWL11] WAN L., MAK S.-K., WONG T.-T., LEUNG C.-S.: Spatiotemporal sampling of dynamic environment sequences. *IEEE Transactions on Visualization and Computer Graphics 17* (2011), 1499–1509. 2, 4

[WWL05] WAN L., WONG T., LEUNG C.: Spherical q2-tree for sampling dynamic environment sequences. In *Proc. of Eurographics Symposium on Rendering* (2005), pp. 21–30. 2

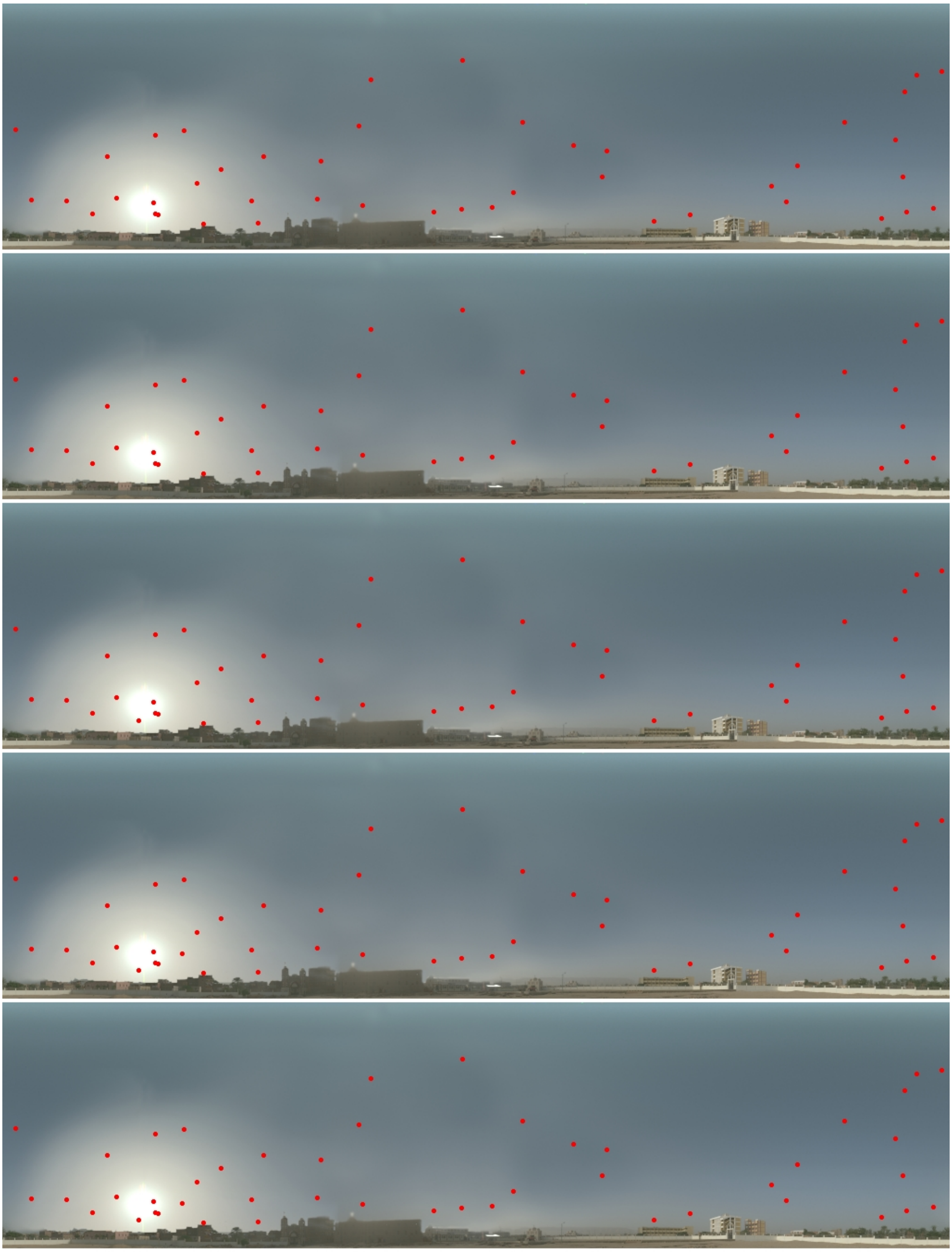

Figure 4: *Positions of the light sources generated from our clustering method shown over five consecutive frames.*

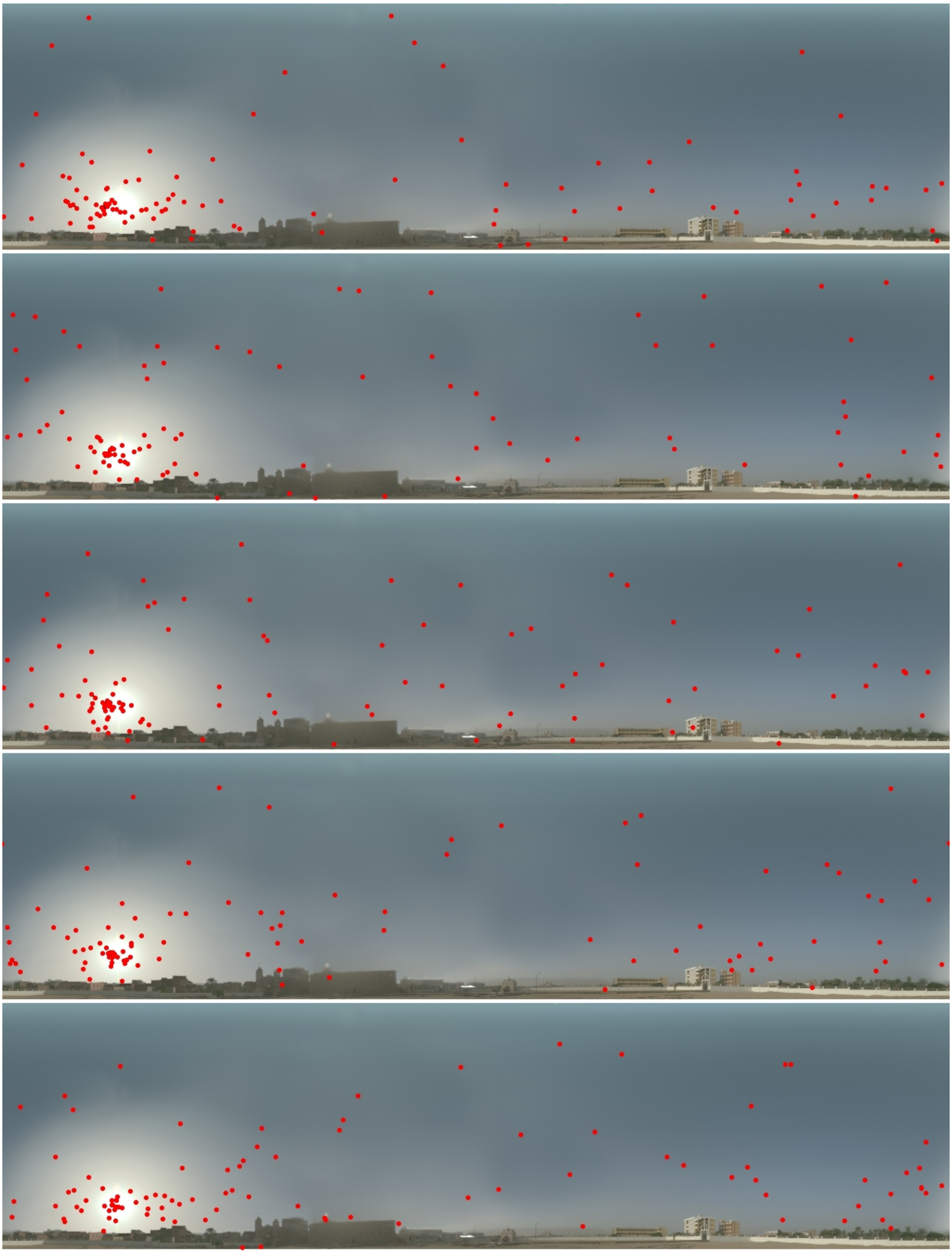

Figure 5: *Positions of samples generated by importance sampling over the same five frames used in Figure 4.*