Colour compression of plenoptic point clouds using raht-klt with prior colour clustering and specular/diffuse component separation

Maja Krivokuća, Christine Guillemot

COLOUR COMPRESSION OF PLENOPTIC POINT CLOUDS USING RAHT-KLT WITH PRIOR COLOUR CLUSTERING AND SPECULAR/DIFFUSE COMPONENT SEPARATION

Maja Krivokuća and Christine Guillemot

INRIA, Rennes, France

ABSTRACT

The recently introduced *plenoptic point cloud* representation marries a 3D point cloud with a light field. Instead of each point being associated with a single colour value, there can be multiple values to represent the colour at that point as perceived from different viewpoints. This representation was introduced together with a compression technique for the multi-view colour vectors, which is an extension of the RAHT method for point cloud attribute coding. In the current paper, we demonstrate that the best-proposed RAHT extension, RAHT-KLT, can be improved by performing a prior subdivision of the plenoptic point cloud into clusters based on similar colour values, followed by a separation of each cluster into specular and diffuse components, and coding each component separately with RAHT-KLT. Our proposed improvements are shown to achieve better rate-distortion results than the original RAHT-KLT method.

***Index Terms*—** Point clouds, light fields, colour coding, RAHT

1. INTRODUCTION

A *3D point cloud* is a set of points in 3-dimensional Euclidean space, $\mathbb{R}^3$, where each point has a spatial *position*, (x, y, z), and optionally other *attributes*, most typically colour. Recent years have seen point clouds rapidly becoming the representation of choice for 3D objects in various application areas. This is largely due to their flexibility in representing both manifold and non-manifold geometry, and their potential for real-time processing since they do not require the coding of explicit surface topology. However, the point positions (*geometry*) and *attributes* still require compression in order to be practically usable. The research community and industry's interest in point cloud compression can be seen in numerous recent publications, many of which have been incorporated into the recent MPEG Point Cloud Compression (MPEG-PCC) standardisation activity [1]. The past decade has also seen a revival of research on light field imaging [2, 3], leading to light field compression becoming an active area of research (e.g., [4]) and standardisation activities in JPEG and MPEG [5, 6]. This recent popularity of both, 3D point clouds and light fields, can be attributed to the same enduring goal: to achieve digital representations of the world around us, which are ever more realistic and more versatile, but still remaining practical to use, store, and distribute. Realising the potential of both 3D point clouds and light fields towards achieving this goal, recent publications [7, 8, 9, 10] are now considering a new direction: a point cloud representation of a light field. The *plenoptic point cloud* [8, 9] is a natural extension of a 3D point cloud to a *Surface Light Field* (SLF) [11]. Such a representation is very promising for applications that require both, the versatility and convenience of a point cloud, and the richness of visual information provided by a light field. The work in [10] also proposes a SLF on a point cloud. As shown in [8, 9, 10], these representations are amenable to efficient compression, partly because existing image and point cloud coding techniques can be extended to them fairly easily. In [8, 9], four extensions of the *Region-Adaptive Hierarchical Transform* (RAHT) method [12, 13] (which is part of the emerging MPEG-PCC standard for attribute coding [1]) are introduced to compress the plenoptic colour vectors. In [10], light field images are mapped to a point cloud to obtain a SLF, then the points' "view maps" are encoded using a B-Spline wavelet basis and compressed spatially using existing point cloud codecs.

Our focus in the current paper is the compression of the *multi-view colour vectors* in plenoptic point clouds [8, 9], assuming a *lossless geometry*. We wish to improve upon the rate-distortion (R-D) performance shown for the best-performing RAHT extension in [8, 9], RAHT-KLT. To the best of our knowledge, currently there exist no other compression methods for plenoptic colour vectors that take into account both the correlations across the camera viewpoints and the spatial correlations between the 3D points. [10] uses existing point cloud codecs (e.g., RAHT) after the B-Spline wavelets, while [11, 14, 15] propose techniques for compressing the SLF "colour maps" but without any spatial compression. We have identified two main areas for improvement in RAHT-KLT: (1) The computation of covariance matrices for the KLT requires averaging the colours across *all* the input points, even if these colours are very different; (2) There is no specific identification, or handling, of regions of higher specularity, which are the most problematic for compression. Therefore, in this paper, we propose the following new contributions:

1. The idea that a plenoptic point cloud should first be subdivided into clusters based on similar colour values. We demonstrate a simple way to do this clustering using *k-means*.
2. The idea that each cluster should be further separated into *specular* and *diffuse* components, which are encoded separately by RAHT-KLT for each cluster. We propose a way to do this separation using *Robust Principal Component Analysis* (RPCA).
3. We demonstrate that the above contributions result in better R-D performance than when applying RAHT-KLT on the *entire* plenoptic point cloud as in [8, 9].
4. We show results for RAHT-KLT on the 12-bit geometry 8iVSLF point clouds (recently contributed to MPEG as the first plenoptic point cloud dataset [7]) for the first time. (In [8, 9], lower-resolution versions of 8iVSLF were used.)

2. PLENOPTIC POINT CLOUDS AND RAHT-KLT

In 3D point clouds, colour is usually represented as one (R, G, B) triplet per point. However, for realistic representations of 3D objects that contain *specular* surfaces, where the reflected light differs depending on the viewing angle, a single colour per point is insufficient. The *plenoptic point cloud* [8, 9] overcomes this, as it is essentially a point cloud representation of the *plenoptic function* [16]:

$$P(x, y, z, \theta, \phi), \tag{1}$$

where P is the radiance of light observed from every possible viewing position (x, y, z), with every viewing angle (θ, ϕ), where θ is the azimuth and ϕ the elevation. Since the (x, y, z) points are defined directly on the surface of a 3D object, the plenoptic point cloud is equivalent to a SLF [11]. The SLF can be regarded as a function $f(\boldsymbol{w}|\boldsymbol{p})$, such that for a point $\boldsymbol{p}$ on the surface, $f(\boldsymbol{w}|\boldsymbol{p})$ represents the (R, G, B) value of a light ray starting at $\boldsymbol{p}$ and emanating outwards in direction $\boldsymbol{w}$. We thus end up with a "view map", or "colour map", for each surface point $\boldsymbol{p}$, which describes the colour of $\boldsymbol{p}$ as seen from different viewpoints. The SLF, and therefore the plenoptic point cloud, can thus be considered generalisations of the *lenslet* light field representation, to a 2D manifold embedded in 3D [10].

In the current paper, we consider a *sampled* version of the plenoptic point cloud, as in [8, 9]. That is, for each surface point $\boldsymbol{p}_i$ in a finite set of points $\{\boldsymbol{p}_i | i \in [1, N_p]\}$ in $\mathbb{R}^3$, there is a finite number of viewpoints, N_c, equal to the number of camera viewpoints used to capture the 3D object. Therefore, for a point $\boldsymbol{p}_i$ with spatial coordinates (x_i, y_i, z_i), the sampled plenoptic point cloud representation in RGB colour space is:

$$\boldsymbol{p}_i = [x_i, y_i, z_i, R_i^1, G_i^1, B_i^1, \ldots, R_i^{N_c}, G_i^{N_c}, B_i^{N_c}], \quad (2)$$

where $[R_i^1, \ldots, B_i^{N_c}]$ is the *plenoptic* or *multi-view* colour vector. In [8, 9], four extensions of the point cloud attribute coding method, RAHT [12, 13], were proposed to compress the plenoptic colour vectors. The best-performing extension, *RAHT-KLT*, begins by computing an $N_c \times N_c$ covariance matrix $\boldsymbol{\Gamma} = \{\Gamma(i, j), 1 \leq i, j \leq N_c\}$ for each colour channel C (Y, U, V channels were used in [8, 9]), where

$$\Gamma(i, j) = \frac{1}{N_p - 1} \sum_{n=1}^{N_p} (C_i(n) - \mu_C^i)(C_j(n) - \mu_C^j), \quad (3)$$

and

$$\mu_C^i = \frac{1}{N_p} \sum_{n=1}^{N_p} C_i(n). \quad (4)$$

Then, the eigenvectors of each $\boldsymbol{\Gamma}$ are computed through a *Singular Value Decomposition* (SVD) and are used to perform a *Karhunen-Loève Transform* (KLT) on each colour vector $\mathbf{c}(n) = [C_1(n), C_2(n), ..., C_{N_c}(n)]^T$ for each point n in the point cloud, for each colour channel C. The $N_p \times 3$ matrix of KLT-transformed vectors for each of the N_c viewpoints is then encoded with RAHT.

3. PROPOSED PRE-PROCESSING TO RAHT-KLT

(3) and (4) indicate that the computation of covariance matrices in RAHT-KLT requires averaging the colours across *all* N_p points in a plenoptic point cloud. However, in practice, the distribution of colours across these points is likely to be quite wide, resulting in relatively high standard deviations for their averages. Therefore, the associated KLT vectors will not fit the input data as well as they could. For these reasons, we propose that instead of applying RAHT-KLT on the *entire* plenoptic point cloud as in [8, 9], this point cloud should first be clustered into sub-clouds based on similar colour values, then each sub-cloud encoded separately with RAHT-KLT. The encoding and decoding could then be performed in parallel across the clusters.

3.1. Clustering based on similar colour values

Our idea for point cloud clustering is not limited to any particular clustering method. For the work in this paper, we use the well-known *k-means* as an example, with a squared Euclidean distance measure, *k-means++* [17] to choose the initial seeds, and a stopping criterion of 100 iterations. The value of k is chosen heuristically, to correspond roughly to the number of different colours in the input point cloud. Our input matrix to k-means has N_p rows, or "observations". For the columns, we rely on the assumption that in practice, most plenoptic point clouds are likely to represent mostly *Lambertian* [18] or near-Lambertian surfaces. Therefore, we should be able to obtain a reasonable approximation of each point's colour from the average of its colours across the N_c viewpoints. So our input matrix to k-means has 3 columns, each representing the average colour in one of the 3 colour channels. To decide which colour space to use, we tested RAHT-KLT on the plenoptic point clouds (without clustering) from the 8iVSLF *static* dataset [7], in the RGB, YUV, and HSV colour spaces. Since our goal is to improve the R-D results of RAHT-KLT, the choice of colour space depends on which space RAHT-KLT works best in. Our hypothesis was that the best colour space would be the one that has the lowest average standard deviation in colour across the N_p input points (computed as explained in Table 1's caption). Table 1 shows these standard deviations, which all have the same units, as the colour values in each colour space were scaled to be in [0, 255]. An R-D plot for *Thaidancer* coded with RAHT-KLT in different colour spaces is shown in Fig. 1 and is representative of the results for all the 8iVSLF *static* point clouds.

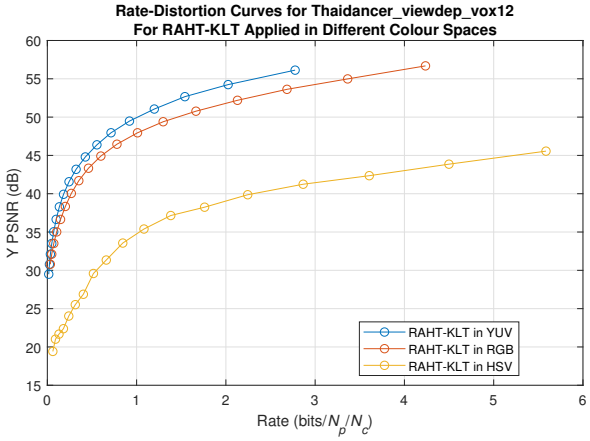

Fig. 1: RAHT-KLT applied on an entire plenoptic point cloud, using different colour spaces for the input plenoptic colour matrix.

Fig. 1 and Table 1 support our hypothesis: RAHT-KLT achieves the best R-D performance in YUV space, where the average colour standard deviation across the N_p input points is the *lowest*, and the worst performance in HSV, where this standard deviation is the highest. These observations also confirm that the performance of RAHT-KLT suffers if the colour variability across the input points is high, which motivates the need for prior clustering. Fig. 2 shows three example clusters for *Thaidancer*. We see that meaningful colour separations are produced, even when the k-means seeds are chosen semi-randomly. Moreover, the average standard deviation across the points in each cluster (see Fig. 2's caption) is lower than the corresponding standard deviation for the *entire* point cloud in the same colour space (Table 1). Section 4 will show that these smaller standard deviations lead to better R-D performance for RAHT-KLT.

3.2. Separation of specular and diffuse components

The R-D performance of RAHT-KLT suffers if the input point cloud contains highly specular regions, but the identification of such specular regions was left for future work in [8, 9]. Here we demonstrate that *Robust Principal Component Analysis* (RPCA) [19] can be used on our proposed clusters, to successfully separate diffuse and spec-

Dataset Name	No. of Input Points (N_p)	No. of Camera Viewpoints (N_c)	RGB S. D.	YUV S. D.	HSV S. D.
Thaidancer_viewdep_vox12	3,130,215	13	37.59	**23.30**	56.70
redandblack_viewdep_vox12	2,770,567	12	32.77	**19.49**	72.08
longdress_viewdep_vox12	3,096,122	12	40.09	**23.16**	56.04
soldier_viewdep_vox12	4,001,754	13	28.33	**11.21**	45.62
boxer_viewdep_vox12	3,493,085	13	33.95	**14.69**	38.06
loot_viewdep_vox12	3,017,285	13	33.86	**15.00**	35.70

Table 1: 8iVSLF *static* point clouds [7], with average colour standard deviations (S. D.) in different colour spaces. Standard deviations are computed across all N_p points, per camera viewpoint, per colour channel, then the resulting N_c standard deviations in each colour channel are averaged, and finally the average across the 3 colour channels is computed. Lowest S. D. values for each point cloud are shown in bold.

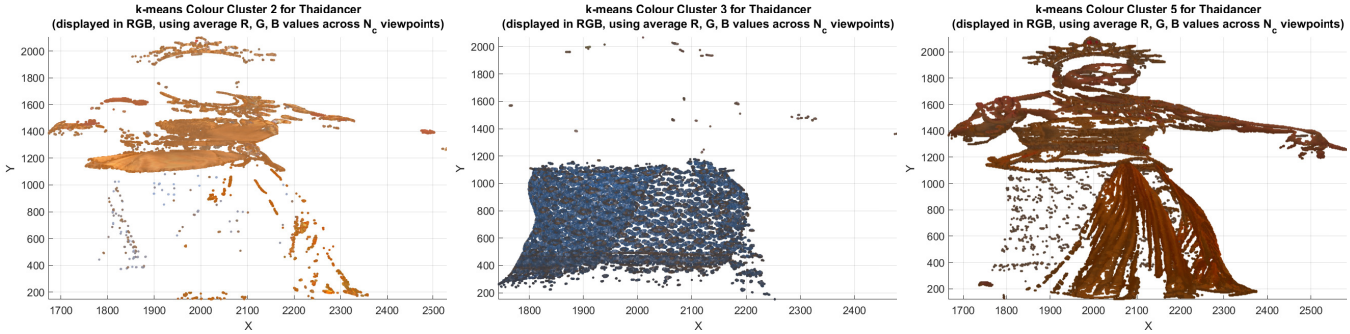

Fig. 2: 3 out of $k = 8$ clusters obtained from k-means applied on average Y, U, V values. Average YUV standard deviations for each cluster, computed similarly to Table 1 but using only the points in each cluster instead of N_p, (left to right): 11.55, 11.81, 7.35. Average (over R, G, and B components) ranks of $\mathbf{L}$ (left to right): 6, 2, 6. Average sparsities (% 0 values) of $\mathbf{S}$ (left to right): 31.73, 58.72, 26.99.

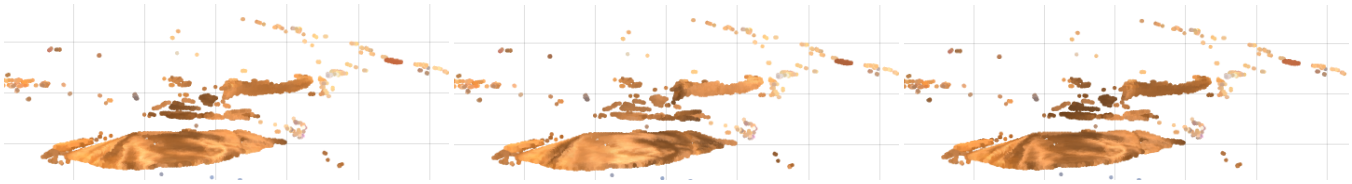

Fig. 3: Specular components extracted from *Thaidancer* Cluster 2 (from Fig. 2), shown for camera viewpoints 1, 2, and 7 (left to right).

ular components. RPCA decomposes a data matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ into a low-rank *approximation* matrix $\mathbf{L_0}$ and a sparse *error* matrix $\mathbf{S_0}$, such that $\mathbf{M}$ can be recovered as $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0$. The inverse problem, of recovering $\mathbf{L_0}$ and $\mathbf{S_0}$ from $\mathbf{M}$, can be formulated as a *Principal Component Pursuit* (PCP) optimisation problem:

$$\min_{\mathbf{L},\mathbf{S} \in \mathbb{R}^{n_1 \times n_2}} ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1 \text{ subject to } \mathbf{L} + \mathbf{S} = \mathbf{M}, \quad (5)$$

where $||.||_*$ is the *nuclear norm*, $||.||_1$ is the ℓ_1 norm (sum of all absolute values), and $\lambda = 1/\sqrt{n_{(1)}}$, where $n_{(1)} = \max(n_1, n_2)$ [19]. To solve (5), we use the *Augmented Lagrangian Multiplier* (ALM) method with the *Alternating Direction Method of Multipliers* (ADMM), similarly to [19]. The ALM method operates on the *augmented Lagrangian*,

$$\ell(\mathbf{L}, \mathbf{S}; \mathbf{Y}) = ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1 + \frac{1}{\tau} < \mathbf{Y}, \mathbf{M} - \mathbf{L} - \mathbf{S} > + \frac{\mu}{2} ||\mathbf{M} - \mathbf{L} - \mathbf{S}||_F^2, \quad (6)$$

where $||.||_F$ is the Frobenius norm, $\langle \cdot, \cdot \rangle$ is the trace inner product, $\mathbf{Y}$ is a matrix of Lagrangian multipliers, and $\tau = 1/\mu$. We use $\mu = n_1 n_2 / 4||\mathbf{M}||_1$, as in [19]. The ADMM iteratively solves a sub-problem for each matrix, $\mathbf{L}, \mathbf{S}, \mathbf{Y}$, as described in Algorithm 1 in [19]. As in [19], our stopping criterion is $||\mathbf{M}-\mathbf{L}-\mathbf{S}||_F \leq \delta ||\mathbf{M}||_F$, with $\delta = 10^{-7}$. We apply RPCA separately on each point cloud cluster proposed in Section 3.1. For a cluster with N_{pclust} points ($N_{pclust} \ll N_p$), the $\mathbf{M}$ input to RPCA is the $N_{pclust} \times N_c$ colour matrix for each colour channel (R, G, B) separately. We found that better RAHT-KLT R-D results are achieved when RPCA is applied in RGB than in YUV colour space. Our motivation for using RPCA is the assumption that a plenoptic colour matrix should be able to be decomposed into a low-rank matrix $\mathbf{L}$ that describes the *diffuse* points, where the colours do not vary (much) across the N_c viewpoints, and a sparse matrix $\mathbf{S}$ (with rank $= N_c$) that contains non-zero values where the corresponding points' colours are not fully described by $\mathbf{L}$. We thus assume that $\mathbf{S}$ will allow us to detect the locations of the *specular* points. RPCA has been applied previously for specular/diffuse separation in 2D images (e.g., [20]), but to the best of our knowledge, never before in plenoptic point clouds.

Experimentally, we have found the above assumptions to be true: we are indeed able to obtain a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ by applying RPCA on our point cloud clusters. As expected, rank($\mathbf{L}$) is higher and the sparsity of $\mathbf{S}$ (% of 0 values) is lower for clusters that contain more specular components, e.g., note the ranks of $\mathbf{L}$ and sparsities of $\mathbf{S}$ for the clusters in Fig. 2. Since Clusters 2 and 5 contain regions with more specular highlights (see the full point cloud in [7]), their $\mathbf{S}$ sparsities are lower and $\mathbf{L}$ ranks higher than Cluster 3, which contains more diffuse regions. In order to separate the specular points from the diffuse, we need to rely on threshold values to decide what constitutes a significant enough error in $\mathbf{S}$ for the corresponding point to be considered "specular". For the work in this paper, this threshold is the upper quartile (75th percentile) of the sorted sums of absolute values of $\mathbf{S}$. These sums are computed by summing the absolute values of the elements across the N_c columns for each row of $\mathbf{S}$. Rows with sums above the threshold represent the

"specular" points. We compute separate $\mathbf{S}$ thresholds for the R, G, and B colour channels, then collect the specular points selected from each to form the final set of specular points for the cluster. The remaining points are said to be "diffuse". Fig. 3 shows some examples of specular regions identified in this way, in Cluster 2 of *Thaidancer* (from Fig. 2). We see that meaningful segmentations are produced, as the chosen specular points have noticeably varying colours from different viewpoints. Section 4 will show the compression benefits of doing such a separation before applying RAHT-KLT.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We present a representative selection of our R-D results for the 8iVSLF data (Table 1), when RAHT-KLT is applied on the proposed clusters from Section 3. We assume a lossless geometry, but that the decoder knows which points belong to which cluster, so that the correct colours can be assigned to the points. In practice, this could be achieved by using the same clusters for colour and geometry coding. This would not require sending any extra signalling bits, except for the negligible cost of a flag indicating the start of a new cluster in the bitstream. This would also make the entire encoding and decoding processes parallelisable. The bitrates presented here, however, comprise only the colour bits: the RLGR-encoded RAHT coefficients and the covariance data, as in [8, 9]. The covariance data includes $N_c(N_c+1)/2$ elements (32 bits each) per Y/U/V channel, per sub-cloud. The total bitrates for RAHT-KLT applied on sub-clouds are the sums of colour bits across *all* the sub-clouds, divided by N_p and N_c. The PSNR values also account for all the sub-clouds. The same PSNR computation and RGB→YUV conversion is used as in [8, 9]. The R-D curves are obtained by exponentially varying the RAHT coefficients' quantization stepsize from 1.5 to 300.

Fig. 4 demonstrates that applying RAHT-KLT separately on point cloud clusters containing similar colour values indeed produces better R-D results than when applying RAHT-KLT on the entire plenoptic point cloud at once. When these point clouds contain highly specular regions (e.g., *Thaidancer*), Fig. 4 shows that it is further beneficial to separate each cluster into *specular* and *diffuse* sub-clouds, then encode each separately with RAHT-KLT. However, Fig. 4 also shows that when the input point cloud does *not* contain highly specular regions (e.g., all the 8iVSLF point clouds except *Thaidancer*), there are no obvious additional R-D benefits of specular/diffuse separation on top of the prior clustering. In fact, we see that for *longdress* and *redandblack*, applying RAHT-KLT separately on specular and diffuse components sometimes has a slightly worse performance than when RAHT-KLT is applied on the same clusters *without* specular/diffuse separation. This small difference is partly due to the overhead of transmitting twice as much covariance data in the specular/diffuse case, but without much quality gain. Unfortunately, we currently only have this limited plenoptic point cloud dataset to test our ideas on. Even in *Thaidancer*, the specular regions are very few, so the R-D improvement of specular/diffuse separation on top of the prior clustering is rather small. In any case, all of our results indicate that the prior clustering by colour has a noticeable positive impact on the R-D performance of RAHT-KLT.

5. CONCLUSION

In this paper, we showed that the R-D performance of the RAHT-KLT coder for plenoptic point clouds [8, 9] suffers if the colour variation across the input points is high. We demonstrated that better R-D results can be achieved if the point cloud is first subdivided into clusters based on similar colour values (e.g., by using k-means) and

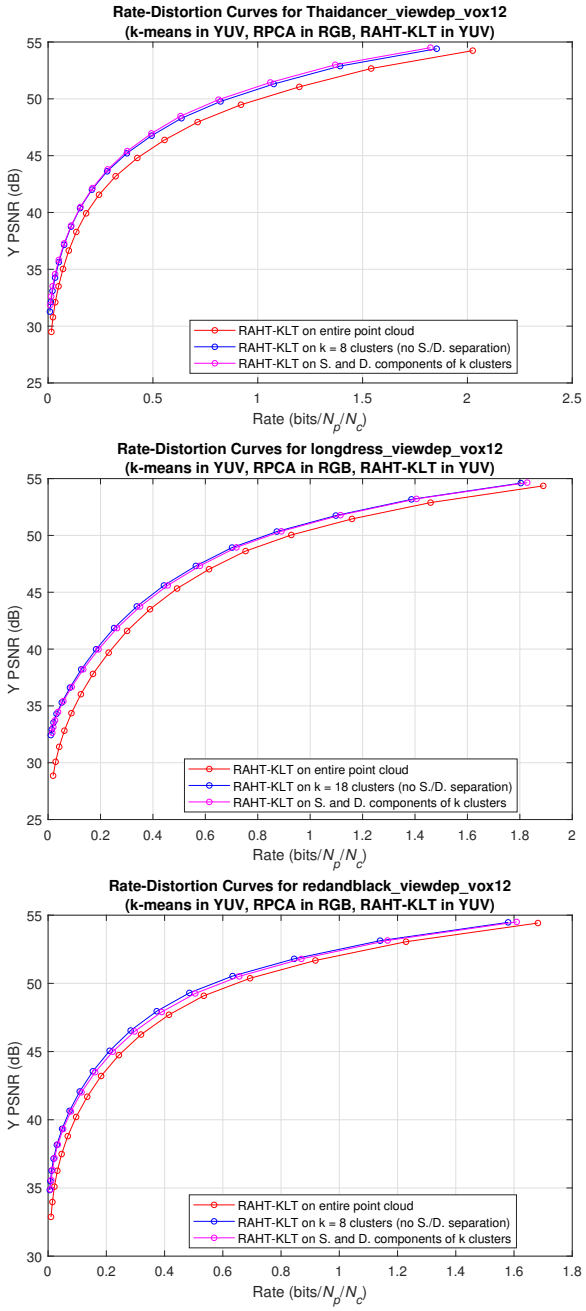

Fig. 4: Representative selection of R-D results for the 8iVSLF *static* dataset. S. and D. stand for *specular* and *diffuse*, respectively.

RAHT-KLT is applied on each cluster separately. We also proposed a method to separate the *specular* and *diffuse* points in each cluster, by using RPCA, and showed that for point clouds containing highly specular regions, applying RAHT-KLT on the specular and diffuse sub-clouds separately further improves the R-D results.

6. ACKNOWLEDGEMENT

This work has been funded by the EU H2020 Research and Innovation Programme under grant agreement No. 694122 (ERC advanced grant CLIM). The authors would also like to thank Gustavo Sandri and Ricardo de Queiroz, for providing us with the RAHT-KLT code.

7. REFERENCES

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, Mar. 2019.

[2] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging: Briefly revisiting 25 years of research," *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, Sept. 2016.

[3] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, Oct. 2017.

[4] I. Viola, M. Řeřábek, and T. Ebrahimi, "Comparison and evaluation of light field image coding approaches," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1092–1106, Oct. 2017.

[5] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an efficient representation of visual reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, Oct. 2016.

[6] M. Domański, O. Stankiewicz, K. Wegner, and T. Grajek, "Immersive visual media - MPEG-I: 360 video, virtual navigation and beyond," in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, May 2017, pp. 1–9.

[7] M. Krivokuća, P. A. Chou, and P. Savill, "8i voxelized surface light field (8iVSLF) dataset," input document m42914, ISO/IEC JTC1/SC29 WG11 (MPEG), Ljubljana, Slovenia, Jul. 2018.

[8] G. Sandri, R. de Queiroz, and P. A. Chou, "Compression of plenoptic point clouds using the region-adaptive hierarchical transform," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 1153–1157.

[9] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Compression of plenoptic point clouds," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1419–1427, Mar. 2019.

[10] X. Zhang, P. A. Chou, M. Sun, M. Tang, S. Wang, S. Ma, and W. Gao, "Surface light field compression using a point cloud codec," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 163–176, Mar. 2019.

[11] G. Miller, S. Rubin, and D. Ponceleón, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques*, pp. 281–292. Springer, Vienna, 1998.

[12] R. L. de Queiroz and P. A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, Aug. 2016.

[13] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Comments on 'Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform'," *ArXiv:1805.09146v1 [eess.IV]*, May 2018.

[14] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3d photography," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, Jul. 2000, pp. 287–296.

[15] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 447–456, Jul. 2002.

[16] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, pp. 3–20. MIT Press, Cambridge, MA, 1991.

[17] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, 2007, SODA '07, pp. 1027–1035, Society for Industrial and Applied Mathematics.

[18] S. J. Koppal, "Lambertian reflectance," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed., pp. 441–443. Springer US, Boston, MA, 2014.

[19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, May 2011.

[20] J. Guo, Z. Zhou, and L. Wang, "Single image highlight removal with a sparse and low-rank reflection model," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., pp. 282–298. Springer International Publishing, Cham, 2018.