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

Maja Krivokuća and Christine Guillemot

*(presented by Maja Krivokuća)*

***SIROCCO***
**INRIA Rennes – Bretagne Atlantique, France**

ICASSP2020, May 2020

# What is a 3D Point Cloud?

- A set of points in 3D Euclidean space ($\mathbb{R}^3$), where each point has a *position* defined by its $(x,y,z)$ coordinates (i.e., its *geometry*) and, optionally, other *attributes*, most typically colour (R, G, B).

(Provided by 8i)

- Both *geometry* and *attributes* require compression in order for the point cloud to be efficiently stored, distributed, and used in practice.
- Point clouds are rapidly becoming the representation of choice for 3D objects/scenes in many application areas, largely due to their flexibility in representing both manifold and non-manifold surface geometry, since there is no explicit surface topology that needs to be encoded.
- The recent and still ongoing MPEG-PCC (Point Cloud Compression/Coding) standardisation activity [1] has greatly contributed to this recent popularity of point clouds and to the popularity of research into point cloud compression.

[1] S. Schwarz, et al., “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.

# What is a Light Field?

- A representation of all the observed light rays in free space, described by the *plenoptic function* [2]:

$$P(x, y, z, \theta, \phi),$$

  where $P$ is the radiance of light observed from every possible viewing position $(x, y, z)$, with every possible viewing angle $(\theta, \phi)$ (azimuth, elevation).
- In practice, the parameters of $P$ are sampled.
- When the points $(x, y, z)$ are defined directly on the surface of a 3D object, we have a *Surface Light Field* (SLF) representation.
- The SLF can be regarded as a function $f(w|p)$, such that for a point $p$ on the surface, $f(w|p)$ represents the (R, G, B) value of a light ray starting at $p$ and emanating outwards in direction $w$.
- We thus end up with a “view map”, or “colour map”, for each surface point $p$, which describes the colour of $p$ as seen from different viewpoints.
- The past decade has seen a revival of research on light field imaging, mainly due to the relatively recent availability of consumer light field capture systems. This has consequently led to light field compression becoming an active area of research [3], as well as an active part of both JPEG and MPEG standardisation activities [4, 5].


[2] 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.

[3] 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.

[4] 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.

[5] 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.

# What is a Plenoptic Point Cloud?

- It is essentially a *point cloud* representation of a *Surface Light Field*.
- Usually in a 3D point cloud, colour is represented as *one* (R, G, B) triplet per point. But 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.
- For this reason, in the recently-introduced *plenoptic point cloud* representation [6, 7], the geometry and RGB colours for each surface point $p_i$ in a finite set of points $\{p_i | i \epsilon [1, N_p]\}$ in $\mathbb{R}^3$, are represented as the following vector:

$$p_i = [x_i, y_i, z_i, R_i^1, G_i^1, B_i^1, \dots, R_i^{N_c}, G_i^{N_c}, B_i^{N_c}],$$

  where $N_c$ is a finite number of camera viewpoints used to capture the 3D object, and $[R_i^1, G_i^1, B_i^1, \dots, R_i^{N_c}, G_i^{N_c}, B_i^{N_c}]$ is the *plenoptic* or *multi-view colour vector*, which describes the colour values of point $p_i$ as seen from each of the $N_c$ viewpoints.
- 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.

[6] 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.

[7] 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.

# Compressing the Plenoptic Colour Vectors

- The plenoptic point cloud representation [6, 7] was introduced together with four extensions of the well-known *Region-Adaptive Hierarchical Transform* (RAHT) [8, 9] method (which is part of the upcoming MPEG G-PCC standard for point cloud attribute coding), to efficiently compress the plenoptic colour vectors.
- The best-performing RAHT extension, **RAHT-KLT**, works as follows:
  1. An $N_c \times N_c$ covariance matrix is computed for each colour channel $C$ (Y, U, V colour channels were used in [6, 7], transformed from the original (R, G, B) space).
  2. The eigenvectors of each covariance matrix are computed through a *Singular Value Decomposition* (SVD).
  3. These eigenvectors are used to perform a *Karhunen-Loève Transform* (KLT) on each colour vector $\boldsymbol{c}(n) = [C_1(n), C_2(n), \ldots, C_{N_c}(n)]^T$ for each point $n$ in the point cloud, for each colour channel $C$.
  4. The $N_p \times 3$ matrix of KLT-transformed vectors for each of the $N_c$ viewpoints is then compressed spatially with RAHT.


[6] 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.

[7] 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.

[8] 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.

[9] 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.

# Our Chosen Problems to Solve

- Our focus in this paper is the **compression of the multi-view colour vectors** in plenoptic point clouds [6, 7], assuming a **lossless geometry**.
- In particular, **we wish to improve upon the rate-distortion (R-D) performance shown for RAHT-KLT** in [6, 7].
- We have identified two main areas for improvement in RAHT-KLT:
  1. The computation of covariance matrices currently 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.
  2. The R-D performance of RAHT-KLT was shown in [6, 7] to suffer if the input point cloud contains highly specular regions, where the colour of each point varies noticeably across the different viewpoints. But the identification and handling of such specular regions was left for future work in [6, 7].

[6] 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.

[7] 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.

# Our Contributions

- We show that the rate-distortion (R-D) performance of the RAHT-KLT coder [6, 7] suffers if the colour variation across the input points is high.
- We therefore propose that instead of applying RAHT-KLT on the entire plenoptic point cloud as in [6, 7], this point cloud should first be clustered into sub-clouds based on similar colour values, then each sub-cloud encoded separately with RAHT-KLT. We demonstrate a simple way to do this clustering by using *k-means*.
- We propose that each colour cluster should be further separated into *specular* and *diffuse* components, which are also encoded separately by RAHT-KLT. We propose a way to do this separation using *Robust Principal Component Analysis* (RPCA).
- We demonstrate that the above contributions result in better rate-distortion performance than when applying RAHT-KLT on the entire plenoptic point cloud as in [6, 7].
- We show, for the first time, rate-distortion results for RAHT-KLT on the 12-bit geometry 8iVSLF (*8i Voxelized Surface Light Field*) *static* point clouds, which were recently contributed to MPEG as the first plenoptic point cloud dataset [10]. (In [6, 7], lower-resolution versions of 8iVSLF were used.)

[6] 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.

[7] 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] 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.

# Plenoptic Point Cloud Dataset Used for Testing: 8iVSLF Static Point Clouds

*boxer_viewdep_vox12*

*longdress_viewdep_vox12*

*loot_viewdep_vox12*

*redandblack_viewdep_vox12*

*soldier_viewdep_vox12*

*Thaidancer_viewdep_vox12*

| Dataset | No. of Input Points ($N_p$) | No. of Camera Viewpoints ($N_c$) |
|---|---|---|
| *boxer* | 3,493,085 | 13 |
| *longdress* | 3,096,122 | 12 |
| *loot* | 3,017,285 | 13 |
| *redandblack* | 2,770,567 | 12 |
| *soldier* | 4,001,754 | 13 |
| *Thaidancer* | 3,130,215 | 13 |

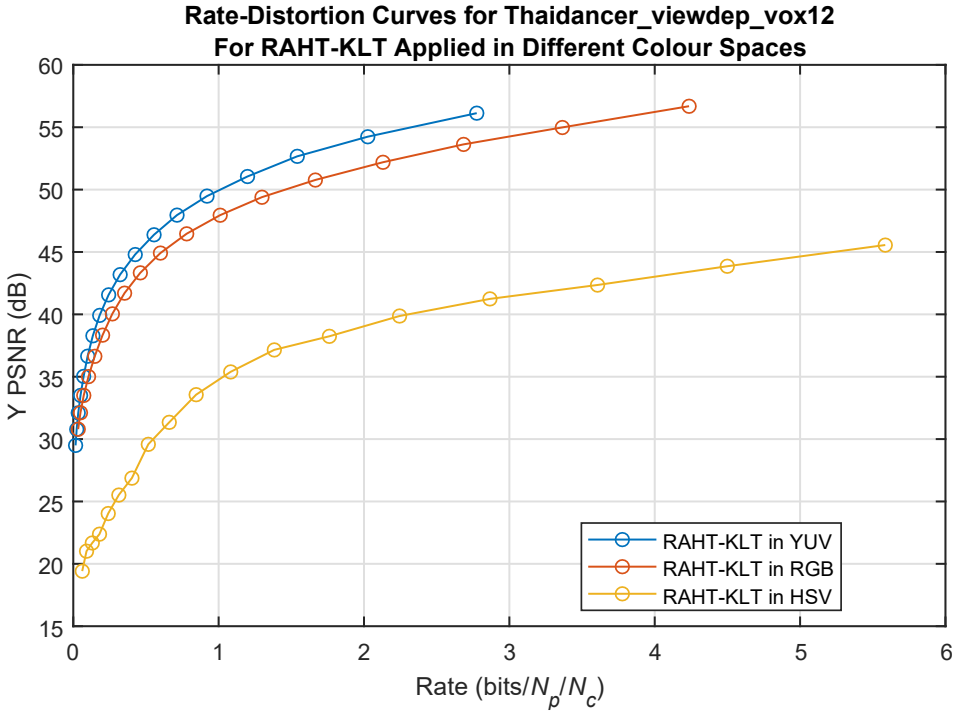
# Proposed Colour-Based Clustering

- Not limited to any particular clustering method. For the work in this paper, we use *k-means* as an example, with a squared Euclidean distance measure, *k-means*++ [11] 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 (one row per input point in the point cloud), and 3 columns (average colour in each of the 3 colour channels).
- For the columns, we rely on the assumption that in practice, most plenoptic point clouds are likely to represent mostly *Lambertian* [12] or near-Lambertian surfaces, so 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.
- To decide which colour space to use for the input plenoptic colour matrix, we first tested RAHT-KLT on the 8iVSLF point clouds [10] (without clustering), in 3 commonly used colour spaces: RGB, YUV, and HSV ...

[10] 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.
[11] 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.
[12] S. J. Koppal, "Lambertian reflectance," in Computer Vision: A Reference Guide, K. Ikeuchi, Ed., pp. 441–443. Springer US, Boston, MA, 2014.

# Proposed Colour-Based Clustering

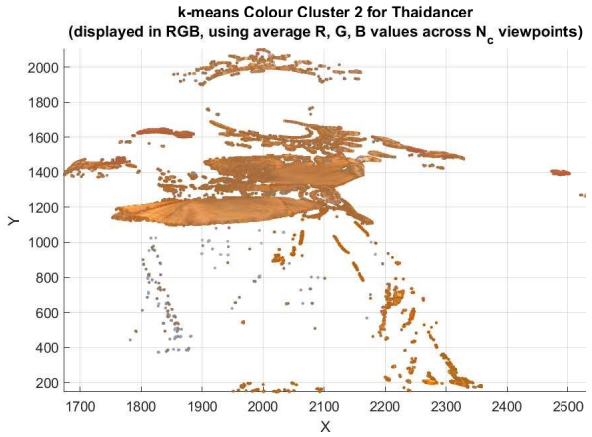
- Our hypothesis: The colour space where RAHT-KLT achieves the best R-D results would be the one that has the *lowest* average standard deviation in colour across the $N_p$ input points.
- Our experiments showed that this hypothesis is true: RAHT-KLT achieves the best R-D performance in YUV space, where 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.

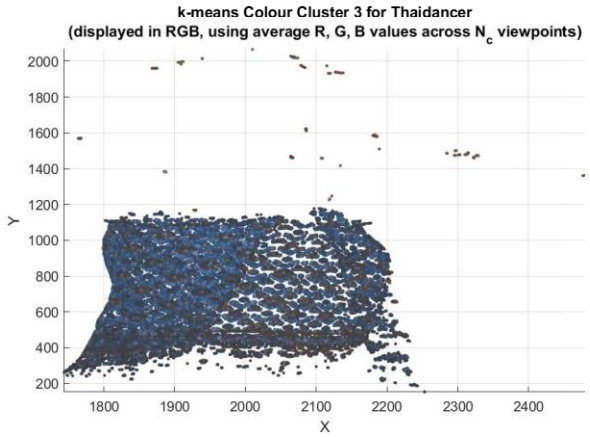


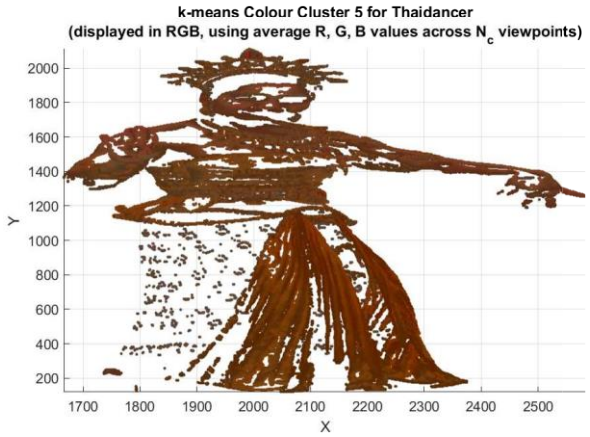

| Dataset | RGB Std. Dev. | YUV Std. Dev. | HSV Std. Dev. |
|---|---|---|---|
| *boxer* | 33.95 | **14.69** | 38.06 |
| *longdress* | 40.09 | **23.16** | 56.04 |
| *loot* | 33.86 | **15.00** | 35.70 |
| *redandblack* | 32.77 | **19.49** | 72.08 |
| *soldier* | 28.33 | **11.21** | 45.62 |
| *Thaidancer* | 37.59 | **23.30** | 56.70 |

# Proposed Colour-Based Clustering

- Some example clusters for *Thaidancer_viewdep_vox12* are shown below.
- 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 is lower than the corresponding standard deviation for the entire point cloud in the same colour space.
- Later we will show that these smaller standard deviations lead to better rate-distortion performance for RAHT-KLT.




*3 out of k = 8 clusters obtained from k-means applied on average Y, U, V values.*
*Average YUV standard deviations for each cluster, using only the points in each cluster, (left to right): 11.55, 11.81, 7.35.*
*Compare to average YUV standard deviation across all $N_p$ input points of Thaidancer = 23.30.*

# Proposed Specular/Diffuse Component Separation with RPCA

- We demonstrate that *Robust Principal Component Analysis* (RPCA) [13] can be used on our proposed colour clusters, to successfully separate the specular and diffuse components.
- RPCA decomposes a data matrix $\mathbf{M} \epsilon \mathbb{R}^{n_1 \times n_2}$ 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:

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

where $\|\mathbf{L}\|_*$ is the *nuclear norm* of $\mathbf{L}$ (sum of singular values of $\mathbf{L}$), $\|\mathbf{S}\|_1$ is the $\ell_1$ norm of $\mathbf{S}$ (sum of all absolute values in matrix $\mathbf{S}$), and $\lambda = 1/\sqrt{n_{(1)}}$, where $n_{(1)} = \max(n_1, n_2)$.

- We solve the PCP problem by using the *Augmented Lagrangian Multiplier* (ALM) method with the *Alternating Direction Method of Multipliers* (ADMM), similarly to [13]. (Please see our paper and [13] for mathematical details of these methods.)

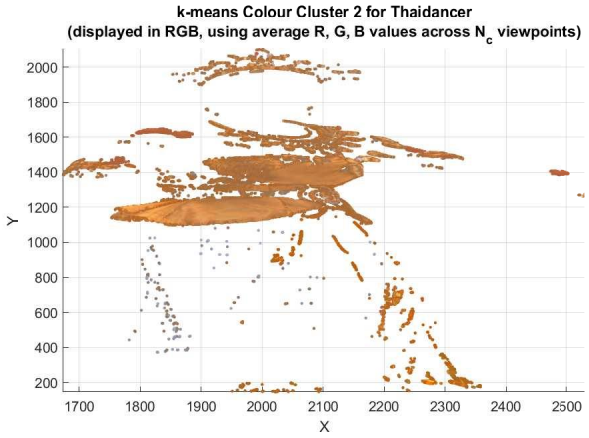
[13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," Journal of the ACM, vol. 58, no. 3, pp. 11:1–11:37, May 2011.

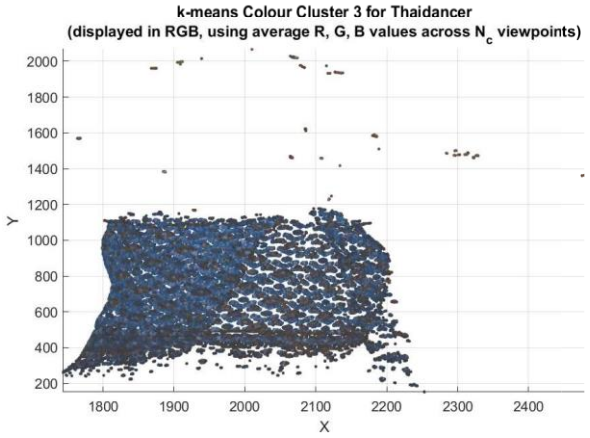
# Proposed Specular/Diffuse Component Separation with RPCA

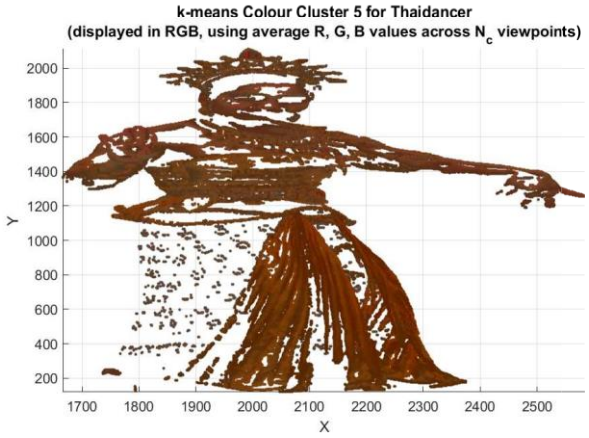
- We apply RPCA separately on each of our proposed colour-based point cloud clusters. For a cluster with $N_{pclust}$ points ($N_{pclust} \ll N_p$), the $\mathbf{M}$ input to RPCA is the $N_{pclust} \times N_c$ colour matrix for each colour channel (R, G, B) separately.
- We found that better R-D results for RAHT-KLT are achieved when RPCA is applied in RGB colour space than in YUV, even though RAHT-KLT itself is applied in YUV space.
- To separate the specular points from the diffuse, we must 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 in $\mathbf{S}$ (summed across the $N_c$ columns for each row of $\mathbf{S}$).
- We compute separate $\mathbf{S}$ thresholds for the R, G, and B colour channels in each cluster. Rows with sums above the threshold in any colour channel represent the "specular" points in the cluster. The remaining points are said to be "diffuse", meaning that for these points the colours do not vary (much) across the $N_c$ viewpoints.

# Proposed Specular/Diffuse Component Separation with RPCA

- Experimentally, we have found our 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 regions. In the examples below, since Clusters 2 and 5 contain regions with more specular highlights, their $\mathbf{S}$ sparsities are lower and $\mathbf{L}$ ranks higher than Cluster 3, which contains more diffuse regions.



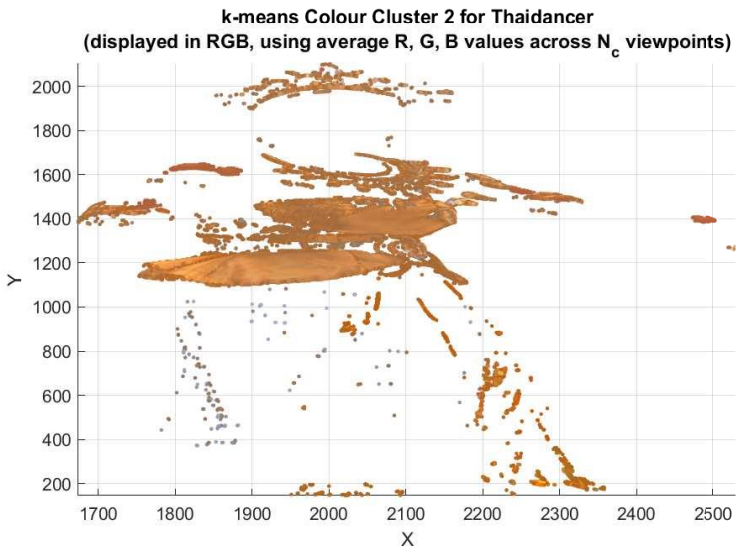



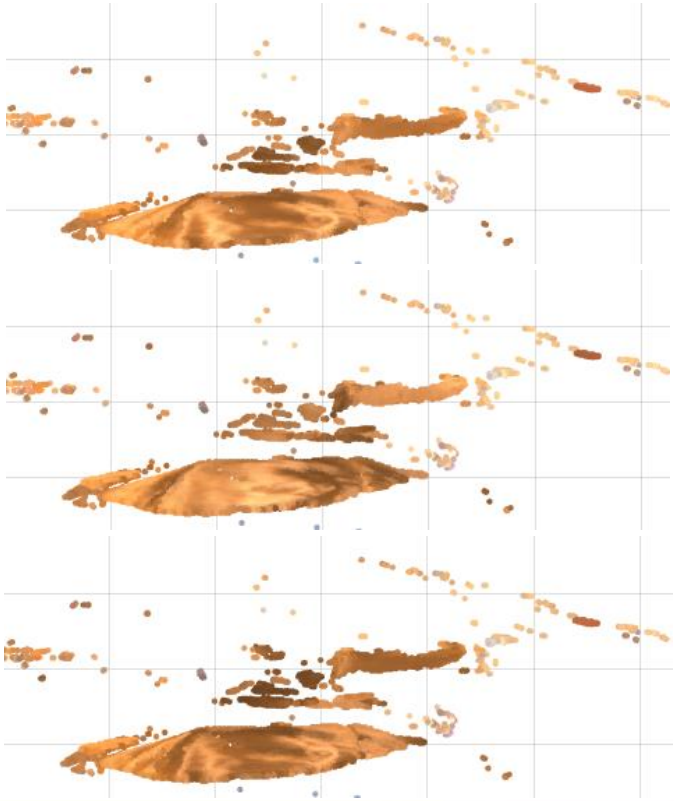




*3 out of k = 8 clusters obtained from k-means applied on average Y, U, V values.*
*Average (over R, G, and B components) ranks of $\mathbf{L}$ (left to right): 6, 2, 6.*
*Average sparsities (% 0 values) of $\mathbf{S}$ (left to right): 31.73, 58.72, 26.99.*

# Proposed Specular/Diffuse Component Separation with RPCA

- Examples of specular regions identified in Cluster 2 of *Thaidancer* are shown below, for 3 different camera viewpoints.
- We see that meaningful segmentations are produced, as the chosen specular points have noticeably varying colours from different viewpoints.
- Next we will show the compression benefits of doing such a separation before applying RAHT-KLT.



Viewpoint 1

Viewpoint 2

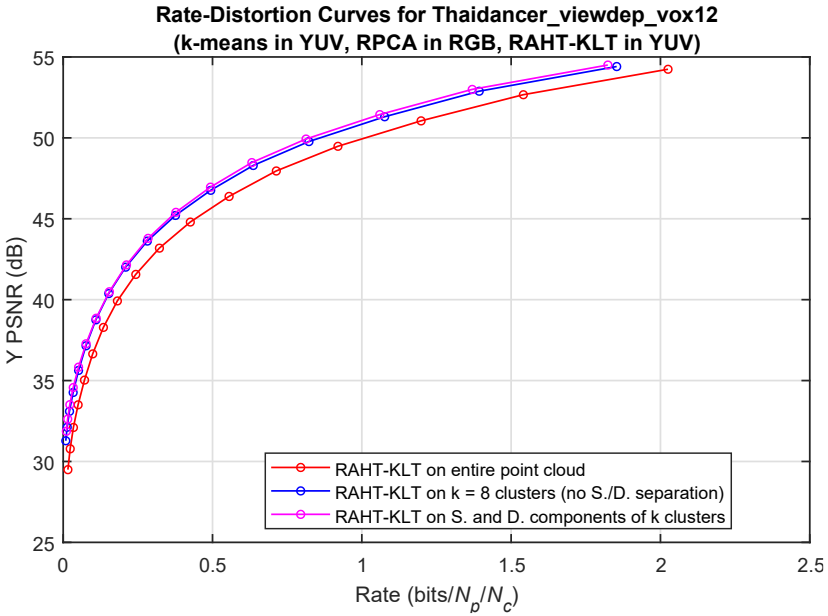
Viewpoint 7

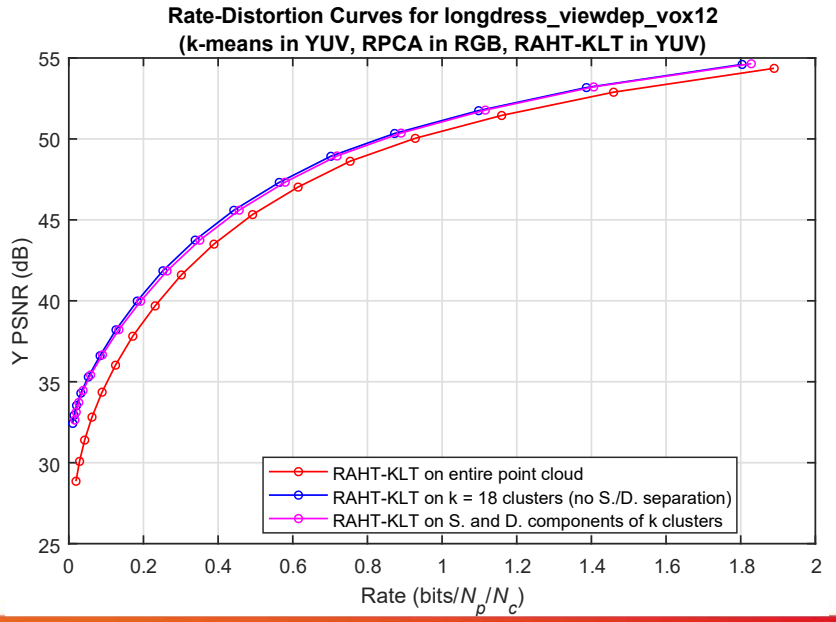
# Representative Rate-Distortion Results

- 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. This was shown consistently in all our tests.

- When these point clouds contain highly specular regions (e.g., *Thaidancer*), it is further beneficial to separate each cluster into *specular* and *diffuse* sub-clouds, then encode each separately with RAHT-KLT.

- When the input point cloud does *not* contain highly specular regions, there are no obvious additional R-D benefits of specular/diffuse separation on top of the prior clustering.



# Conclusions

- In this paper, we showed that the **rate-distortion performance of the RAHT-KLT coder** for colour compression of plenoptic point clouds [6, 7] **suffers if the colour variation across the input points is high**.

- We demonstrated that **better rate-distortion 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 **RAHT-KLT is applied on each cluster separately**.

- We 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**.

- We showed, **for the first time, rate-distortion results for RAHT-KLT on the 12-bit geometry 8iVSLF *static* point clouds**, which were recently contributed to MPEG as the first plenoptic point cloud dataset [10].

[6] 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.
[7] 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] 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.

# Acknowledgements


- 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.

**THANK YOU FOR YOUR ATTENTION.**

**QUESTIONS WELCOME.**

# Rate-Distortion Results: Experimental Details

- 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 in our paper comprise only the colour bits: the RLGR-encoded RAHT coefficients and the covariance data, as in [6, 7].
- The covariance data includes $Nc(Nc + 1)/2$ 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 $Np$ and $Nc$. The PSNR values also account for all the sub-clouds.
- The same PSNR computation and RGB→YUV conversion is used as in [6, 7].
- The R-D curves are obtained by exponentially varying the RAHT coefficients' quantization stepsize from 1.5 to 300.

[6] 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.
[7] 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.