ISSN 1213-6972 Journal of WSCG Vol.25, 2017
No.2
[Extracted from http://wscg.zcu.cz/WSCG2017/!_2017_Journal_WSCG-No-2.pdf]

Multiview Layered Depth Image

Rafael dos Anjos
FCSH/IST
Av de Berna 26, 1.24
Portugal (1069-061), Lisbon
rafael.kuffner@fcsh.unl.pt

João Madeiras Pereira
INESC-ID
R. Alves Redol 9
Portugal(1000-029), Lisbon
jap@inesc-id.pt

José António Gaspar
ISR
Av. Rovisco Pais 1
Portugal (1049-001), Lisbon
jag@isr.tecnico.ulisboa.pt

Carla Fernandes
FCSH
Av de Berna 26, 1.24
Portugal (1069-061), Lisbon
fcar@fcsh.unl.pt

ABSTRACT

Layered Depth Images (LDI) compactly represent multiview images and videos and have widespread usage in image-based rendering applications. In its typical use case scenario of representing a scanned environment, it has proven to be a less costly alternative than separate viewpoint encoding. However, higher quality laser scanner hardware and different user interaction paradigms have emerged, creating scenarios where traditional LDIs have considerably lower efficacy. Wide-baseline setups create surfaces aligned to the viewing rays producing a greater amount of sparsely populated layers. Free viewpoint visualization suffers from the variant quantization of depths on the LDI algorithm, reducing resolution of the dataset in uneven directions. This paper presents an alternative representation to the LDI, in which each layer of data is positioned in different viewpoints that coincide with the original scanning viewpoints. A redundancy removal algorithm based on world-space distances as opposed to to image-space is discussed, ensuring points are evenly distributed and are not viewpoint dependent. We compared our proposed representation with traditional LDIs and viewpoint dependent encoding. Results showed the multiview LDI (MVLDI) creates a smaller number of layers and removes higher amounts of redundancy than traditional LDIs, ensuring no relevant portion of data is discarded in wider baseline setups.

Keywords

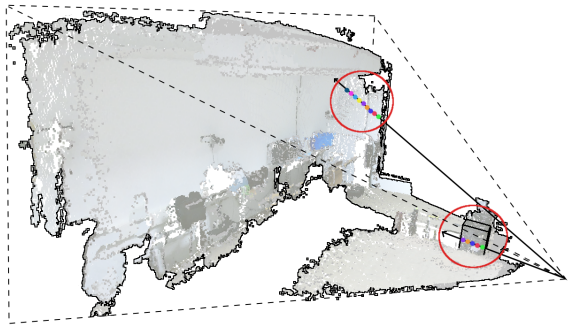
Video-based rendering, Image-based representation, Point clouds.

1 INTRODUCTION

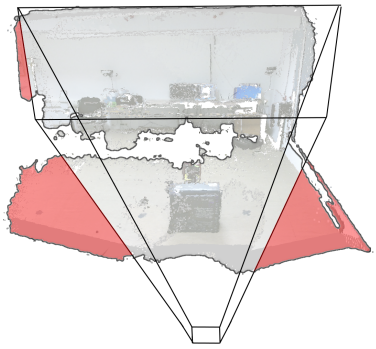
Image-based representations for rendering were initially introduced as more efficient alternatives to render geometrically complex scenarios. More recently, they have been tightly connected to Video-based rendering (VBR), a growing field that has applications with high rendering and storing requirements. Multiview+depth (MVD) encoding and Layered Depth Images (LDI) [11] have been the most popular approaches in these scenarios, with LDI being a reportedly less costly representation [13]. They are popular in this scenario due to the fact that, being image-based, they can easily incorporate advances in video compression algorithms.

However, both of the proposed representations are targeted for applications with a predefined user paradigm (3DTV, head-face parallax) and a preferably narrow-baseline capture setup. Advances in 3D capturing technology have enabled developers to more accurately represent the real world, enabling less restrictive applications to emerge [10] mainly in the fields of virtual and mixed reality. While recent work has been focused on coding tools for MVD, which can be applied to these new scenarios, we consider an alternative representation for a single frame, enabling higher compression from the start of the process.

The key advantage of the LDI representation over its alternatives is the fact that redundancy between views can be minimized during the encoding process [4]. However, its classical representation of a central + residual viewpoints establishes a main viewing direction in which data can be optimally visualized, having disadvantages in a free camera navigation scenario. This has a direct effect in the redundancy computation, meaning that the sampling rate of the data is lower in the direction of the optical rays coming from the central viewpoint (Section 3.1). Moreover, on wide baseline scenarios where cameras might be in an opposite direction to the central viewpoint, it might not be possible for all data to be captured by one chosen central viewpoint (Figure 1b), requiring a more distant virtual viewpoint to be generated which, besides not being a trivial task, will decrease the sampling rate of the data. Lastly, parallel surfaces to the central viewpoint optical rays are intersected once in each layer, increasing the number



(a) Surfaces parallel to an optical ray creates low populated additional layers.



(b) Choice of LDI central viewpoint might not encode part of the data.

Figure 1: Problems with the classical LDI representation

of low populated residual layers in order to encode the whole dataset (Figure 1).

We propose a novel image-based representation and encoding algorithm which successfully answers the aforementioned problems: the Multiview Layered Depth Image (MVLDI), where each layer is encoded according to a different viewpoint among the several capturing positions. By utilizing a modified view-generation algorithm, we have better redundancy detection and a smaller number of generated layers, while being able to correctly encode data for a wide-baseline scenario, which was only possible using MVD and no redundancy estimation.

We tested our technique with different datasets, camera setups and redundancy detection techniques, always achieving higher compression rates than the alternatives (MVD, LDI) without discarding necessary data. MVLDI is shown to provide a more efficient representation for a single frame while being applicable to video scenarios, and correctly incorporates the advances in video compression technology and depth encoding.

This article reviews related work in image-based representations, followed by a description of the MVLDI generation algorithm and redundancy detection. Finally, results are presented with an in-depth comparison between our approach and the current image-based representations.

2 RELATED WORK

Regarding image-based representations for video-based rendering or depth image-based rendering, two main lines of research can be identified: multiview+depth coding and layered depth images or videos. This section will describe relevant work in both areas.

Merkle et al. [8] and Smolic et al. [12] introduce multiview+depth coding, where depth data is associated with the video and encoded as a video stream. Although compression artifacts can be found in the rendered results, the authors claim that intermediate views are more easily generated at the user side with full data. More recent work by Do et al. [2] proposes an inpainting algorithm which has a fast GPU implementation where holes are filled with the average in a 5x5 neighborhood.

Recent developments in this area have been related to depth coding; how to avoid the loss of precision due to compression, and how to use inter frame relations between depth values make better use of predictive frames. The two main strategies are independent [6] and texture-assisted [5] depth coding. Kim et al. [3] propose a method to evaluate how depth coding influences the quality of the results.

Merkle et al. [7] introduces a plane fitting approximation to simplify blocks of data, segmenting by contour. This can be used to extract meshes (using Delaunay triangulation on the edges of the contour) and to simplify data by simplifying geometric information, thus having a better encoding of the data.

Despite allowing effective 3D representations, multiview+depth coding implies dense representations not considering the redundancy always present when multiple cameras image the same scenario. The LDI concept was introduced to allow saving data by reducing redundancy.

Yoon et al. [13] applies the LDI concept for VBR, proving it to be more compact than standard multiview coding. The data size of the multiview video linearly increases with the number of cameras. The authors suggest using the LDI representation to compress and transmit this data, mainly due to the fact that any redundant data seen by more than one point of view is not transmitted. All optimizations applied to MVD can be applied to LDIs, while not including repeated points, a claim that is supported by Kirshantan et al. [4]. In their following work [14], improvements to the LDI representation are proposed, in particular layer aggregation and layer filling, so temporal coding has a better performance.

Muller et al. [9] discuss image-based representations in the context of 3DV systems and head motion parallax as an interaction paradigm. The authors claim that for stereo video, V+D is enough, while for several views, multiview+depth where only a subset of views with depth would be transmitted and intermediate views synthesized at the receiver side. They then introduce a different concept of LDI which is focused on 3DV systems. Instead of transmitting all the views, they transmit a central view close to the desired by the user, and residual information from side views to correct errors.

More recent work has focused on proposing better encoding for residual layers. Daribo and Saito [1] propose a different residual layer estimation algorithm which includes inpainting and hole-filling. Also, Kirshantan et al. [4] propose efficient encoding for this residual information including pre-processing the data for easier layer generation.

In our work we propose a novel image-based representation and encoding algorithm, which varies the viewpoint among the several capturing positions, in order to allow larger acquisition and visualization baselines.

3 DESCRIPTION

The LDI is represented by a set of M layers $L_{ldi} = \{l_1, l_2, ..., l_M\}$ with each layer being an image-based representation of the world as seen from the same chosen "central viewpoint" v_c among the set of acquisition viewpoints of the data $V = \{v_1, v_2, ..., v_N\}$.

We propose a different representation, where one MVLDI will consist of a set of M layers $L_{mvldi} = \{l_1, l_2, ..., l_M\}$ where each layer l_i represents one of the viewpoints v_i in $V = \{v_1, v_2, ..., v_N\}$. Each viewpoint v_i has its own intrinsic and extrinsic calibration parameters in order to generate the layer information.

The number of layers M has no direct relation to the N in the case of the LDI, while in our approach, M is typically equal to N, only being higher in the case of camera calibration misalignment.

The classical representation for LDI has two disadvantages in wider baseline capture scenarios, as exemplified in Figure 1. Encoding of parallel surfaces to optical rays, and excluding data from the process due to the central viewpoint choice. This does not happen in MVD encoding, due to the fact that encoding is performed according to different points of view. No data is discarded since it is seen at least once by its original recording viewpoint, and unpopulated layers are not created by parallel surfaces due to the fact that each subsequent layer will have optical rays in different directions. Our proposed representation MVLDI combines the advantages of both approaches, removing redundant data, and correctly encoding wide-baseline scenarios.

Input: Point Cloud P, Acquisition viewpoints V
Output: MVLDI M
$L \leftarrow$ set of layers for each viewpoint
$R \leftarrow$ empty cloud
$t \leftarrow$ threshold distance
for all point $p_i \in P$ **do**
  for all layer $l_j \in L$ **do**
    $d_{ij} \leftarrow$ worldToImageSpace(p_i, l_j)
    $e_j \leftarrow l_j[\, d_{ij}.u,\, d_{ij}.v\,]$
    if isInside(d_{ij}, l_j) **then**
      if isRedundant(d_{ij}, e_j, t) **then**
        break
      else
        if e_j is empty **then**
          addToLayer(d_{ij}, l_j)
          break
        else
          if $d_{ij}.d < e_{ij}.d$ **then**
            replaceInLayer(d_{ij}, e_j, l_j)
            retryPoint(e_j)
            break
  addPoint(R, e_j) //not encoded in any layer
if R is not empty **then**
  re run with R
$M \leftarrow$ all non-empty layers

Algorithm 1: MVLDI encoding algorithm

3.1 Encoding algorithm

Our layer generation process is similar to the traditional LDI algorithm described by Shade et al. [11]. Recently, different processes have been proposed for this step [1, 4], which targeted optimizations for specific use case scenarios. The proposed hole-filling and inpainting techniques were targeted to a head-face parallax user interaction paradigm, where corrections can be made at image space, and assuming all pixels must be filled at all times. If such assumptions were done about the scenario to be used with MVLDI, these could be easily incorporated. However, we established a more general scenario, only assuming depth values were available per pixel, which is common nowadays using commodity depth cameras (e.g. MS-Kinect, Asus Xtion Pro, Intel RealSense).

Initially, each of the RGBD frames is transformed into a point cloud using the u,v image coordinates of each pixel, the depth value $d(u,v)$, and the intrinsic parameters of the capture device. All resulting clouds are then combined in a single volume P, using the transformations in the extrinsic matrix. Algorithm 1 describes the encoding process for a given volume P.

We create N data structures, one for each layer l_j, where N is the number of cameras in the capture process. Each one is positioned according to the extrinsic calibration parameters for each one of the viewpoints, and is sized

according to the recording resolution of the input devices.

For each input point p_i, we try to encode it in a layer l_j by projecting it to that layer image space as a depth pixel d_{ij}. If $isInside(d_{ij}, l_j)$, meaning that (u,v) coordinates of d_{ij} are positive and smaller than the width and height of the layer, we check for redundancy $isRedundant(d_{ij}, e_j, t)$ (described in Section 3.2) where e_j is the depth pixel in layer l_j at the (u,v) coordinates of d_{ij}. In the case the data point is redundant, we start processing p_{i+1}, marking p_i accordingly, not including it in l_j. Otherwise, the point is encoded if the depth pixel e_{ij} is currently empty. If it is not, and d_{ij} has smaller depth than e_{ij}, we add d_{ij} to l_j, and e_{ij} is added for re-encoding. If e_{ij} is already filled with a point with smaller depth than d_{ij}, we go to the next layer.

Finally, in the case the point can not be placed in any of the layers, we add it to a collection R which will be processed with newly created layers. This is only the case when calibration errors exist. In all of the tested datasets these points were always encoded to a single layer, and represented less than 0.1% of the original point cloud. The final step will be adding all non empty layers to the MVLDI M.

3.2 Redundancy detection

In the classical LDI definition, a point P_{uvd} is considered to be redundant if $\|P_{uvd}.d - L_i[u,v].d\| < t$. Due to the fact that a central viewpoint is defined, and only depth comparisons are made in this particular image-space, sampling of data is not equal in all directions. When parallel to the viewing plane, pixel distance is evaluated, which in world coordinates is higher proportionally to the depth value and focal length f. When along the optical rays coming from the LDI viewpoint, a fixed threshold value t is used.

On a head-face parallax or 3DTV scenario, this was not seen as a problem, due to the fact that visualization is meant to be parallel to the central viewpoint. The sampling rate in the viewing direction is not perceived in these scenarios due to the visualization position restrictions. The lower sampling rate in the background pixels is also not noticed due to the fact that a parallel movement to the central viewpoint viewing plane does not change their image-space distance, not revealing possibly empty pixels.

Algorithm 2 shows our approach to the same problem. The first key aspect of our method is using world coordinates to estimate the distance between the pretended point and the correspondent point in the layer, opposed to just the depth value. The further away from the central viewpoint of an LDI, the greater the discrepancy between a depth-based threshold, and one based in Euclidean coordinates.

Input: P_{uvd}, P_{xyz} point to be encoded in image and world coordinates
Output: true or false
$L \leftarrow$ current Layer
$t \leftarrow$ threshold distance
$f \leftarrow$ focal lenght from intr. matrix
$s \leftarrow \frac{P_{uvd}.d}{f}$
$n \leftarrow \frac{t}{s}$
$v_0 \leftarrow d_{ij}.v - n$
for $i = P_{uvd}.u - n$ to $i = P_{uvd}.u + n$ **do**
  for $i = P_{uvd}.v - n$ to $i = P_{uvd}.v + n$ **do**
    $Q_{uvd} \leftarrow L[i,j]$
    if $\|Q_{xyz} - P_{xyz}\| < t$ **then**
      return true
return false

Algorithm 2: Redundancy detection algorithm. n surrounding pixels are checked in order to properly evaluate all points that might be under the threshold distance t.

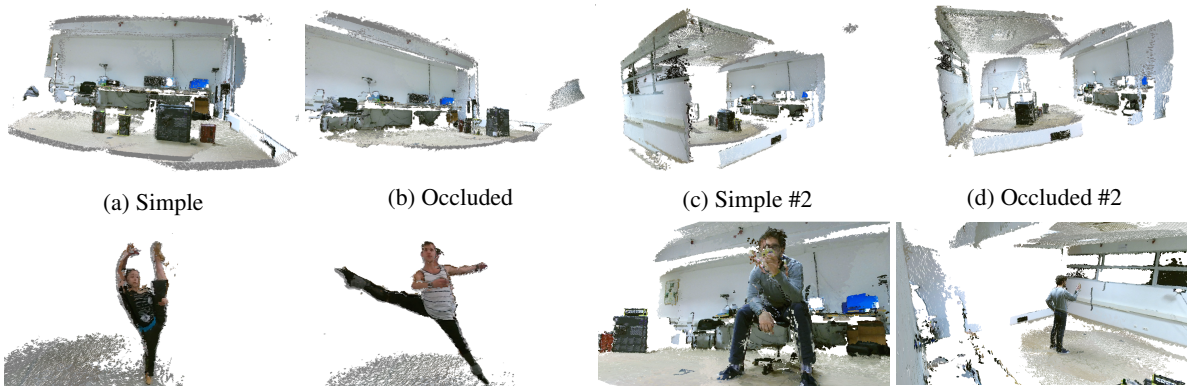
The second aspect is the fact that we also consider surrounding pixels to P_{uvd}. We first calculate the world distance s between pixels at that depth by calculating $\frac{P_{uvd}.d}{f}$ where $P_{uvd}.d$ is the depth value of the point being analysed, and f the focal distance from the intrinsics matrix. We then calculate the number of pixels to be checked around the encoded point by dividing the threshold t by s. By doing so, we very efficiently check every possible point that might be in a distance smaller than t from the considered point. Further layers do not need to be checked since the order of layer consideration is the same for each point, so in the case of a point not being considered redundant in that layer, we will naturally move to the next one to perform the same test.

Name	# Cameras	Baseline	Pt. number
Dancer M	3	Wide	68.932
Dancer F	3	Wide	75.146
Simple	4	Narrow	563.315
Simple #2	4	Wide	491.707
Simple #3	7	Wide	828.822
Occluded	4	Narrow	567.331
Occluded #2	4	Wide	505.232
Sitting	4	Narrow	580.736
Selfie	4	Wide	511.161

Table 1: Description of the tested datasets.

4 RESULTS

We tested our approach with varied datasets captured with the Microsoft Kinect sensor, each one holding different properties. Table 1 gives a brief description of each dataset, and Figure 2 shows a snapshot of each. We compared MVLDI with LDI using both the proposed global thresholding algorithm, and the traditional



(a) Simple (b) Occluded (c) Simple #2 (d) Occluded #2

(e) Dancer F (f) Dancer M (g) Sitting (h) Selfie

Figure 2: Snapshot of our used datasets. Simple #3 is the same as Simple #2 but having 3 more cameras.

image-based technique in order to individually evaluate the effect of each contribution. MVLDI with global thresholding represents our proposed technique, and LDI with image-based thresholding, the traditional LDI implementation. Our goal was to minimize the number of generated layers, while more effectively detecting redundant data. Computing time of the MVLDI is typically smaller since less layers are created. However, using a large threshold on the redundancy detection may increase computation time.

Both narrow and wide baseline were contemplated in order to validate our claim of MVLDI having a superior performance than LDI on a wide scenario. Different numbers of acquisition devices were also tested in order to validate the scalability of the process, and also validate the claim that redundancy is proportional to the number of capture devices.

Finally, while most of our datasets were controlled lab scenarios in order to control the disposition of the objects related to the cameras, we also included data captured from a realistic dance scenario (Figures 2e 2f). "Simple" datasets contained lines of boxes visible by most of the cameras (Figures 2a 2c) , "Occluded" datasets had a line of boxes occluded in one of the points of view (Figures 2b 2d). Sitting and Selfie (Figures 2g 2h) included a person in a controlled scenario in order to better evaluate if redundancy detection had a negative effect in the perception of more delicate shapes.

4.1 Quantitative Analysis

Figure 3 shows the achieved results regarding redundancy detection. The values in the table are calculated regarding the total number of points considered for encoding. Table 2 shows the percentage of the total dataset that was discarded in the case of LDI due to being outside the viewing volume of the central viewpoint.

Our approach had a superior performance over LDI in all tested datasets. Notably, the LDI approach had a high number of low populated layers (over 100 in Simple, Occluded #2 and Selfie, as seen in Table 3) This experimentally confirms the problems exemplified in Figure 1a, experimentally confirmed in Figure 5. With a wide-baseline capture setup, surfaces perpendicular to the central viewpoint viewing plane will create an elevated amount of created layers. Also, redundancy detection on the wide-baseline scenarios was lower with the classical approach. This is related to the amount of non-encoded points (over 40% of in some scenarios (Table 2, example in Figure1b) which would include walls and floor sections where typically a lot of redundancy was found, but also due to the redundancy detection algorithm.

The proposed global thresholding technique performed better in all scenarios. The difference was smaller in narrow scenarios, since the data was captured and sampled from the same perspective, so a pixel-based comparison still had an impact, albeit smaller. The biggest difference was found in wide baseline scenarios, where several points under the used threshold were located in neighboring pixels but not considered redundant only considering the depth value.

When comparing solely the difference in the number of points of view, the multiview approach had a smaller number of layers in all scenarios, being close or equal to the baseline for comparison (MVD, where the number of layers is equal to the number of input devices). The multiview approach did not suffer from the problem presented in Table 2, where big segments of the point cloud were discarded due to not being visualized by the central viewpoint. This is essential in a free visualization application through a virtual camera, specially in wide-baseline scenarios. Redundancy detection in MVLDI was higher in all scenarios except for the Dancer scenarios where only three cameras were used, and no data was left out of the encoding process.

An argument could be made about the choice of the central viewpoint for the LDI, and considering a virtually generated point of view that would include all of the

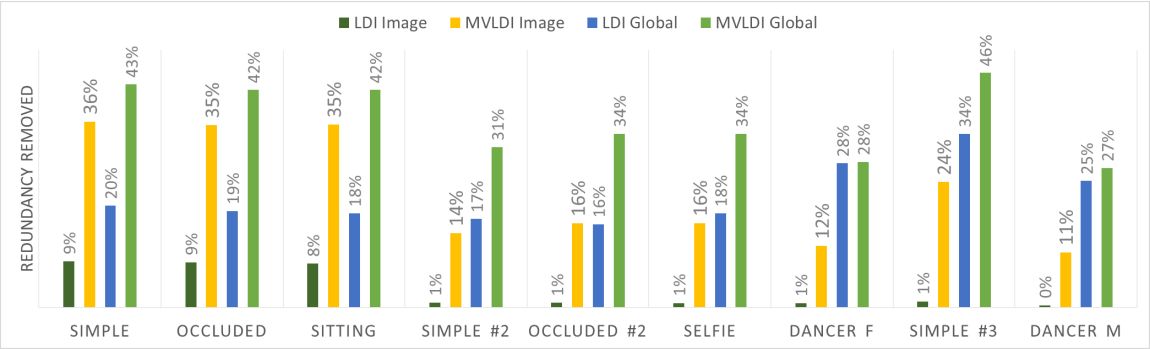


Figure 3: Percentage of data detected as redundant from the original cloud.

Dancer M	Dancer F	Simple	Simple #2	Simple #3	Occluded	Occluded #2	Sitting	Selfie
0%	0%	19%	48 %	39%	19%	46%	20 %	45%

Table 2: Percentage of points discarded by the LDI approaches due to being outside of the frustum of the central viewpoint.

data. The main problem with this proposition is the fact that a single viewpoint that includes all the data would necessarily be placed further away than the existing acquisition viewpoints, which certainly increases the problem in Figure 1a. The further the viewpoint is from the data, the more likely the planar surfaces in the cloud are aligned with the optical rays, increasing the number of low-populated layers.

Several previous works on LDI have reported higher rates of redundancy detection than the ones presented in this article [13]. The datasets typically used for benchmark are the breakdancers and ballet sequences from microsoft research, which are aimed to a head-face parallax interaction, or 3DTV. Cameras are separated by approximately 20cm from each other, which is a very narrow baseline. Also, depth precision is considerably lower, with only 256 values to represent the whole range of the scene. The depth cameras used in this work have a precision in the order of milimiters, which is why less values with the same depth are encountered using the image-based algorithm.

4.2 Qualitative Analysis

Although we report high redundancy removal in all of the presented scenarios, the quality of the visualization was not compromised. Figure 4 shows a side by side comparison of the input cloud, the redundant data, and the encoded result. On the dancer example where 27% redundancy was reported, data from the dancers back, silhouette, and floor were correctly reported as redundant (Figure 4b), not compromising the final visualization, as seen in Figure 4c.

On "Occluded" where a narrow baseline setup was used, we can clearly see a large amount of data (42%) consisted of walls, floor, and only the front part of the non-occluded box as being reported as redundant (Figure 4e), with the remainder of the data, included the boxes behind the front one, being perfectly visible in the encoded version (Figure 4f).

Also, considerations about the viability of MVLDI on a video scenario can be made. Our generated layers are less sparse as seen on figure 5, and in a smaller number, as seen in Table 3. We can more easily guarantee coherent matches between frames using block matching algorithms due to having more densely populated regions on the image. Also, our final number of layers is typically equal to the number of acquisition viewpoints, unlike LDI's which are inherently dependent on the content of the scene, and might create uneven distribution of layers per frame, which are less reliable for temporal matching.

Name	LDI Image	MVLDI Image	LDI Global	MVLDI Global
Dancer M	23	4	7	3
Dancer F	20	4	6	3
Simple	10	5	8	4
Simple #2	115	5	11	4
Simple #3	88	11	16	8
Occluded	13	5	12	5
Occluded #2	119	5	10	4
Sitting	17	5	12	5
Selfie	125	5	9	4

Table 3: Number of layers generated by each approach. MVLDI with global thresholding has the overall lower number of layers.

5 CONCLUSION

We presented MVLDI, an alternative data representation for a single frame of multiview video that allows wide-baseline VBR applications to take advantage of the redundancy detection of the LDIs. Moreover, MVLDI is also a more efficient alternative to represent a point cloud for an image-based rendering scenario.

(a) Original *Dancer M* (b) Redundant *Dancer M* (c) Encoded *Dancer M*

(d) Original *Occluded* (e) Redundant *Occluded* (f) Encoded *Occluded*

Figure 4: Result of the encoding process. Original (a, d), redundancy removed (b, e), and encoded data (c, f) for a wide and a narrow baseline scenario (top vs bottom row).

An alternative redundancy detection technique was introduced, which considers points in a global space opposed to the image-space thresholding of LDI, ensuring a homogeneous sampling of the data.

Our results showed that the proposed approach creates a smaller number of layers and detects redundancy at higher rates in both narrow and wide baseline scenarios, while also showing higher efficiency in scenarios with more cameras. The generated Layers are more dense than the typical residual LDI layers. They can be effectively used for temporal compression on video, and also geometry estimation, as proposed by [7].

Although results with a higher number of cameras were positive, the more cameras we have, the order of evaluation of viewpoints becomes more important. Our future work will be focused on evaluating the optimal order of encoding for the data, or the generation of alternatives viewpoints that provide an efficient encoding, and evaluating the application of temporal compression and depth encoding to our work. This technique will also be essential on encoding a point cloud that has no capture viewpoint information, such as a single sensor performing a sweep scan of large structures.

6 ACKNOWLEDGMENTS

This work was supported by the European Research Council under the project (Ref. 336200). This work was partially supported by national funds through FCT - Fundação para a Ciência e Tecnologia, under project PEst-OE/EEI/LA0021/2013 and by national funds through FCT with reference UID/CEC/50021/2013.

7 REFERENCES

[1] I. Daribo and H. Saito. A novel inpainting-based layered depth video for 3dtv. *Broadcasting, IEEE Transactions on*, 57(2):533–541, 2011.

[2] L. Do, G. Bravo, S. Zinger, and P.H.N. de With. Gpu-accelerated real-time free-viewpoint dibr for 3dtv. *Consumer Electronics, IEEE Transactions on*, 58(2):633–640, May 2012.

[3] W. S. Kim, A. Ortega, P. Lai, and D. Tian. Depth map coding optimization using rendered view distortion for 3d video coding. *IEEE Transactions on Image Processing*, 24(11):3534–3545, Nov 2015.

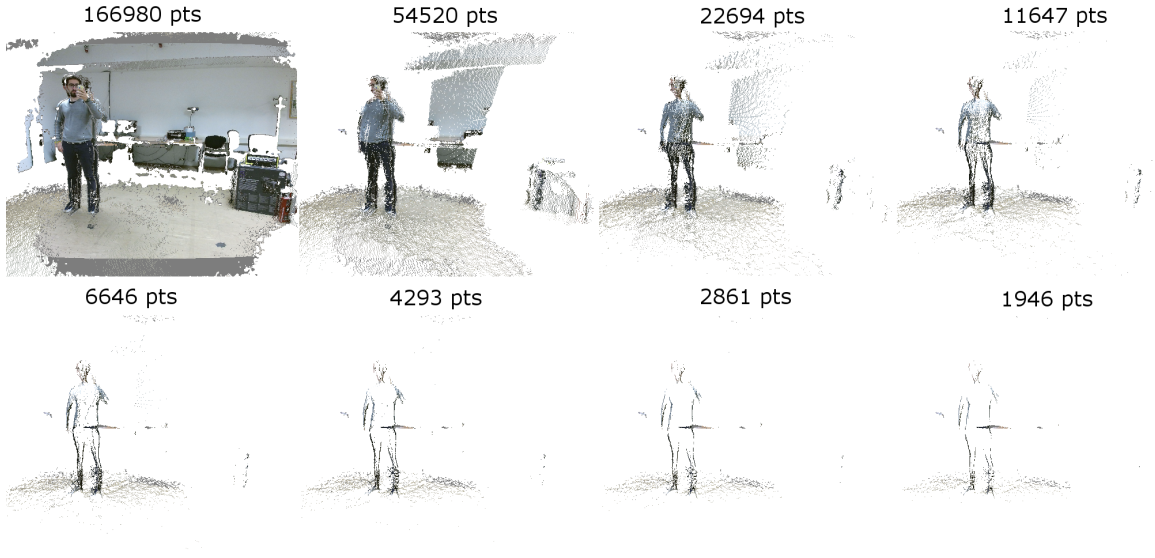
[4] S. Kirshanthan, L. Lajanugen, P.N.D. Panagoda, L.P. Wijesinghe, D.V.S.X. De Silva, and A.A. Pasqual. Layered depth image based hevc multiview codec. In G. et al Bebis, editor, *Advances in Visual Computing*, volume 8888 of *Lecture Notes in Computer Science*, pages 376–385. Springer International Publishing, 2014.

[5] J. Lei, S. Li, C. Zhu, M. T. Sun, and C. Hou. Depth coding based on depth-texture motion and structure similarities. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(2):275–286, Feb 2015.

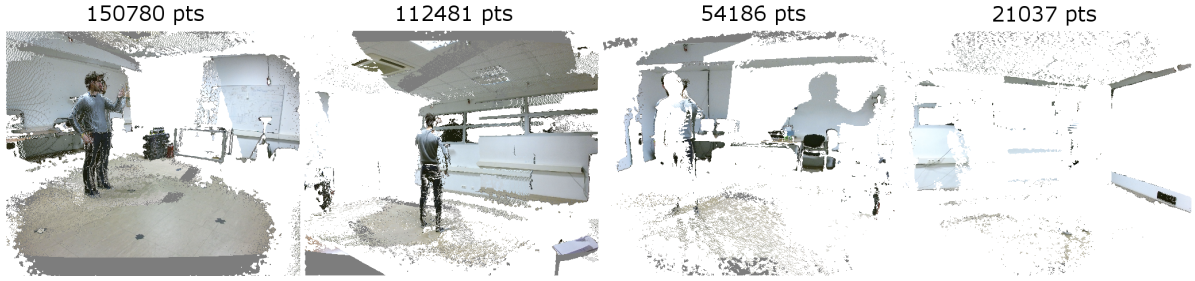
[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand. The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communication*, 24(1-2):73 – 88, 2009. Special issue on advances in three-dimensional television and video.

[7] P. Merkle, K. Müller, D. Marpe, and T. Wiegand. Depth intra coding for 3d video based on geometric primitives. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):570–582, March 2016.

[8] P. Merkle, A. Smolic, K. Müller, and T. Wiegand. Multi-view video plus depth representation and coding. In *Image Processing, 2007. ICIP 2007.*



(a) LDI



(b) MVLDI

Figure 5: Comparison between the first 8 LDI layers, and the full 4 MVLDI layers of the "Selfie" dataset, with number of points per layer. The silhouette of the subject, floor and a table in the background generate low populated layers being parallel to the optical rays.

IEEE International Conference on, volume 1, pages I – 201–I – 204, Sept 2007.

[9] K. Müller, A. Smolic, K. Dix, P. Kauff, and T. Wiegand. Reliability-based generation and view synthesis in layered depth video. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 34–39, 2008.

[10] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A/ Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 741–754, New York, NY, USA, 2016. ACM.

[11] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM.

[12] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems. In *2008 15th IEEE International Conference on Image Processing*, pages 2448–2451, 2008.

[13] Seung-Uk Yoon, Eun-Kyung Lee, Sung-Yeol Kim, and Yo-Sung Ho. A framework for multi-view video coding using layered depth images. In *Advances in Multimedia Information Processing-PCM 2005*, pages 431–442. Springer, 2005.

[14] Seung-Uk Yoon, Eun-Kyung Lee, Sung-Yeol Kim, and Yo-Sung Ho. A framework for representation and processing of multi-view video using the concept of layered depth image. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 46(2-3):87–102, 2007.