Elemental Images Labelling and Grouping to Minimise Disparity Error in Texture-less Regions of Holoscopic Images

Bodor Almatrouk, Hongying Meng, and Rafiq Swash
Dept. of Electronic and Electrical Engineering, Brunel University London, Uxbridge, United Kingdom
{bodor.almatrouk, hongying.meng, rafiq.swash}@brunel.ac.uk

***Abstract*—Holoscopic imaging is a promising technique that captures full-colour spatial images using a single aperture. It uses a micro-lens array to view the scene at different angles and record 4D information on a two-dimensional surface, making it useful for depth estimation. However, current disparity estimation methods suffer from poor performance in texture-less regions. This paper proposes a novel method to reduce the disparity error in these regions by directly labelling and grouping elemental images from a Holoscopic image. The proposed approach involves extracting a subset of viewpoint images from the Holoscopic image and subjecting them to conventional image segmentation. Labels are then applied to the elemental images corresponding to each segmented object using viewpoint images/elemental image pixels mapping. Content-based image retrieval is also employed to improve segmentation. The proposed technique has wide applications for 3D imaging, including augmented and virtual reality, inspection, robotics, security, and entertainment.**

***Index Terms*—Holoscopic, elemental image, viewpoint image, segmentation, texture-less disparity**

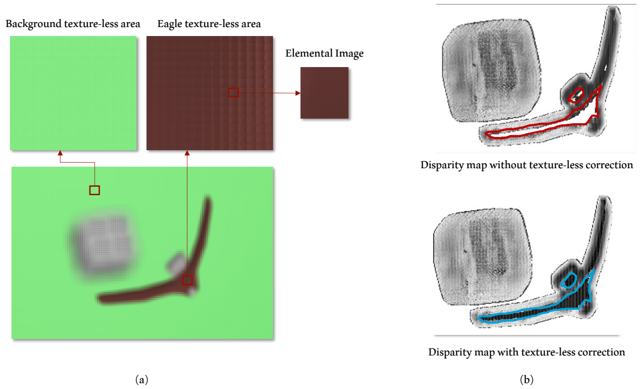

Background texture-less area
Eagle texture-less area
Elemental Image
Disparity map without texture-less correction
Disparity map with texture-less correction
(a)
(b)

Fig. 1: (a) Holoscopic with two texture-less regions. (b) The top EIs-based disparity map before correction. Due to texture-less EIs, the region outlined in red has disparity = 0. The bottom disparity image shows the corrected disparity.

I. Introduction

Holoscopic cameras capture light in a scene in its entirety, both in terms of spatial and angular dimensions. The main lens in traditional cameras translates the object plane into the camera's own image plane. Holoscopic cameras are built on the same fundamental principles as conventional cameras, with the addition of an array of micro-lenses (MLA) in front of the image sensor. The micro-lenses in the MLA focus the light beams from various directions onto a sub-image of the sensor, capturing the entire scene in three dimensions. As a result, the pixels forming behind the micro-lens record the same data as traditional cameras but do so more precisely by separately measuring the information coming in from various angles. The images behind the micro-lens are known as elemental images (EIs). Each EI pixel indicates a unique angle at which light hits its micro-lens. Therefore the location and orientation of each light beam can be determined, pixel by pixel, by analysing the EIs. A sub-aperture image of a scene can be created from a specific viewpoint by extracting pixels from the same locations of the EIs. This is known as a viewpoint Image (VPI).

The raw Holoscopic image is the original data that is captured by the image sensor and is represented as a grid of EIs. Each EI convergence point has multiple light rays, each representing a unique perspective. All EIs receive light from one VPI at almost the same pixel. As shown in Fig. 3 (d), pixels from EIs with the same location can be combined to extract a VPI. VPIs are usually used for disparity estimation because they cover the entire scene from one perspective, while EIs only cover a portion of the scene. VPIs also resemble 2D orthographic stereo images, thus stereo image disparity estimation methods can be applied to them with minor adjustments. To extract VPIs, one must map sensor data to reconstruct the scene, which may be challenging. The lens error and camera calibration should be performed first. Fig. 2 (a) and (b) show extraction without and with lens correction. The scene's geometry (depth of each object) should be considered when creating VPI, otherwise areas that are not ”in focus” will result in image artefacts like those in Fig. 2 (c). To increase resolution and reduce artefacts, the viewpoint extraction method requires selecting a set of pixels from each EI in the same location. Combining multiple VPIs by selecting a set of pixels rather than a single pixel is time-consuming and reduces angular information [1]. EIs estimate disparity without pre-processing beyond lens correction for the reasons above. However, EI-based disparity computation fails in texture-free areas as seen in Fig. 1. A flood-fill technique could fill texture-less areas with the correct disparity values from the textured EIs around them, but not knowing whether they belong to the same object could lead to more disparity errors.

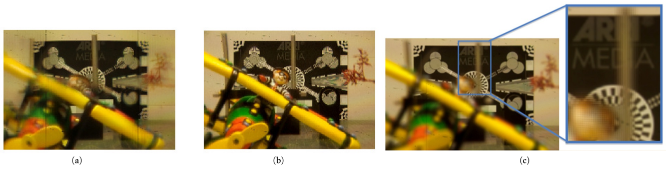


Fig. 2: Extracted viewpoint images

II. Related Work

Stereo VPI, focal stack, and epipolar plane images are the most common Holoscopic depth estimation methods. Stereo VPIs is the most popular Holoscopic depth estimation method. The central VPI generates depth from disparity data from several outer VPIs [2]. As demonstrated in [2], depth maps, especially in occluded areas, benefit from using multiple VPIs as reference images rather than the central view. [3] presented multiple VPIs with occlusion-aware depth estimation algorithm. Holoscopic cameras sample many VPIs, creating narrow baseline VPIs unsuitable for correspondence-block matching. Phase-shift-based sub-pixel displacement improved correspondence-block matching in [4]. Using MLA in front of the camera sensor, each EI pixel represents a different angle of light from different focal planes, allowing the VPIs to be refocused after capturing the image. One of a Holoscopic camera's most intriguing features. These focus cues can be used alone or with correspondence and other cues [5, 6]. Holoscopic cameras produce calibrated and rectified VIs that can be stacked to create an epipolar plane image (EPI). EPI slices can be compared to see if the correspondence positions have moved due to VPI angle shifts [7, 8].

In this paper, we propose a new method to minimise the error in texture-less EIs by first identifying the collection of EIs that encompass the same object and then using that information to fill the disparity in the texture-less regions by the surrounding textured EIs.

III. Methodology

This section presents a methodology to minimise disparity error in texture-less areas of raw Holoscopic images using EI labelling and grouping. The solution involves two steps: object identification and labelling, and disparity correction. The object segmentation is performed on central and corner VPIs, and then the pixels of the segmented objects are mapped back to their associated EIs to label them. Content-based image retrieval is used to enhance the classification. For disparity correction, texture-less EIs are identified and grouped based on neighbourhood information. The segments are filled with EIs around them only if they belong to the same object. The proposed methodology is illustrated in Fig. 3.

A. Viewpoint Images Extraction and Segmentation

VPIs are used first for object segmentation because they can encompass all of the objects in the scene, The objects in VPIs are segmented using contours tracing algorithm [9]. The canny filter returns image edges without object data. Thus, the image contouring algorithm searches for image holes to fill. Each hole represents a VPI object. Aggregating edge pixels into smooth curves provides this. Fig. 3 (c) shows how each VPI object is masked. VPI segmentation classifies EIs. Each Holoscopic VPI pixel represents an EI. We can label an EI with A by mapping VPI mask pixels from A to its EI. Multiple VPI per Holoscopic image means an EI could theoretically belong to multiple classes (one EI can have multiple objects). Central and corner VPIs are extracted. Structural similarity ('SSIM') [10] orders objects from multiple VPIS for coherence. Due to object positioning, some viewpoints may have fewer segmented objects. The segmentation algorithm can merge nearby objects. For consistency, SSIM ignores such images.

B. Enhancing Elemental Images Labelling using Content-Based Image Retrieval

Improved Content-Based Image Retrieval (CBIR) [11] retrieves images from a dataset with features such as textures, colours, shapes, and more using a query image or user-specified features. The CBIR system uses query images for searching. A correctly labelled EI from each class (eagle, dice, and background) can be used as a query image . Multiple query images per object should be chosen to accurately represent the object in various EIs.

The query EIs are selected unsupervisedlyFirst, the mean pixel value μ_{obj} of the object image from the viewpoint segmentation is computed, as well as the mean pixel value μ_{EI} of each EI labelled to contain that object in the initial EI classification.

The query image with the lowest mean difference will correspond to the segmented object. Thus, elemental images are sorted in ascending order by the difference between μ_{obj} and μ_{EI}. Finally, the object's query images are Top-k images from the sorted results.

CBIR employs the generated query EIs per object to extract the appropriate EIs for each query. The procedure is described in terms of images queried. Firstly, The EI are selected on every query image. The holoscopic image's EIs are selected using the query's average and standard deviation. Texture and edges are computed from selected EIs. Before using descriptors from each group, elemental image selection is done. This process removes EIs with colours that differ from the query image to reduce the number of candidates.

Image colour mean μ can be sensitive to factors such as brightness and artefacts. Therefore, comparing the means of the query image and EI is not optimal. As a result, the colour standard deviation σ is employed to define a range $range_{low} = \mu_q - \sigma_q, \quad range_{high} = \mu_q + \sigma_q$. Lastly, EIs whose mean pixel value lies within the interval $[range_{low}, range_{high}]$ are selected.

Local binary pattern features are extracted for each query image. To LBP_q, Canny edges replace the V channel. The image is also converted from RGB to HSV and edge detected on the V channel. To obtain HSb_q (edges with coloured image). Each selected EI (EI_s) has texture and edge descriptors (LBP_q, HSb_s). After that, the texture and edge differences between the EI and the query image are calculated ($diff_{lbp}$ and $diff_{edge}$, respectively) and added to get the final feature distance, $diff_{feat}$:

$$diff_{feat} = (LBPq - LBPs) + (HSb_q - HSb_s) \quad (1)$$

Given that the EIs that are the closest matches to the query will have the lowest diff feat value, the list of differences is arranged in ascending order. To conclude, the list is sorted, and the top-k values are used for the corresponding EIS.

Algorithm 1 CBIR to retrieve corresponding elemental images per query image

```
1. Extract local binary patterns ($LBP_q$) features for the query image $q$.
2. Convert $q$ to HSV colour space, and replace the $V$ channel with Canny edges to obtain $HSb_q$.
3. Compute mean ($\mu_q$) and standard deviation ($std_q$) of $HSb_q$.
4. Define: $range_{low} = \mu_q - std_q$ , $range_{high} = \mu_q + std_q$
5. Select elemental images whose mean pixel value falls in $[range_{low}, range_{high}]$.
for selected $EL_s \in EI$ do
    a. Extract local binary patterns $LBP_s$
    b. Convert $EI_s$ to HSV colour space, and replace V channel with Canny edges to obtain $HSb_s$
    c. Compute $diff_{lbp}$: difference between $LBP_q$ and $LBP_s$ features
    d. Compute $diff_{edge}$: difference between $HSb_q$ and $HSb_s$
    e. Sum $diff_{lbp}$ and $diff_{edge}$ to get final feature distance, $diff_{feat}$
    f. Append $diff_{feat}$ to a list
end for
7. Sort the $diff_{feat}$ list in ascending order, the idea being that the most matching EIs to the query will have the smallest $diff_{feat}$ value and so on.
8. Choose top-k values from the sorted list as the matched EIs
```

IV. Disparity Correction

Due to its ability to calculate depth using a single-aperture camera, depth estimation from holoscopic images has gained interest in recent years. We computed disparity from raw Holoscopic image from consecutive EIs pair using semi-global block matching as in [12]. As seen in Fig. 1, raw holoscopic images (EI-based) divide texture-less areas into texture-less EIs. Incorrect disparity (disparity = 0) will result from this. It is ineffective to fix all texture-less regions in a Holoscopic image with the disparity value of the surrounding area without identifying if they belong to the same object. Thus, image labelling and EIs classification are needed to locate texture-less areas and determine which object they belong to in order to find the right disparity value to fill the space.

Holoscopic images are 4D, with (s, t) being the viewpoint images plane and (u, v) being the elemental images plane [13]. First, we use the raw Holoscopic image to determine the binary (0 or 1) mask for texture-less EIs M_e and textured EIs M_t (which EI has texture and which does not). The mask's raw Holoscopic image dimension (u, v) is initialised to zero. M_e is calculated by measuring pixel value variations of each EI:

$$\Delta p = p_{max} - p_{min} \quad (2)$$

where Δp is the difference between maximum pixel (p) value and minimum pixel value. A predetermined threshold t is compared to the difference value:

$$\mathrm{EI}_{\left(u_{(i,j)}, v_{(i,j)}\right)} = \begin{cases} 1; \Delta p \leqslant t \\ 0;\ \text{else} \end{cases} \quad (3)$$

The condition is met if the difference is less than or equal to the threshold. True is assigned to the EI at ($u_{(i,j)}$, $v_{(i,j)}$) position in M_e, where (i, j) is the EIs coordinates in the uv space. M_t is computed by taking a negation of the M_e.

Inaccurate, texture-less EIs are usually found in groups or segments. To group neighbouring coordinates together, a list of the texture-less EI coordinates is stored, and neighbouring coordinates are selected to divide the list into multiple segments. The problem of grouping neighbouring coordinates in a two-dimensional plane can be solved by repeatedly checking if each pair of coordinates is adjacent. Given a list of coordinates $C = c_1, c_2, \ldots, c_n$, we define the set of neighbouring coordinates $N(c_i)$ of a coordinate c_i in a two-dimensional plane as:

$$N\left(c_i\right) = \left\{c_j \mid j \neq i \text{ and } \mathrm{d}\left(c_i, c_j\right) \leq \sqrt{2}\right\} \quad (4)$$

where $d(c_i, c_j)$ is the Euclidean distance between the coordinates c_i and c_j. If $d(c_i, c_j) \leq \sqrt{2}$, then c_i and c_j are considered neighbours. We define a function $get_{n}eighboring_{l}ists(C)$ that groups the input coordinates C based on their neighbours.

To create a mask for background regions in foreground EIs (bgr_{fg}), the mean and standard deviation of each RGB channel is calculated. If its value is less than one standard deviation from the mean across all three RGB channels, an EI's foreground pixel is considered background. This assumes that most foreground image pixels are background pixels, and the average pixel value should be within 1 standard deviation of the background pixel intensity. Calculate the disparity mode for bgr_{fg}. Finally, all background EI disparity values (in D) are replaced by the mode (mean or median) of bgr_fg disparity values.

A list stores the texture-less EI coordinates. By selecting neighbouring coordinates, this list is divided into multiple texture-less segments. Given a list of coordinates in a two-dimensional plane, the objective is to group neighbouring coordinates together. Two adjacent coordinates are considered neighbours. The problem can be solved by repeatedly checking if each pair of coordinates is adjacent. For each pair of coordinates that are not neighbours, a new group is formed, and the remaining neighbours are added. The input list is C

and the output is G. Assume c_i is C's $i-th$ coordinate. For simplicity, we'll call c_i's neighbours $N(c_i)$. We can define the set of neighbouring coordinates $N(c_i)$ of a coordinate c_i in a two-dimensional plane given a list of coordinates $C = c_1, c_2, \ldots, c_n$:

$$N\left(c_i\right)=\left\{c_j \mid j \neq i \text{ and } \mathrm{d}\left(c_i, c_j\right) \leq \sqrt{2}\right\} \tag{5}$$

where $d(c_i, c_j)$ is the Euclidean distance between the coordinates c_i and c_j. If $d(c_i, c_j) \leq \sqrt{2}$, then c_i and c_j are considered neighbours. We can then define a function $get_neighboring_lists(C)$ that groups the input coordinates C based on their neighbours. This function is defined as follows:

Algorithm 2 Neighbour list extraction

```
function GET_NEIGHBOURED_LISTS($C$)
    $G \leftarrow []$
    for $c_i$ in $C$ do
        $found \leftarrow$ False
        for each $group$ in $G$ do
            if exists $c_j \in group$ such that $c_i \in N(c_j)$ then
                $group \leftarrow group \cup c_i$
                $found \leftarrow$ True
                break
            end if
        end for
        if not $found$ then
            $G \leftarrow G \cup c_i$
        end if
    end for
    return $G$
end function
```

The output G is a list of lists with each inner list representing a group of neighbouring coordinates. The coordinates in each group are sorted by their addition to G.

Each segment is labelled to an object by checking the labels of the EIs in that segment (use the most presented label). Disparity values are assigned to each segment. Each segment's disparity value is chosen from adjacent EIs. Background EI disparity values are corrected using the correct background disparity values in the foreground EIs using colour descriptors [14, 15]. To create a mask for background regions in foreground EIs (bgr_{fg}), the mean and standard deviation of each RGB channel is calculated. If its value is less than one standard deviation from the mean across all three RGB channels, an EI's foreground pixel is considered background. This assumes that most foreground image pixels are background pixels. This implies that the average pixel value should be within 1 standard deviation of the background pixel intensity. Calculate the disparity mode for bgr_{fg}. Finally, all background EI disparity values (in D) are replaced by the mode (mean or median) of bgr_fg disparity values.

V. Evaluation

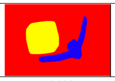
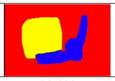
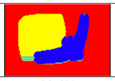
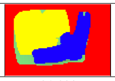
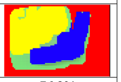
A. Labelling and Grouping Evaluation

Viewpoint Images Segmentation Configuration The proposed algorithm's performance was assessed using synthetic Holoscopic images generated a the Blender [16] add-on simulator. The accuracy of the segmentation is measured by dividing the number of estimated EIs in a class EI_{est} (for example number of EIs labelled as "dice") by the number of accurate EIs EI_{acc} of that class (counted manually):

$$Accuracy = \frac{EI_{est}}{EI_{acc}} * 100 \tag{6}$$

A single EI pixel can contain multiple objects and belong to a different VPI. Thus, corner and central VPIs are extracted for precision. The more VPIs used, the more precise the segmentation around the object's boundaries, as shown in Table. I. The VPIs should be evenly distributed around the object's centre for full coverage. Naturally, the algorithm's execution speed decreases as the number of VPIs increases, as shown in Table. III. However, the results did not change between 15 and 30 VPIs because holoscopic images have redundant data, so not all VPIs are needed.

TABLE I: Image segmentation using various numbers of VPIs to test the effect of the number of VPIs on the accuracy

Viewpoint image resolution	397x265	198x132	132x88	99x66	79x53
Elemental image resolution	20x20	40x40	60x60	80x80	100x100
Result prior to CBIR correction					
Accuracy	100%	89.8%	89.4%	81.3%	76.2%
Time (seconds)	94.354	25.0888	13.0785	9.60737	7.03534

Holoscopic Image Configuration Although the segmentation method relies on EIs, the first step is VPI-based object segmentation. The outcome is more precise with higher-resolution VPIs. As shown in Table. III, higher-resolution VPIs such as 397×265 produce a result with perfect precision. However, a holoscopic image with an 8K sensor size and a VPI of 397×265 would generate a 20x20 EIs, which is very low in resolution. Even though a higher VPI produces more defined and distinct EIs, a higher resolution EI is still needed for more accurate and reliable CBIR similarity matching and disparity estimation (broader baseline, higher spatial resolution). Because of their higher resolution, larger EIs improve CBIR correction. In conclusion, the first phase yields a more precise result with high-resolution and roughly 9 VPIs, but at the cost of a longer execution time and low-resolution EIs. EIs with lower resolution VPIs are more effective for correction.

B. Disparity Evaluation

The precision of a disparity map is evaluated by comparing it to the ground truth disparity map, which was generated in Blender, using the Mean Relative Error (MRE) matrix.

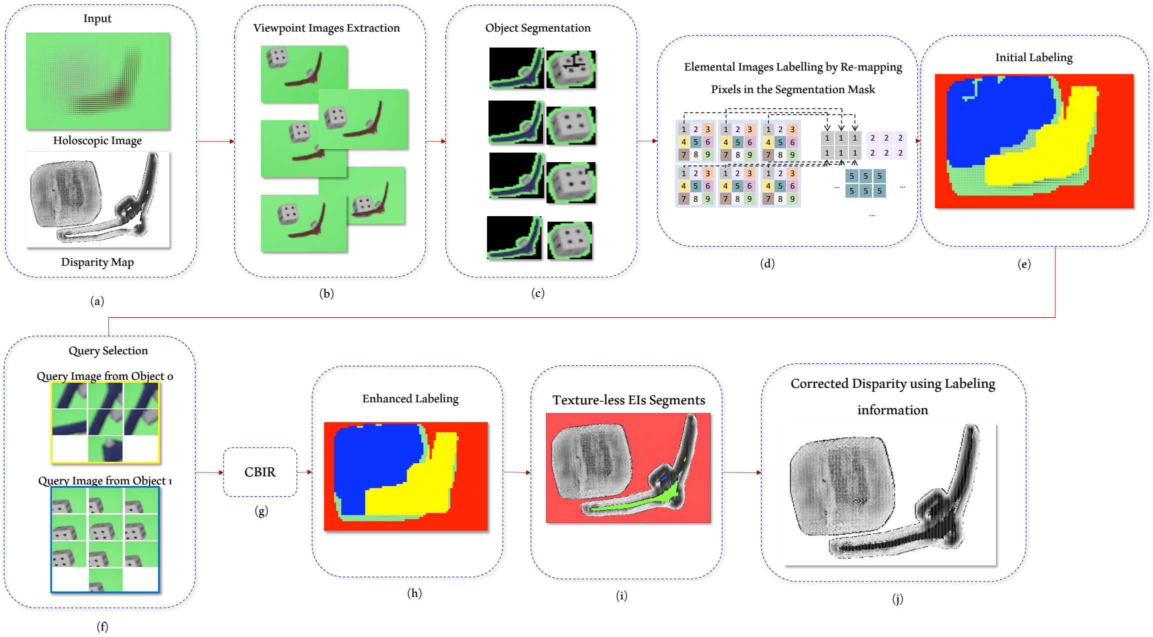

Input
Holoscopic Image
Disparity Map
(a)
Viewpoint Images Extraction
(b)
Object Segmentation
(c)
Elemental Images Labelling by Re-mapping Pixels in the Segmentation Mask
(d)
Initial Labeling
(e)
Query Selection
Query Image from Object 0
Query Image from Object 1
(f)
CBIR
(g)
Enhanced Labeling
(h)
Texture-less EIs Segments
(i)
Corrected Disparity using Labeling information
(j)

Fig. 3: Methods pipeline. (a) Input Holoscopic and initial disparity images. (b) Extract corner and central VPIs. (c) Mask VPI-segmented objects. (e) Initial labelling of EIs (yellow: elemental images with the eagle, blue: dice, red: background). (f) Labelling creates query images. (g) CBIR labelling for enhancement. (i) Selecting and segmenting texture-less EIs. (j) Corrected disparity in texture-less regions.

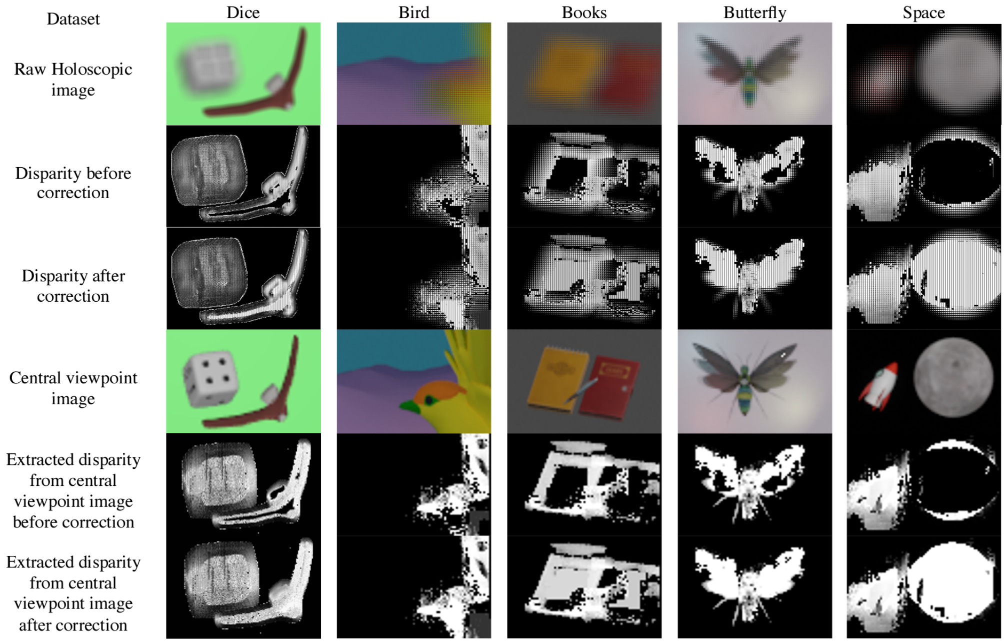
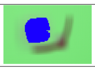
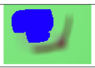
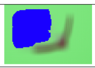
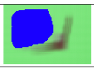
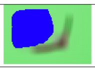
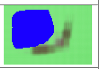

Dataset
Dice
Bird
Books
Butterfly
Space
Raw Holoscopic image
Disparity before correction
Disparity after correction
Central viewpoint image
Extracted disparity from central viewpoint image before correction
Extracted disparity from central viewpoint image after correction

Fig. 4: The table shows five Holoscopic images. The first row shows raw Holoscopic images. After computing the disparity map from the elemental images, the area without texture can be seen (darker pixels are farther from the scene and vice versa). Textured elemental images correct the disparity map. Clutter and incorrect disparity values can make the correction inaccurate. The viewpoint images and the central view-raw disparity man disparity were extracted.

TABLE II: MRE performance of the disparity estimation before and after the correction on 8 generated holoscopic images.

Method	Generated Images								Average
	Dice	Brid	Books	Butterfly	Space	Cup	Vehicle	Greek	
Before correction	12.47	20.84	18.68	14.73	16.95	14.38	16.84	13.39	16.035
After correction	8.94	20.13	16.28	12.85	13.48	13.14	15.29	11.27	13.021

TABLE III: Image segmentation using various numbers of VPIs to test the effect of the number of VPIs on the accuracy

Number of VPI used	Central VPI	3 VPIs	6 VPIs	9 VPIs	15 VPIs	30 VPIs
Result						
Accuracy	41.2%	62.6%	79.4%	85.3%	91.1%	91.1%

$$MRE = \frac{1}{N}\sum_{i=1}^{n}\left|\frac{D_{est} - D_{gt}}{D_{gt}}\right| \quad (7)$$

where N is the number of image pixels, D_{est} is the disparity estimate for each pixel, and D_{gt} is the corresponding ground truth disparity value.

Instead of comparing the absolute differences between the estimated and ground-truth values, MRE looks at the ratio of the two. It can be computed by averaging the absolute disparities between the estimated and ground truth values and then normalising them by the estimated disparity values. Table II displays individual and aggregate results of averaging 8 Holoscopic disparity maps.

VI. Conclusion

We introduce an innovative approach to Holoscopic segmentation using EIs. This technique group EIs under specific scene objects, with the understanding that some EIs may fall under more than one object. First, we use VPI segmentation to assign labels to the EIs that correspond to each object visible in the corresponding VPI pixel. To improve the labelling of the EIs further, CBIR is used, in which the query images are chosen randomly from among the labelled EIs. The disparity between texture-less EIs is fixed using the segmentation information obtained. The algorithm was tested at various EI resolutions and has shown promising results.

References

[1] O. Abdul Fatah, "Post-production of holoscopic 3d image," Ph.D. dissertation, Brunel University London, 2015.

[2] J. Peng, Z. Xiong, Y. Zhang, D. Liu, and F. Wu, "Lf-fusion: Dense and accurate 3d reconstruction from light field images," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[3] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3487–3495.

[4] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1547–1555.

[5] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from shading, defocus, and correspondence using light-field angular coherence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1940–1948.

[6] I. K. Park, K. M. Lee *et al.*, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2484–2497, 2017.

[7] H. Lv, K. Gu, Y. Zhang, and Q. Dai, "Light field depth estimation exploiting linear structure in epi," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[8] H. Sheng, P. Zhao, S. Zhang, J. Zhang, and D. Yang, "Occlusion-aware depth estimation for light field using multi-orientation epis," *Pattern Recognition*, vol. 74, pp. 587–599, 2018.

[9] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[11] L. Pavithra and T. S. Sharmila, "An efficient framework for image retrieval using color, texture and edge features," *Computers & Electrical Engineering*, vol. 70, pp. 580–593, 2018.

[12] B. Almatrouk, H. Meng, and M. R. Swash, "Disparity estimation from holoscopic elemental images," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*. Springer, 2021, pp. 1106–1113.

[13] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 31–42.

[14] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 20–54, 2015.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[16] B. Foundation, "Blender," https://www.blender.org/.