

High Quality Light Field Extraction and Post-Processing for Raw Plenoptic Data

Pierre Matysiak, Mairéad Grogan, Mikaël Le Pendu, Martin Alain, Emin Zerman, Aljosa Smolic

Abstract—Light field technology has reached a certain level of maturity in recent years, and its applications in both computer vision research and industry are offering new perspectives for cinematography and virtual reality. Several methods of capture exist, each with its own advantages and drawbacks. One of these methods involves the use of handheld plenoptic cameras. While these cameras offer freedom and ease of use, they also suffer from various visual artefacts and inconsistencies. We propose in this paper an advanced pipeline that enhances their output. After extracting sub-aperture images from the RAW images with our demultiplexing method, we perform three correction steps. We first remove hot pixel artefacts, then correct colour inconsistencies between views using a colour transfer method, and finally we apply a state of the art light field denoising technique to ensure a high image quality. An in-depth analysis is provided for every step of the pipeline, as well as their interaction within the system. We compare our approach to existing state of the art sub-aperture image extracting algorithms, using a number of metrics as well as a subjective experiment. Finally, we showcase the positive impact of our system on a number of relevant light field applications.

I. INTRODUCTION

Theorised in the mid-19th century, light fields are a description of light rays passing through a volume, and more generally an interpretation of the properties and interaction of light in three-dimensional space, as thoroughly described by Levoy et al. [1]. This is commonly represented by using a simplification of the plenoptic function proposed by Adelson et al. [2] into a four-dimensional function describing spatial and angular dimensions. A number of different methods have been developed to capture light fields. First, a camera array can be constructed, by placing identical cameras in a regular planar grid to simultaneously capture images of a scene, as introduced by Wilburn et al. [3]. Alternatively, a single camera can be fixed on a gantry and moved at regular intervals to simulate the effect obtained from a camera array [4], although only applicable to static scenes. Finally, through the use of plenoptic cameras that contain an array of micro-lenses between the main lens and the sensor, proposed by Ng et al. [5], which produces dense light fields.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776. This work was supported by TCHPC (Research IT, Trinity College Dublin). Some calculations were performed on the Lonsdale cluster maintained by the Trinity Centre for High Performance Computing. This cluster was funded through grants from SFI. This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes additional results and comparisons, as well as a more in-depth study of colour and noise patterns in LF data. Contact matysiak@tcd.ie for further questions about this work.

Plenoptic cameras have gained interest in recent years, after the release of two models by the Lytro company which aimed at allowing professionals and amateurs alike to capture light fields with a dense angular sampling, where each micro-lens outputs unfocused micro-images. Unfocused cameras are generally exploited by extracting sub-aperture images (SAI), each with a very wide depth of field and representing different viewpoints of the scene. Shortly after, another similar type of camera called plenoptic 2.0 was developed by Lumsdaine et al. [6]. As opposed to the previous design, here each micro-lens outputs a focused micro-image. Focused cameras are typically used to render focused images where the focus can be dynamically adjusted based on user input. Due to their unique design, plenoptic cameras generate much more complex RAW data compared to traditional cameras, and the exploitation of this data is made more difficult as a result.

Classically, computer vision applications using light field data prefer to use output in the form of SAIs, as they are more practical to handle. In this paper, we focus our attention on these and explain our method to extract SAIs from unfocused plenoptic camera RAW data. Despite the different solutions proposed by Cho et al. [7], Xu et al. [8] or Seifi et al. [9], the light field toolbox presented by Dansereau et al. [10] is the most widely used in the research community as it offers the most complete pipeline to extract SAIs. It has for instance played a central role in the standardisation effort for light field compression as it is now used as part of the JPEG PLENO [11] test set. The extraction method comprises four steps which can be summarised as follows: a devignetting step first compensates for the vignetting effect of the micro-lenses, i.e. darker pixels on the edges of each micro-lens; demosaicing is then applied to retrieve the RGB colour components of each pixel from the partial colour information actually captured by camera sensors; a compensation of possible rotation of the micro-lens array is performed; finally the pixels are rearranged to convert the image into a set of sub-aperture images.

However, the extracted views suffer from several types of artefacts such as noise, unnatural horizontal stripes, ghosting effects, colour and brightness inconsistencies on external SAIs, inaccurate colour balance, and substantial loss of dynamic range. Unfortunately, these defects have a negative impact on many light field applications including depth estimation, segmentation, rendering and compression. An overview of these issues is described by Wu et al. [13]. Because of these distortions, a good portion of the external views are generally ignored for these applications. This impacts their results, as using less SAIs means potentially missing out on the critical

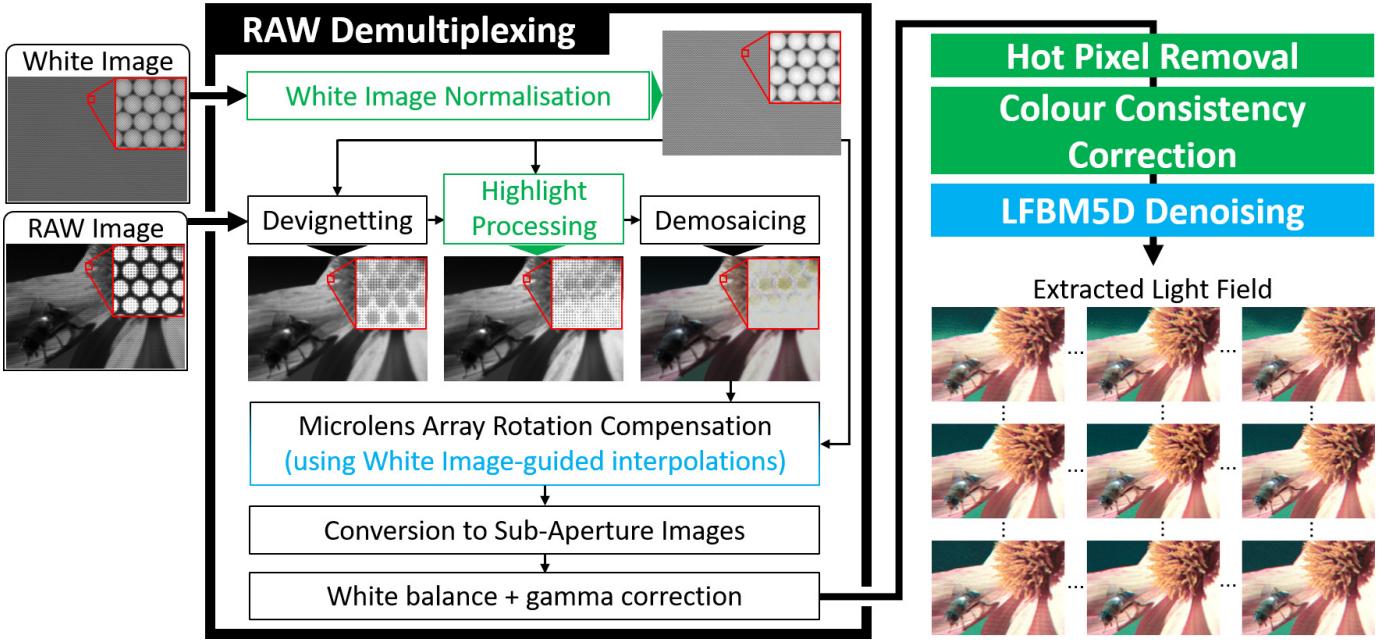


Fig. 1: Overview of the proposed Light Field Pipeline. The steps in green correspond to the contributions described in this paper (these include the contributions of our preliminary paper [12]). The steps in blue are state of the art methods that we additionally included in our pipeline, but which are not present in the traditional pipeline [10].

information they could provide. Although the proprietary Lytro Desktop software compensates for many of these issues, it is still unsuitable for the generation of SAI arrays as its main goal is to render refocused images.

In this paper, we propose an improved processing pipeline for lenslet-based plenoptic cameras which is an extension of our previous work [12]. First, in order to take the most advantage of the captured RAW data, we propose several improvements on the low level steps of the traditional demultiplexing method of Dansereau et al., converting the RAW data into sub-aperture images. We specifically show that the previously used devignetting step had a negative impact on the image quality, as it tampered with the colour balance and brightness and caused loss of dynamic range, and we propose ways to correct this. Additionally, we propose a highlight processing method to compensate for colour issues related to sensor saturation. In order to reduce the ghosting effect of external SAIs, we recommend the use of White Image-guided interpolation following the work of David et al. [14]. Once the sub-aperture images are extracted, further quality enhancement steps are then proposed as post-processing tools. These include hot pixel removal, correction of colour inconsistencies between SAIs and denoising. Finally, we show the benefits of the different steps of the proposed pipeline by conducting a subjective experiment and analysing the impact of our results on several applications, in comparison with the state of the art demultiplexing proposed by Dansereau et al. The applications studied include light field rendering, compression, super-resolution and editing. For improved readability, and because our pipeline covers many different methods, the related previous work done for each of them will be covered at the beginning of this paper's major sections.

II. OVERVIEW OF THE PROPOSED PIPELINE

The essential steps of our pipeline are depicted in Fig. 1. The input data consists of the RAW image formed on the plenoptic camera sensor. Due to the Bayer filter array placed on the sensor, each pixel contains colour information only for one of the RGB components. Another RAW image, called White Image (WI) is obtained by a preliminary calibration process involving the capture of a uniform white surface.

First, a RAW demultiplexing method building upon that of Dansereau et al. [10] is proposed. After a normalisation step, the White Image is used to remove the vignetting in the input RAW image. A novel highlight processing step is then proposed to retrieve natural colours in bright areas where some pixels reach the sensor's saturation level. A standard demosaicing method (Malvar et al. [15]) then recovers the full RGB colour components at each pixel. Similarly to the work of Dansereau et al., we compensate for slight misalignments between the microlens array and the sensor. The recent White Image-guided interpolation method of David et al. [14] is used for that purpose. The last steps in the method of Dansereau et al. are applied without modification. Pixels are reorganised to convert the lenslet image into a set of sub-aperture images. Due to the hexagonal lenslet pattern, this step includes a resampling of each image from a hexagonal to a square grid of pixels. Finally, white balance and gamma correction are performed. The novel aspects of the RAW demultiplexing and the challenges addressed are presented in Sec. III.

After the RAW demultiplexing, several defects remain to be corrected. The failure of isolated pixels is a common problem in digital imaging. We choose to correct these so-called 'hot pixels' in a post-processing stage detailed in Sec. IV. A colour correction method is then proposed in Sec. V to ensure colour consistency between the light field views. Finally, plenoptic

imaging is prone to noise that we remove using the LFBM5D method of Alain et al. [16] (see Sec. VI).

Compared to our preliminary paper [12], the proposed pipeline additionally contains a highlight processing (Sec. III-B) and a hot pixel removal (Sec. IV) steps. Furthermore, we present a complete evaluation of the pipeline with a subjective study (Sec. VII) and a study of the effect of our quality enhancement tools on various applications (Sec. VIII).

III. RAW LIGHT FIELD DEMULTIPLEXING

In plenoptic cameras, the micro-lens array forms a specific pattern on the sensor, which introduces new difficulties when processing the RAW data. While the light field toolbox presented by Dansereau et al. [10] is capable of converting the RAW data into a set of SAIs, the final images suffer from various artefacts. Further research on the subject has essentially focused on adapting the demosaicing step. For instance, a specific demosaicing method was designed by Yu et al. [17] for focused plenoptic cameras, i.e. plenoptic 2.0. For the more common case of unfocused plenoptic cameras, different optimisation methods have been employed in the demosaicing of Xu et al. [8], Huang et al. [18] and Lian et al. [19]. These methods perform respectively 4D kernel regression, dictionary learning with sparse optimisation, and total variation minimisation. An original approach is proposed by Seifi et al. [9], where the demosaicing is performed after the demultiplexing so that a disparity map can be estimated first, and then used to guide the demosaicing step. Finally, White Image-guided demosaicing and interpolation tools are proposed by David et al. [14] to avoid mixing colour information from different micro-lenses.

However, we believe that a more global analysis of the demultiplexing is necessary, since many inaccuracies can occur in other steps of the pipeline, or during the capture process itself.

A. White Image Normalisation

Dansereau et al. [10] perform lenslet devignetting first as it results in more uniform brightness over the sensor array and thus, easier demosaicing. This step simply consists of a pixel-wise division of the RAW image by a RAW White Image (WI) that exhibits the pattern of micro-lens vignetting. Note that the WI was previously captured during a calibration step by the same device as the picture being processed. However, the red, green and blue filters in the Bayer filter array have different responses to the white light. For this

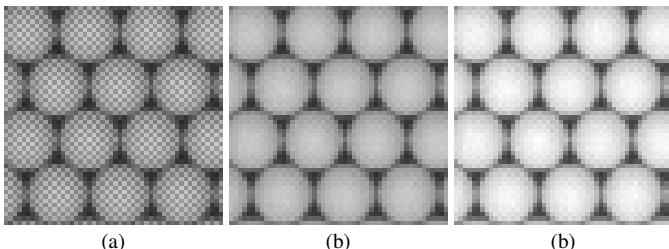


Fig. 2: Detail of a White Image: (a) unprocessed, (b) after colour normalisation, (c) after both colour and global normalisation.

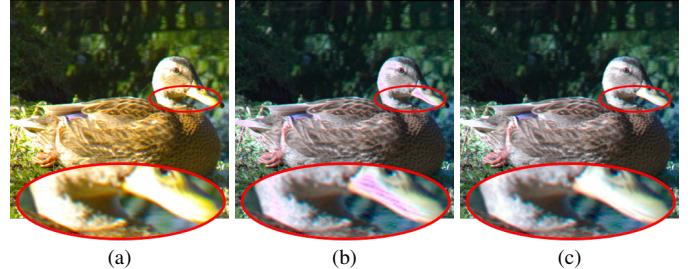


Fig. 3: One view of the light field *duck*: (a) without WI normalisation [10], (b) with WI normalisation, (c) with WI normalisation and highlight processing.

reason, the Bayer pattern is visible on the WI as shown in Fig. 2(a). Therefore, performing the devignetting step using the unprocessed WI, as in the method of Dansereau et al., interferes with the white balance of the final result. We correct this issue by multiplying the red and blue pixels of the WI by normalisation factors provided as metadata of the camera and accounting for the different responses of the RGB filters. Note that these factors may also be obtained by colour calibration of the sensor.

Furthermore, since the pixel values of the WI are lower than 1.0 even at micro-lens centres, the devignetting of Dansereau et al. also increases the overall brightness of the light field. Bright areas reaching higher values than 1.0 after devignetting are considered saturated in the rest of the process, and the information is lost. Therefore, we also apply a global normalisation of the WI by dividing all the pixels by its 99.9th percentile (we do not use the maximum value to exclude possible hot pixels). The effect of the White Image normalisation step on the colours and brightness of the final result is clearly visible in Fig. 3. However, by decreasing the overall brightness, this normalisation step also reveals unnatural colours in the highlights (see the pink colour in Fig. 3(b)). We correct this issue in a highlight processing step presented in the next subsection.

B. Highlight Processing

Due to the different saturation levels of the red, green and blue pixels on the sensor, the highlights have unnatural colours. This is a common problem in digital imaging, observed in over-exposed regions after applying the white balance. However, in conventional cameras those regions are typically uniform, making it possible to correct the highlights after the demosaicing (e.g. [20]–[23]). In plenoptic cameras, the micro-lens vignetting as well as possible inaccuracies in the devignetting (e.g. slight mismatch between white image and RAW image) and demosaicing steps may create artefacts in those regions for some of the extracted SAIs (see Fig. 3(b)). The simplest approach for solving the issue is to clip the highlights after the white balance as done in [10]. However, this results in a loss of details in the highlights (see Fig. 3(a)). Therefore, we propose a highlight processing step taking into account the vignetting pattern (i.e. the normalised White Image) and applied before the demosaicing in order to retain the details in the highlights without introducing colour artefacts.

For this step, blocks of four pixels on the RAW image are processed independently. Since the highlight processing

is performed before demosaicing, each of the four pixels is associated with only one RGB component organised according to the Bayer pattern. We note the values of these pixels x_r , x_{g_1} , x_{g_2} , x_b . Corresponding values in the normalised WI are noted w_r , w_{g_1} , w_{g_2} , w_b . In this step, we also take into account the white balance parameters s_r , s_g , and s_b by which the red, green and blue components will be respectively multiplied later in the white balance step (see Fig. 1). These values can also be interpreted as the saturation levels of each component.

First, we consider that saturated pixels are such that $x_c \cdot w_c > T$, with $c \in \{r, g_1, g_2, b\}$ and T is a threshold set to 0.99. Note that $x_c \cdot w_c$ is the original pixel value on the sensor before the degivetting.

Two cases are considered. In the case where the four pixels are saturated, no colour information is present. However, the white balance, applied to those pixels later in the process, results in an unnatural colour. Hence, we cancel the effect of the white balance by setting each pixel of index c to the value $x_c \cdot \hat{s}/s_c$, where $\hat{s} = \max(s_r, s_g, s_b)$. When at least one of the four pixels is not saturated, we find the index m of the pixel with lowest value. A saturated pixel of index c then takes the value $x_m \cdot s_m/s_c$. However, in practice, separating these two cases may cause abrupt changes of brightness. Therefore, we blend between these two behaviours using the following formula for modifying a saturated pixel x_c into x'_c :

$$x'_c = \max \left((1 - \alpha) \frac{x_m \cdot s_m}{s_c} + \alpha \frac{x_c \cdot \hat{s}}{s_c}, x_c \right), \quad (1)$$

where $\alpha \in [0, 1]$ is the blending parameter indicating the total amount of saturation as $\alpha = \min(1, x_m \cdot \frac{1}{4} \sum_c w_c)^2$. The maximum between the modified and the original value is used since $x_m \cdot s_m/s_c$ may be lower than the original saturated pixel x_c . This operation prevents possible discontinuities with neighbour pixels slightly below the saturation detection threshold.

Note that after the white balance step, the regions recovered by the highlight processing may have values above 1. In order to retain those details in the final image without affecting the overall brightness, we apply a soft saturation function $softSat$ to each pixel after the white balance step:

$$softSat(x) = 1 - \frac{\ln(1 + e^{R(1-x)})}{\ln(1 + e^R)}, \quad (2)$$

where R is a parameter controlling the smoothness of the curve (lower R resulting in smoother saturation). We set $R = 7$ in our implementation. The soft saturation curve is illustrated in Fig. 4 and the final result is shown in Fig. 3(c).

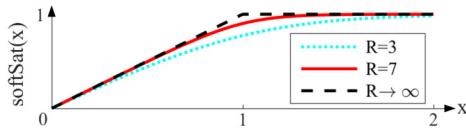


Fig. 4: Soft saturation function with different parameters R .

C. White Image-guided Interpolations

Previous analysis by David et al. [14] has shown how standard demosaicing and interpolations introduced both ghosting

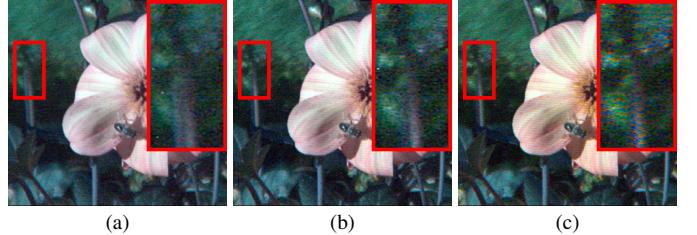


Fig. 5: Advantages and limitations of the White Image-guided method of [14]: (a) standard demosaicing [15] and bicubic interpolations, (b) standard demosaicing [15] and WI-guided interpolations, (c) WI-guided demosaicing and interpolations.

artefacts and fading of the colours in the external SAIs. In order to reduce the problem, they adapted those steps by weighting the contribution of each pixel using the vignetting pattern of the White Image. Two observations can be made from their results. Firstly, the ghosting effect is essentially reduced by the adaptation of the interpolation step (see Fig. 5(b)). Secondly, while their modified demosaicing improves the overall colour consistency between SAIs, it may also create colour noise (see Fig. 5(c)). Hence, we suggest that only the WI-guided interpolations should be used, and we propose in Sec. V a post-processing step to enforce colour homogeneity in the light field.

IV. HOT PIXEL REMOVAL

Hot pixels are isolated pixels taking extreme values due to internal errors on the camera sensor. Their detection within the RAW demultiplexing stage is challenging due to the fact that the demosaicing step retrieves inaccurate colours, not only for the hot pixels, but also for their neighbours, corresponding to angular neighbours in the light field. However, in the sub-aperture images obtained by the demultiplexing, the spatial neighbours of the hot pixels are unaffected. Furthermore, hot pixels are not accurately removed by traditional light field denoising methods, such as the ones presented in Section VI. Thus we directly perform hot pixel removal after RAW demultiplexing.

A typical issue for hot pixels is the fact that they exhibit extreme values in their colour components, but this in itself is not a sufficient criteria for detection. Instead, for each SAI I , we identify hot pixels by comparing the colour values x^i of each pixel i to those of its neighbours in $\Omega_{n \times n}(i)$, the $n \times n$ window centred on the pixel i . Based on this, we compute a probabilistic measure ρ^i to indicate how likely i is to be a hot pixel and threshold this value to detect the most likely hot pixels in the SAI. We tested colour values in both the RGB and CIELAB colour spaces [24], and found that CIELAB helped us identify hot pixels more easily, so we chose to use this colour space exclusively.¹

The procedure we use to detect hot pixels is described in Algorithm 1. For each pixel colour x^i in CIELAB space, if it lies within a colour distance t_d to only a small number of pixels (less than t_c) in the window $\Omega_{n \times n}(i)$, the value ρ^i will be high (see Algorithm 1). The distance we use in CIELAB

¹We use Matlab's `rgb2lab` function to convert to CIELAB d65, with L* taking values between 0 and 100 and a* and b* values between ± 110 .

space is the Euclidean distance. In Fig. 6(b) we display ρ_i values for each pixel in SAI (a), with the red values in (b) showing pixels with the highest ρ value. We then threshold these values in order to detect the most likely hot pixels, with $\rho_i > t_\rho$ selected (Fig. 6(c)). Since hot pixels do not typically appear as white in an image, we also add a check to make sure pixels that lie within a distance t_w of the colour white (such as small regions of white highlights on an object) are not incorrectly detected as hot pixels (see Algo. 1). Here, we can see that our detection method is robust to colour changes along edges, with very few edges being detected incorrectly as hot pixels. Finally, we correct the hot pixel i using a 3×3 median filter centred on it, which takes the median value for the L*, a* and b* components (ignoring the hot pixel values) and applies it to the hot pixel. Fig. 6(d) shows the final results in which the isolated red and green hot pixels have been successfully detected and restored via our hot pixel removal.

Result: SAI I with hot pixels removed.

Define thresholds $t_d = 30$, $t_w = 30$ and $t_\rho = 0.8$, window size $n = 7$;

for $i \in I$ **do**

```

    Compute  $\Omega_{n \times n}(i)$ ;
    /* Compute hot pixels probability map      */
    Define count = 0;
    for  $i' \in \Omega_{n \times n}(i)$  do
        if  $\|x^{i'} - x^i\|_2 < t_d$  then
            | count  $\leftarrow$  count + 1;
        end
    end
     $\rho^i = 1 - \frac{count}{n^2}$ ;
    /* Filter hot pixels                      */
    if ( $\rho^i > t_\rho$  and  $\|white - x^i\|_2 > t_w$ ) then
        |  $x^i \leftarrow \text{median}_{L^* a^* b^*}(\Omega_{3 \times 3}(i) - \{i\})$ ;
    end
end

```

Algorithm 1: The process used to detect and correct hot pixels in an SAI I . Here, $white = [100, 0, 0]$ is the colour white in CIELAB space and $\|\cdot\|_2$ denotes the Euclidean distance.

V. COLOUR CONSISTENCY CORRECTION

After RAW demultiplexing, large differences in colour still exist between the centre and external SAIs, as can be seen in Fig. 7(b). We refer the reader to the supplementary materials providing insights on how the colour consistency is affected by the demosaicing and its interaction with the devignetting step. To correct this, we chose a recent image recolouring approach proposed by Grogan et al. [25] (described in Sec. V-A) and adapt it to light fields. Similar to other colour correction approaches proposed in multiview geometry and panorama stitching applications, such as the ones by Oliveira et al. [26], Park et al. [27], Xia et al. [28] and Hwang et al. [29], this approach uses colour correspondences between a target and palette image to compute a transfer function that maps the colours from the target image to match those of the palette.

In [25], Grogan et al. show that their approach outperforms several leading colour correction approaches [27], [28], [30]–[33] when applied to images with similar content. Overall, their correspondence based method is shown to outperform those that do not consider correspondences [30]–[32] while

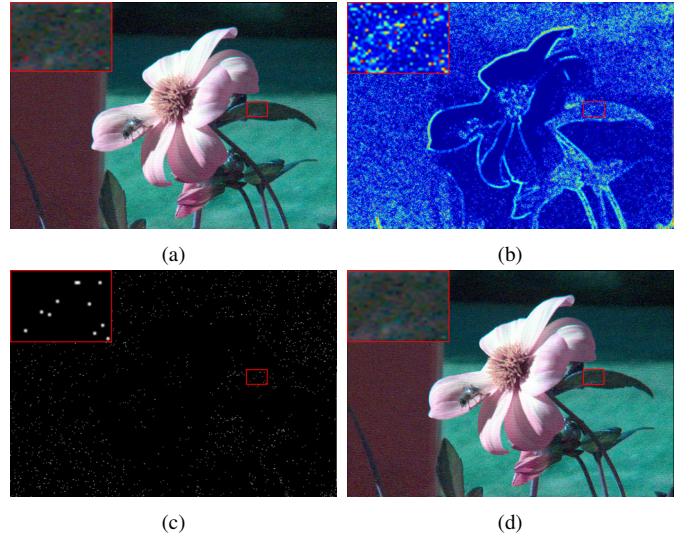


Fig. 6: (a) Input SAI with zoom clearly showing red and green hot pixels as described in Sec. IV; (b) heat map showing values ρ^i for all pixels i ; (c) detected hot pixels with $\rho^i > t_\rho$; (d) our corrected SAI.

their flexible thin plate spline colour transfer function allows them to correct more non-linear colour differences between images, outperforming methods whose transfer functions depend on only a small number of parameters [27], [28]. They also found that Hwang et al.'s method [33] can introduce visual artefacts when correspondence outliers are used to estimate the transfer function, while Grogan et al.'s cost function is shown to be more robust to outlier pairs, with the smooth transfer function also ensuring that similar colours stay similar after recolouring. For these reasons, we decided to adapt Grogan et al.'s method to light field data, and in this section give further details about our approach.

A. Correspondence Estimation

For the colour transfer algorithm to produce good results, we needed to compute accurate correspondences between both views. We explored existing methods for correspondence estimation between SAIs following the example of Chen et al. [34], who used optical flow successfully in their work on light fields and chose to adapt a similar method. As the colour transfer algorithm does not require that all the pixels of an image pair are matched to obtain satisfying results, a preference towards lower computational complexity was taken in this step. We therefore used only the first step of coarse-to-fine patch matching (CPM) developed by Hu et al. [35] to obtain a set of sparse correspondences between pairs of views.

It is similar to PatchMatch [36] and works by taking n pixels on a regular grid in the target SAI as seed pixels, noted $c_t^{(n)}$, and finds their matching pixels, or correspondences, in the palette view, noted $c_p^{(n)}$. To compute these correspondences, a candidate set of correspondences is first found using SIFT features. In the second step, points are sampled around each candidate correspondence, and if they prove to be more accurate, replace the original. This process iterates a number of times until a globally stable set of correspondences is

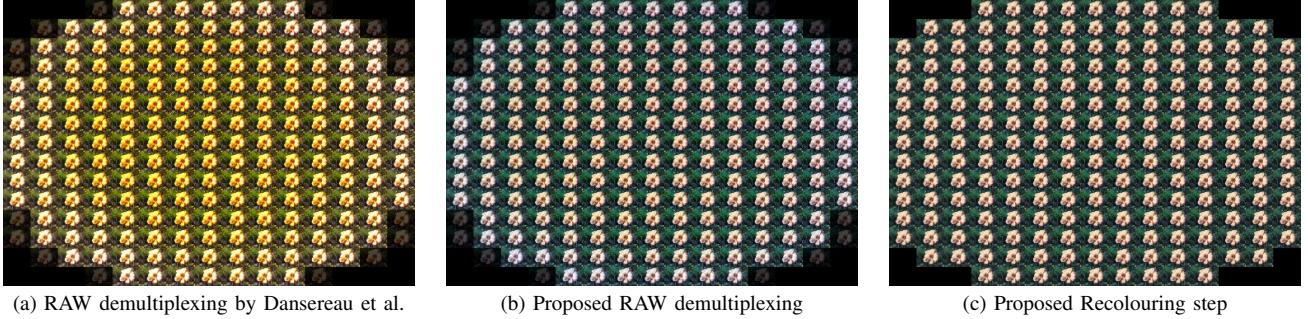


Fig. 7: Matrix of sub-aperture images of the *bee_2* light field. This view allows to better perceive the improvement on the fidelity of the colours our demultiplexing (b) offers over the demultiplexing of Dansereau et al. [10] (a), and highlights the colour inconsistencies on the external views that we fix (c).

found. Finally, outliers are detected and removed from the pool, creating the final set of correspondences $\{c_t^{(n)}, c_p^{(n)}\}$.

This process provides us with on average 30k correspondences between two images of size 625×434 . We extract the colour information of these pixel pairs to form the colour models representing both target and palette images and pass them on to the colour transfer algorithm.

B. Colour Transfer

Given a set of n colour correspondences $(c_t^{(k)}, c_p^{(k)})_{k=1\dots n}$ between the target and palette image, where the set of colours $c_t^{(k)}$ from the target image should correspond to the colours $c_p^{(k)}$ from the palette after recolouring, Grogan et al. [25] propose to fit a Gaussian Mixture Model to each set of correspondences as follows:

$$p_t(x|\theta) = \sum_{k=1}^n \frac{1}{n} \mathcal{N}(x; \phi(c_t^{(k)}, \theta), h^2 I) \quad (3)$$

and

$$p_p(x) = \sum_{k=1}^n \frac{1}{n} \mathcal{N}(x; c_p^{(k)}, h^2 I) \quad (4)$$

Each Gaussian is associated with an identical isotropic covariance matrix $h^2 I$, and the vector $x \in \mathbb{R}^3$ represents values from a 3D colour space. Transforming the colours $c_t^{(k)}$ by some transformation ϕ which depends on θ creates the colours $\phi(c_t^{(k)}, \theta)$. The goal is to transform the colour distribution of the target image to match that of the palette image by estimating the transformation ϕ that registers $p_t(x|\theta)$ to $p_p(x)$. Grogan et al. propose letting ϕ be a global parametric thin plate spline transformation:

$$\phi(x, \theta) = \underbrace{Ax + o}_{\text{Affine}} + \underbrace{\sum_{j=1}^m -w_j \|x - q_j\|_2}_{\text{nonlinear}} \quad (5)$$

with $\theta = \{A, o, w_j\}$ the parameters to be estimated. Here, A is an affine matrix, o is a translation offset vector and $\{w_j \in \mathbb{R}^3\}$ are coefficients controlling the non-linear part of the transformation with $\{q_j\}_{j=1,\dots,m}$ a set of control points evenly sampled in the colour space.

To estimate the parameter θ controlling ϕ , the following is minimised:

$$\mathcal{C}(\theta) = -\langle p_t | p_p \rangle = \sum_{k=1}^n \frac{1}{n^2} \mathcal{N}(0; \phi(c_t^{(k)}, \theta) - c_p^{(k)}, 2h^2 I) \quad (6)$$

For our application, better results were obtained using the CIELAB colour space rather than the RGB colour space. Similar to [25], we add a regularisation term to ensure our thin plate spline function is smooth. We also found that additional steps had to be taken when optimising this cost function to avoid local minima. Therefore we used a two step process to estimate θ . The first step computes an initial estimate for θ using a subsample of the correspondences (computed using k-means with $K = 1000$). In the second step, the parameters A and o are fixed and only the non-linear parameters w_j are refined using the full set of correspondences. We found that this two step process ensured local minima were avoided and the correct solution was found.

C. Propagation

As an improvement on our previous work [12] we decide here to focus on the propagation scheme that allows for the best visual quality. Our goal here is to guarantee two things: firstly that colours be consistent across the light field, i.e. two consecutive views should not exhibit any visible difference between them, and secondly that true scene colours be preserved as much as possible in all the views.

The propagation scheme we use in this work is twofold. The demultiplexing step of our pipeline ensures we obtain natural colours in all the views, with the central views displaying the most accurate colours. Therefore, when recolouring a target SAI T in the light field we first compute correspondences between T and the centre view M of the light field using the method described in Section V-A. To ensure T displays similar colours to its neighbouring images, we also compute correspondences between T and its inner neighbouring view P . If T lies on the central column of the light field, its inner neighbouring view P also lies on the central column, either above or below T depending on which is closest to the centre view M . Otherwise, P will lie on the same row of the light field as T , again either to the left or right of T depending on which is closest to the centre view M . For each target SAI T , this combination of correspondences is then input into Eqs. (3) and (4), meaning each view will be recoloured using

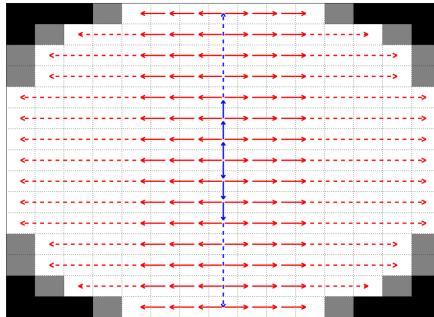


Fig. 8: Pattern representing our propagation of colours in a light field. The centre column is processed first, then each row.

a function computed using correspondences from the centre view and its inner neighbour. We recolour each SAI in the light field starting with the centre column, from the centre view and outward, then in every row, from the middle view outward. This procedure is described in Algorithm 2, with a visual explanation given in Fig. 8.

The choice to include the previously-recoloured neighbour views was made empirically. When only using the centre view, artefacts occur due to large parallax between external views and the centre view. On the other hand using only the inner neighbour views can cause a slight fading of colours as we move toward the edges of the light field, as each successive recolouring causes a minor loss of colour intensity. Therefore, the use of two views simultaneously as palette images helps us ensure that we get both the most vivid colour in every view, and a reduction in the possible artefacts introduced by the method.

Result: Colour corrected Light Field with $m \times m$ SAIs.

Define $M = I_{(\lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil)}$:

```

for  $j = 0 : (\lfloor \frac{m}{2} \rfloor - 1)$  do
    /* Centre column, downward direction */
    colCorrect ( $\lceil \frac{m}{2} \rceil + j + 1, \lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil + j, \lceil \frac{m}{2} \rceil$ );
    /* Centre column, upward direction */
    colCorrect ( $\lceil \frac{m}{2} \rceil - j - 1, \lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil - j, \lceil \frac{m}{2} \rceil$ );
end

for  $k = 0 : \lfloor \frac{m}{2} \rfloor$  do
    for  $j = 0 : (\lfloor \frac{m}{2} \rfloor - 1)$  do
        /* every row, from centre SAI to right */
        colCorrect ( $\lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil + j + 1, \lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil + j$ );
        /* every row, from centre SAI to left */
        colCorrect ( $\lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil - j - 1, \lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil - j$ );
    end
end
```

Function $\text{colCorrect}(row_T, col_T, row_P, col_P)$:

```

 $T = I_{(row_T, col_T)}$ ;
 $P = I_{(row_P, col_P)}$ ;
 $(c_t, c_p) = (c_T, c_P) \cup (c_T, c_M)$ ;
 $\hat{\theta} = \text{argmin}_{\theta} C(\theta)$ ;
 $I_{(row_T, col_T)} \leftarrow \phi(T, \hat{\theta})$ ;
return;
```

Algorithm 2: The propagation technique used to recolour the entire light field. The blue and red regions correspond to the blue and red arrows in Fig. 8.

VI. DENOISING

In addition to the colour artefacts addressed previously, lenslet plenoptic cameras have by design a lower signal to noise ratio than single lens cameras, since light rays coming from different angular directions are no longer averaged on a single pixel sensor. Thus we propose to apply denoising as a final step of the pipeline. Note that in conventional photography, it is sometimes preferred to perform denoising either before or jointly with the demosaicing step when the RAW data is available (e.g. [37]–[39]). However, applying such denoising methods on plenoptic RAW data would not exploit the redundancies in the 4 dimensions of the light field. Furthermore, neighbour pixels on the sensor may correspond to different lenslets and thus belong to diametrically opposed SAIs. Therefore, applying denoising in the early stages of the pipeline is likely to produce cross-talk artefacts on the external SAIs. A similar issue was observed in Fig. 5(a) when using linear interpolation for the lenslet array rotation step. Denoising is then preferably applied at the end of the process, after the colour correction step, since the latter helps to improve the consistency of the light field over the angular dimensions. This benefits most existing light field denoising methods, which rely on the angular redundancy.

A trivial approach to light field denoising consists of applying an existing single image denoising filter (see Dabov et al. [40], Shao et al. [41] or Jain et al. [42]) independently to the SAIs. However, better performances are obtained when taking into account the pixel correlation in-between the SAIs. SAIs can, for instance, be stacked in a pseudo-video sequence and denoised using a state of the art video denoiser such as the VBM4D of Maggioni et al. [43]. The angular correlation can also be exploited along the epipolar plane images (EPI): Li et al. [44] use a two-step method which first denoises EPIs taken along a given spatial and angular dimension (e.g. horizontally), and then processes this first estimate using the complementary EPIs (e.g. vertically). Sepas-Moghaddam et al. [45] stack the EPIs in a pseudo-video sequence and denoise using the video denoiser of Maggioni et al. However, none of these methods fully takes advantage of the 4D structure of the light field. Recent improvements in light field denoising performance are thus based on a better exploitation of the 2D angular dimensions. Chen et al. [46] use two joint convolutional neural networks to denoise the light field along the angular and spatial dimensions respectively. Liu et al. [47] denoise light field 4D patches first using a tensor decomposition. The SAIs are then combined into a single high resolution image which is further denoised, and finally projected back into denoised SAIs at the original resolution.

Any of the denoising methods cited above could be used in the proposed pipeline, but we choose the state of the art LFBM5D filter, which was shown to perform well on lenslet light fields by Alain et al. [16]. As in the BM3D filter of Dabov et al. or the VBM4D filter of Maggioni et al., the LFBM5D filter exploits the non-local self-similarities occurring in natural images, in addition to the spatio-angular redundancies. 5D patches built from similar 2D patches are filtered in the 5D transform domain, where their spectrum is

TABLE I: Details of the processing applied to the different groups of images or videos used for the validation of the proposed pipeline.

	Da	De	DeH	Re	DaN	DeN	ReN
Dansereau et al. [10]	✓				✓		✓
Our demultiplexing (Sec. III)		✓	✓	✓		✓	✓
HPR (Sec. IV)			✓	✓			✓
Recolouring (Sec. V)				✓			✓
Denoising (Sec. VI)					✓	✓	✓

very sparse and offer a good decorrelation between the true underlying signal and noise coefficients. Noise can thus be filtered by applying hard-thresholding on the 5D transform coefficients in a first step, and Wiener filtering in a second step. The LFBM5D output is then obtained by applying the inverse 5D transform on the filtered 5D spectrum.

The denoised light field is the output of the proposed pipeline, and we evaluate the full performance of the pipeline in the next section, as well as its preprocessing advantages for several light field applications in Section VIII.

VII. VALIDATION OF THE PROPOSED PIPELINE

We use a variety of metrics and experiments to validate the effectiveness of our pipeline. 17 light field sets were chosen from the EPFL [48] and INRIA [49] datasets captured with Lytro Illum cameras, as well as datasets captured using our own Lytro Illum camera; those include one set featuring non-Lambertian objects, in order to study the effect of these on selected applications. A metric analysis of 10 light fields from the recent Stanford dataset [50] can also be found in the supplementary material. A more detailed review of the dataset used here, an extended clean and corrected dataset, and code for our pipeline is available online².

In order to validate the different steps of the proposed pipeline, we consider the following seven combinations of settings (see Table I) : 1- Da) demultiplexing of Dansereau et al. [10], 2- De) proposed demultiplexing (Sec. III), 3- DeH) proposed demultiplexing + Hot Pixel Removal (HPR) (Sec. IV), 4- Re) proposed recolouring (Sec. V), 5- DaN) toolbox of Dansereau et al. + our denoising (Sec. VI), 6- DeN) our demultiplexing + our denoising, and 7- ReN) our full pipeline (demultiplexing, HPR, recolouring, denoising).

Note that other demultiplexing methods have been presented in [7]–[9]. However, similarly to [10], they do not consider the issues of wrong white balance and exposure, saturated highlights, colour inconsistencies, hot pixels and noise. Therefore, this section only presents comparisons against the method [10] which we have built upon. Nevertheless, further review and evaluation of the relevant tools in [7]–[9] as well as the more recent Plenopticam software³ [51] are given in the supplementary materials.

A. Colour Consistency

We first show in Fig. 9 the importance of the simple normalisation steps proposed in Section III-A for the colour

²<https://v-sense.scss.tcd.ie/research/light-fields/high-quality-light-field-extraction/>

³code available at: <https://github.com/hahne/plenopticam>



(a) Dansereau et al. vs Lytro Desktop

(b) Our method vs Lytro Desktop

Fig. 9: Below red line: refocused image from Lytro Desktop proprietary software (using ‘as shot’ white balance option). Above red line: central SAI of the *bee_2* light field obtained with (a) Dansereau et al.’s method [10], (b) our method. (Standard sRGB gamma correction is performed in both cases.)

balance and overall brightness. For reference, the bottom right part of each sub-figure shows a refocused image obtained by the Lytro proprietary software with the intended colours, i.e. as displayed by the camera when taking the picture. Note that the results of Dansereau et al. [10] are often wrongly assumed to be gamma corrected, leading to exaggerated contrasts and colour saturation. For a fair comparison, we performed standard sRGB gamma correction for both methods.

We used several metrics to evaluate the colour accuracy of our processed pipeline results including PSNR, SSIM [52], S-CIELab [53] and a histogram distance metric. For each metric, we use the centre SAI as reference and compute the distance between it and all other SAIs in the light field, and averaged the results over all SAIs. We used the centre view as reference for these metrics since the colours in the centre view are the most accurate and are not affected by the colour fading artefacts present in the outside views. Disparity differences between the centre view and all other SAIs may affect the evaluation, but since all methods are compared on the same set of light fields with the same disparity differences, metric values are still indicative of colour correction accuracy. PSNR and SSIM were computed per colour channel and averaged. The results can be seen in Fig. 10. As PSNR, S-CIELab and SSIM capture local colour differences between images, their accuracy can be affected by disparity changes between SAIs. As a result we have also included a global histogram distance which is more robust to changes in the image. For a pair of images, to compute this histogram distance we calculated the average chi-square differences between their L^* , a^* and b^* histograms, each computed on 25 bins.

In Fig. 10, we compare the colour consistency of results generated with Da , DeH , Re and ReN . In terms of PSNR, SSIM, and S-CIELab, Da performs the worst in all cases, followed by DeH , Re and ReN , confirming that each step of our pipeline improves the consistency of the light field and its fidelity with the centre SAI. The histogram distance results tell a similar story, with the initial decoding methods Da and DeH performing the worst in general, followed by Re and ReN . However, this metric indicates that in some cases, Da and DeH are more consistent than Re and ReN (*raoul* and *la_guin*).

Upon close inspection we found that some colour inconsistencies present after decoding (DeH) were not successfully removed after recolouring (Re) due to the smooth, global nature of our thin plate spline colour transfer function which ensures that similar colours in the image cannot become very

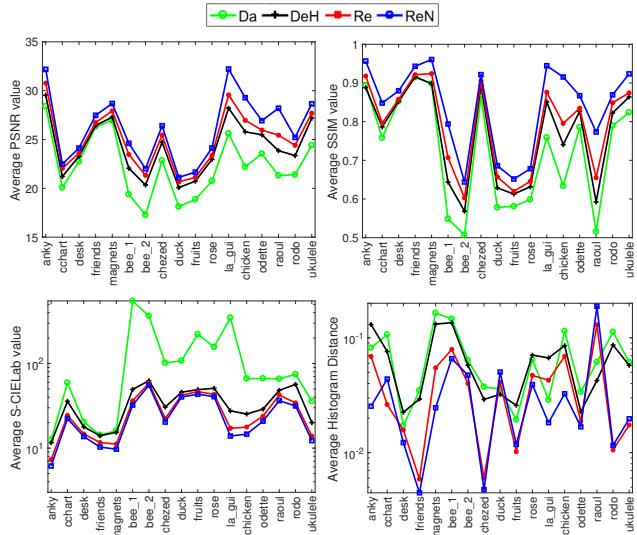


Fig. 10: Metric comparison, using PSNR, SSIM [52], S-CIELab [53] and histogram distance. Higher values are better in terms of PSNR and SSIM, and lower are better for S-CIELab and the histogram distance.



Fig. 11: The centre SAI in *raoul* is overlaid in column blocks onto one of the outside SAIs before recolouring (a) and after recolouring (b). The colours at the bottom of the images indicate which SAIs the columns are taken from - the centre SAI (blue), the outside SAI before recolouring (green) or the outside SAI after recolouring (red). The colour differences in the red background between the centre and outside SAIs in (a) are successfully removed in (b) but slight reddish tones are introduced into the cat's fur.

different after recolouring. For example, in the *raoul* light field, large portions of the red background were darker in colour in the outside SAIs (see Fig 11 (a)). After recolouring using our technique, the dark red regions were brightened to match the centre image, but this also caused the brown colour of the cat's fur, which has pixels similar in colour to the dark brown in the background, to become more red (see Fig 11 (b)). Therefore, although large portions of the recoloured outside and centre SAIs are similar, other smaller regions may still differ slightly in colour. This explains the spike in colour consistency appearing in the local histogram metrics for the *raoul* and *la_guin* light fields. However, we found that these artefacts do not occur regularly, and even when they are present, our propagation technique ensures colours change gradually across the light field SAIs, with neighbouring images displaying similar colours with only slight colour variations. Our subjective experiments also highlight that even in these cases, the recoloured SAIs are more pleasing than those without recolouring (see Table IV).



Fig. 12: Recolouring examples on the *cchart* and *bee_2* light fields. The first column shows the centre SAI (red and blue lines are used to create the EPIs in Fig. 13); the second column is one of the external views, notice the apparent washing out of the colours compared to the centre view; the third column is the same view after our recolouring, restoring most of the original colours.

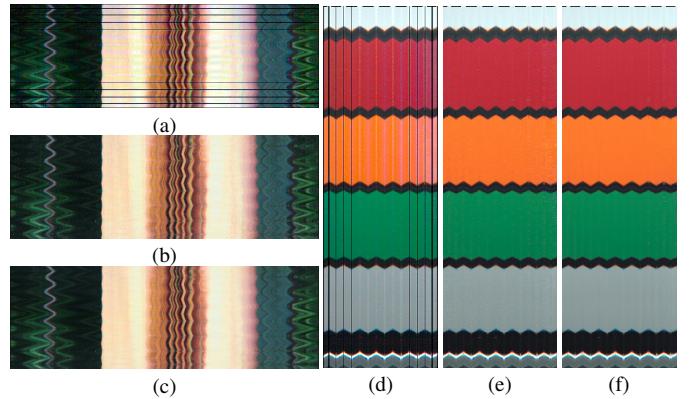


Fig. 13: Stacked EPIs showcasing colour differences in the *bee_2* (a,b,c) and *cchart* (d,e,f) light fields: after our RAW demultiplexing (a,d), after recolouring (b,e), and after denoising (c,f). Dark lines in (a,d) are caused by the dark SAIs in the corner of the light field (see Fig. 15) which are corrected by our recolouring. Selected lines are shown in Fig. 12.

Overall, we see that each step of the proposed pipeline improves colour consistency and reduces the colour or histogram distances while improving the structural similarity by bringing brightness and contrast to similar levels, and overall lowering pixel-wise error.

We visually assess the results of our recolouring method in Figs. 7, 12 and 13. The results are visually pleasing, with smooth transitions between consecutive views, seen in Fig. 7, and the colours overall remaining consistent with those in the centre view (see also Fig. 12). This is particularly visible when computing EPIs (as seen in Fig. 13), which consist of stacks of the same horizontal or vertical line of pixels taken across all the views of the light field. These images show a clear improvement in colour consistency over the whole light field, which is further improved after the denoising process.

B. Noise Analysis

1) *Analysis on a ground truth noise free dataset:* Since the light fields captured with the Lytro camera do not have a noise free ground truth, we propose to quantify the noise level by performing blind noise level estimation. For that purpose we use the method of Chen et al. [54], which estimates the noise

TABLE II: Noise level σ_{est} estimated using [54] for each light field and each setting combination described in Table I. The 3 setting combinations including denoising are shown on the right.

σ_{est}	<i>Da</i>	<i>De</i>	<i>DeH</i>	<i>Re</i>	<i>DaN</i>	<i>DeN</i>	<i>ReN</i>
anky	2.62	1.91	1.90	1.82	0.86	0.48	0.51
cchart	1.92	1.99	1.99	1.53	0.54	0.63	0.65
desk	3.45	3.06	3.11	2.85	1.18	0.97	1.12
friends	3.18	3.02	3.03	2.93	1.96	1.91	1.96
magnets	2.86	2.85	2.85	2.86	1.87	1.89	2.02
bee_1	2.08	2.13	2.13	1.80	0.79	0.89	0.97
bee_2	8.53	5.73	5.69	3.97	6.68	3.42	2.04
chezed	8.20	5.44	5.41	3.72	6.29	3.06	1.55
duck	6.60	5.40	5.49	5.74	5.76	4.66	5.05
fruits	5.87	4.36	4.39	3.48	4.45	3.11	2.50
rose	5.12	3.91	3.92	3.38	3.28	2.29	2.01
la_guin	4.07	3.06	3.06	2.47	1.64	0.88	0.71
chicken	4.90	3.29	3.32	2.86	3.23	1.80	1.69
odette	5.93	4.29	4.20	2.99	3.98	2.25	1.48
raoul	3.94	3.18	3.22	3.09	2.53	2.02	2.03
rodo	8.12	6.26	6.23	3.79	6.21	4.10	1.85
ukulele	4.42	3.43	3.44	3.45	2.94	2.14	2.24
Average	4.81	3.72	3.73	3.10	3.19	2.15	1.79

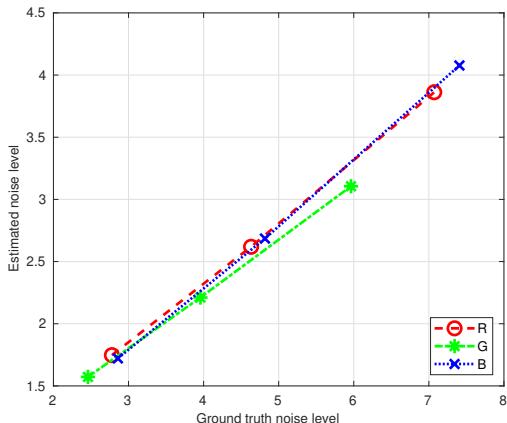


Fig. 14: Blind noise level estimation [54] plotted against the ground truth noise level, averaged over the 5 light fields of the noisy dataset. Although the no-reference metric from [54] does not estimate the exact noise level, it can be used for relative comparison.

level of an image based on the eigenvalues of the covariance matrix of the image patches, based on an Additive White Gaussian Noise (AWGN) model.

To first validate the assumption that the noise of the Lytro camera follows the AWGN model, we created a noisy light field dataset consisting of 5 scenes. For each scene, 3 different noise levels were created by changing the ISO gain and maximising the shutter speed so that the image is as bright as possible without saturation. For each scene and ISO setting, ~30 noisy instances were captured, and a ground truth noise free light field was created by averaging the noisy instances. We ensured that the lighting conditions remained stable. The light field noise can then be obtained by removing the noise free light field from the noisy instance. By analysing the histograms of the light field noise, we observed that the AWGN model is validated for each SAI of the light field. By fitting a normal distribution to the histograms, we then obtained the ground truth noise level for each colour channel as the standard deviation of the normal distribution. More details

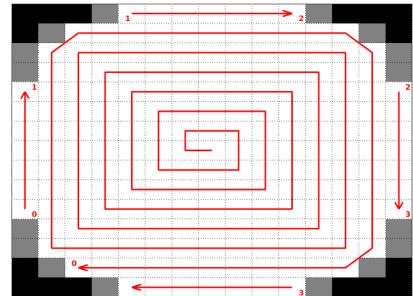


Fig. 15: View of the matrix of SAIs with the pattern of progression used to create the subjective test videos. Black and dark corner images are ignored for comfort to avoid flickering.

on the dataset are given in the supplementary material.

Finally, we evaluated the chosen blind metric [54] by comparing the estimated noise level to the ground truth. The graph of Fig. 14 shows the estimated noise level, averaged over all SAIs and all light fields, against the ground truth noise level. While the blind metric does not evaluate the exact noise level, a near linear relationship between the ground truth and estimated noise level can be observed, which validates the use of the chosen metric for the evaluation of our pipeline.

2) *Noise level estimation of the proposed pipeline:* Here we estimate the noise level after each step of the pipeline using the blind metric [54]. The noise level of the whole light field is computed by first independently estimating the noise level of each SAI, and then averaging the results. Results are shown in Table II for all setting combinations described in Table I and all 17 test light fields.

We observe that our proposed demultiplexing method can slightly reduce the noise level compared to Dansereau. The hot pixel removal step does not impact the noise level significantly, since the hot pixel noise is very different from AWGN. The noise level is again slightly decreased after colour correction, but overall the order of magnitude of the noise level remains unchanged for all these steps. In some cases the noise is even amplified after the colour correction, which further justifies applying denoising last, e.g. *cchart*, *chezed*. A clear reduction of the noise level is observed for all approaches after applying the LFBM5D filter. Overall, our full pipeline provides the smallest noise level compared to applying denoising on the demultiplexing of Dansereau et al. [10] or on our proposed demultiplexing approach. A visual comparison before and after denoising is shown in Fig. 13.

C. Subjective Evaluation

We evaluated the pipeline using a subjective experiment. For this we crafted videos showing all SAIs, starting from the centre view and following an expanding snail-like pattern going clockwise toward the external views (see Fig. 15).

This pattern was chosen instead of a more traditional snake-like pattern going from line to line because it highlighted our modifications of the external views more clearly, and offered smoother transitions. To stay consistent across all methods, and to reduce discomfort, we decided to ignore the four black and four dark views in each corner. Only our recolouring step fixes the dark views and keeping them in the videos causes

unnecessary flickering for the other methods. The videos were created with 25 fps for comfort and were therefore approximately seven seconds long. Using the datasets described in Sec. VII, this resulted in 119 videos so the full session lasted approximately 30 minutes, including time for explanations, setup, a short training and comments at the end.

We collected data from 22 voluntary participants (14 men, 8 women) who were tested for visual acuity and colour blindness, and rated these videos in a traditional side-by-side pairwise comparison experiment. We used Psychtoolbox for Matlab in order to ensure the videos were properly synchronised. To reduce bias, we asked the participants to rate the videos based on their personal appreciation, instead of guiding them toward looking for specific artefacts or particular sets of colours. The only emphasis was put on image quality consistency along the videos. The participants were then guided through a short training session to ensure they understood the task at hand, and the controls to perform it. The experiment took place in a dark room as recommended by ITU [55]. The screen was colour-calibrated beforehand.

The responses were processed using a freely accessible tool performing Thurstonian Case V scaling for pair-wise comparison experiments developed by Perez-Ortiz et al. [56]. After the scaling, just-objectionable-difference scores (JOD), as described by Perez-Ortiz et al., are obtained for each case. A difference of 1 JOD means that one option is selected over another with 75% probability (the mid-point between random guess and certainty). The relationship between the preference probabilities and the JOD follows the Gaussian cumulative distribution function, and the exact JOD values are found through a maximum likelihood estimation as explained in the work of Perez-Ortiz et al. The outcome is summarised in Table III and Fig. 16.

The results show that our demultiplexing is preferred by the subjects more than the one by Dansereau. However, occasionally, some participants mentioned in the comments that they preferred the over-saturated colours obtained with Dansereau et al.'s [10] method more than ours. The results indicate that our hot pixel removal tool has a positive effect of similar magnitude when applied to our demultiplexing. The colour correction step has the biggest effect on the pleasing factor, against all other settings, but even more significantly when associated with the previous steps of our pipeline. Finally, our final denoising step, in all scenarios, shows a level of improvement comparable to that of our demultiplexing and hot pixel removal tool. Overall, we can conclude that SAIs processed using our pipeline are significantly more appealing than when processed with the toolbox of Dansereau et al.

The significance of the results were also analysed by the statistical significance analysis proposed by Perez-Ortiz et al. and reported in Fig. 16. In this figure, the face values indicate the JOD difference, $JOD_i - JOD_j$, between the i^{th} row and j^{th} column, where positive values indicate that the settings in the row are better than that in the column and negative values indicate the opposite. Black boxes indicate this difference is statistically significant. The results show that overall, all of the proposed steps bring a statistically significant difference compared to the previous step. We can easily see

TABLE III: Subjective experiment results: just-objectionable-difference (JOD). First column is 0 we use it as reference for comparison. Negative values indicate the reference (in this case *Da*) was preferred over the method, while positive values indicate the method was preferred over the reference. For explanation and settings details refer to Section VII-C and Table I.

	<i>Da</i>	<i>De</i>	<i>DeH</i>	<i>Re</i>	<i>DaN</i>	<i>DeN</i>	<i>ReN</i>
anky	0	0.64	0.61	2.05	-0.17	0.8	3.07
cchart	0	0.09	1.26	1.95	-0.1	0.38	2.44
desk	0	-0.01	-0.2	0.04	0.14	0.28	1.11
friends	0	0.8	0.7	2.21	0.92	1.31	2.11
magnets	0	0.67	1.25	2.69	0.84	0.94	3.37
bee_1	0	1.35	2.92	3.99	0.7	2.7	4.89
bee_2	0	7.95	8.1	10.03	0.33	8.75	10.14
chezed	0	-0.03	-0.16	1.12	0.41	0.05	1.26
duck	0	0.01	0.22	1.14	1.16	0.1	0.83
fruits	0	0.06	-0.63	1.05	0	-0.72	1.69
rose	0	0.55	0.49	1.5	0.71	0.25	1.63
la_guin	0	0.48	1.13	2.31	0.22	1.53	3.51
chicken	0	0.14	1.37	2.59	1.2	0.96	2.89
odette	0	-0.13	0.23	0.77	1.58	0.13	1.36
raoul	0	0.94	2.82	5.26	1.19	0.84	6.45
rodo	0	0.95	1.24	2.54	-0.44	1.54	1.98
ukulele	0	-1.15	-0.5	0.59	0.84	0.44	1.15
Overall	0	0.32	0.64	1.72	0.55	0.72	2.06

	<i>Da</i>	<i>De</i>	<i>DeH</i>	<i>Re</i>	<i>DaN</i>	<i>DeN</i>	<i>ReN</i>
<i>Da</i>	0	-0.31	-0.64	-1.72	-0.55	-0.72	-2.06
<i>De</i>	0.31	0	-0.33	-1.41	-0.23	-0.4	-1.74
<i>DeH</i>	0.64	0.33	0	-1.08	0.1	-0.07	-1.41
<i>Re</i>	1.72	1.41	1.08	0	1.17	1	-0.34
<i>DaN</i>	0.55	0.23	-0.1	-1.17	0	-0.17	-1.51
<i>DeN</i>	0.72	0.4	0.07	-1	0.17	0	-1.34
<i>ReN</i>	2.06	1.74	1.41	0.34	1.51	1.34	0

Fig. 16: Overall JOD score differences for all contents and subjects, where the face value indicates $JOD_i - JOD_j$, between the i^{th} row and j^{th} column. Positive values indicate the settings of the row are better than that of the column. Black boxes specify that this difference is statistically significant. Refer to Section VII-C for analysis.

that the whole pipeline (i.e., *ReN*) is superior to all cases, and recolouring is also found to be significantly better than the *DaN* and *DeN* cases, which shows that the effect of recolouring is critical for human perception.

D. Aesthetic Appeal

As an additional way to compare our results to the previous state of the art, we use a recent neural metric by Talebi et al. [57] that focuses on the aesthetic aspect of images called NIMA. A summary of this analysis can be found in Table IV.

NIMA simulates an estimation of a group of people's ratings for aesthetic appeal based on its pleasing factor, and thus gives an average score as well as standard deviation for each image. We obtain our results by testing each individual SAI, and average the results to get a unique score for each light field. NIMA can also be used as a metric to measure noise level,

TABLE IV: NIMA results. For better reading, indicated in bold black are the best scores, and in italic blue the worst ones. The values represent for each content the average of individual views' scores. For settings details refer to Section VII-C and Table I.

	<i>Da</i>	<i>De</i>	<i>DeH</i>	<i>Re</i>	<i>DaN</i>	<i>DeN</i>	<i>ReN</i>
anky	4.67	<i>4.57</i>	<i>4.57</i>	4.81	5.49	5.47	5.56
cchart	<i>5</i>	5.05	5.05	<i>5.07</i>	5.23	5.3	5.32
desk	4.7	4.78	4.78	4.83	4.99	5.05	5.16
friends	<i>5.15</i>	5.4	5.39	<i>5.52</i>	5.25	5.47	5.61
magnets	4.22	<i>4.12</i>	<i>4.12</i>	4.27	4.96	4.85	4.97
bee_1	<i>4.26</i>	4.35	4.35	4.46	4.53	4.77	5.05
bee_2	4.5	<i>4.36</i>	<i>4.36</i>	4.46	4.47	4.46	4.6
chezed	<i>5.27</i>	5.33	5.29	5.32	5.34	5.4	5.47
duck	<i>4.84</i>	4.86	4.85	<i>5</i>	4.96	4.98	5.12
fruits	4.66	4.52	4.52	4.52	4.63	<i>4.46</i>	4.47
rose	4.93	4.88	4.88	4.84	4.75	4.67	<i>4.66</i>
la_guin	4.66	<i>4.3</i>	4.32	4.44	4.92	4.95	5.06
chicken	<i>3.98</i>	4.03	4.02	4.18	4.06	4.73	4.95
odette	4.89	<i>4.78</i>	4.79	<i>4.87</i>	4.97	4.84	4.95
raoul	4.24	4.16	4.16	4.31	<i>4.12</i>	4.19	4.7
rodo	4.38	4.38	4.37	<i>4.36</i>	4.39	4.38	4.37
ukulele	4.55	<i>4.5</i>	<i>4.5</i>	4.58	5.36	5.19	5.33
Average	4.64	4.61	<i>4.61</i>	4.7	4.85	4.89	5.02

but since we are interested in the pleasing factor and have more dedicated metrics for noise analysis, we decided to use it by resizing the images instead. As suggested by the authors, each SAI is resized from 625×434 to 224×224 before being evaluated by the pre-trained network, since this allows for the most accurate results based on aesthetic quality.

From Table IV, we can see that, with few exceptions, the results obtained using our full pipeline garner better scores compared to those processed by the toolbox of Dansereau et al [10]. On average, both the recolouring and denoising step improve the image quality, except in the case of the *fruits* and *rose* light fields in which Dansereau et al.'s method performs better. Images obtained with Dansereau et al.'s method have brighter, more saturated colours than those generated using our approach and the NIMA network can associate these unnatural colours with better aesthetic value. This is consistent with comments made by some participants of the subjective experiment described in Section VII-C.

E. Computation time

We report here the average computation times for each part of the pipeline. Most of the steps were implemented in Matlab, and the denoising was implemented in C++. Our demultiplexing step takes $\sim 2'05''$ per light field, whereas in comparison the demultiplexing of Dansereau et al. takes $\sim 1'10''$. The difference is essentially explained by the White Image-guided interpolation. The HPR step runs in $\sim 1'40''$. Correspondences between neighbour views and with centre view (2 sets per view to recolour) are computed in $\sim 5'45''$. The recolouring step runs in $\sim 234'$ ($\sim 60''$ per SAI) and finally the denoising step takes $\sim 50'$.

Possible optimisation includes parallelisation of the colour correction step, as several rows could be processed at the same time, once the centre column images are available. GPU implementation would also speed up the process of the propagation step, or the denoising. Finally, our implementation of the recolouring uses all the available correspondences, when a fraction could be selected to reduce the computation time,

albeit with reduced quality. Finally, in this paper we have proposed using CIELAB space colour values when estimating the colour transfer function to ensure the best results. Reducing the colour space representation from three channels to two could also provide significant computational speed up and would be an interesting avenue for future investigation.

VIII. APPLICATIONS

A. Rendering

One of the first light field applications was the ability to synthesise new images corresponding to novel viewpoints in real time, without requiring any 3D model of the scene, as described by Levoy et al. [1]. For each pixel in the novel image, the intersection of the corresponding light ray and the two light field planes is computed. The intersection with the camera planes allows the closest available SAIs to be found, while the closest pixel positions are computed from the intersection with the image plane. The final value of a pixel in the novel image is then computed by interpolating between the nearest SAIs and the nearest pixels.

In this experiment, we rendered novel views corresponding to a camera close to the object of interest and moving horizontally from left to right. We show a few rendered images for the *cchart* and *bee_2* light fields in Figs. 17 and 18 respectively. On the top row, results obtained for a light field decoded with the toolbox of Dansereau et al. [10] (*Da*) are displayed, and on the bottom row results obtained with our full pipeline (*ReN*). As rendered images are created from multiple source SAIs, clear colour inconsistencies appear in images rendered from Dansereau, but also in between the different novel viewpoints. In addition, images rendered from our pipeline are less affected by the dark SAIs in the corners of the light field.

B. Compression

Due to the large amount of information contained in light fields, their compression is essential for a large scale adoption of this image format. However, aforementioned artefacts in existing plenoptic data are likely to reduce the efficiency of traditional compression methods. In order to evaluate the impact of our quality enhancement tools on the compression performance, we have used a common light field compression method presented by Liu et al. [58]. This method forms a pseudo video sequence from the light field's SAIs and encodes the sequence using the HEVC video coding standard, therefore taking advantage of the redundancies between SAIs.

For this experiment, we have encoded three different versions of each light field corresponding to *Da*, *De* and *Re* in Table I (i.e. demultiplexing of Dansereau et al. [10], our demultiplexing only, and our demultiplexing followed by hot pixel removal and colour consistency correction). Each version was encoded several times with different bitrates by varying the QP parameters in HEVC over the values $\{12, 16, 20, 24, 28, 32, 36\}$. In order to evaluate the quality of the decoded light field, we compute the peak signal to noise ratio (PSNR) using as a reference, the uncompressed light field of the corresponding version. The experiment was performed



Fig. 17: Novel viewpoints rendered from the *cchart* light field, moving from left to right. Top: Dansereau et al. [10] (*Da*). Bottom: ours (*ReN*). Colour inconsistencies inside and across viewpoints are highlighted in red.



Fig. 18: Novel viewpoints rendered from the *bee_2* light field, moving from left to right. Top: Dansereau et al. [10] (*Da*). Bottom: our full pipeline (*ReN*). Colour inconsistencies inside and across viewpoints are highlighted in red.

for 12 light fields including 4 from the EPFL dataset, 4 from the INRIA dataset and 4 from our captures (V-SENSE).

Note that the PSNR is computed from a different uncompressed reference for each version. However, our experiments in Section VII have shown that our modified demultiplexing as well as our additional hot pixel removal and colour consistency correction steps improve the subjective quality in the uncompressed case. Here, we assume that the relative perceived quality of the three version *Da*, *De* and *Re* are unchanged when they are altered with similar compression losses. Therefore, we consider that for the same PSNR scores, the quality of the compressed light fields *De* and *Re* will not be perceived as worse than that of *Da*. Furthermore, the results in Table V show that, on average, the light fields in *De* and *Re* require respectively 38% and 68.6% less bitrate to be encoded with a similar PSNR as *Da*. This clearly demonstrates that the

TABLE V: Bitrate savings obtained for light fields extracted with our demultiplexing (*De*) and with our hot pixel removal and colour correction (*Re*). The gains are computed with the Bjontegaard metric [59] with respect to light fields extracted using Dansereau et al's method [10] (*Da*). These results assume that similar PSNR for each version (*Da*, *De*, *Re*) correspond to similar perceived quality.

		<i>De</i>	<i>Re</i>
EPFL	bikes	-0.9%	-29.1%
	fountain&vincent_2	8.9%	-33.7%
	stone_pillars_outside	-19.8%	-59.2%
	vespa	10.8%	-50.1%
INRIA	bee_2	-67.7%	-92.2%
	bumblebee	-36.1%	-78.1%
	duck	-52.1%	-80.3%
	fruits	-62.3%	-81.1%
V-SENSE	cherry_tree	-35.8%	-55.4%
	chicken	-83.1%	-98.7%
	rodo	-51.2%	-72.2%
	wine_bottles	-67.3%	-93.4%
Average		-38%	-68.6%

enhanced quality resulting from both our demultiplexing and post processing steps also has a very beneficial impact on the light field compression.

C. Super-Resolution

Light fields captured by lenslet cameras have a poor spatial resolution due to the multiplexing of both spatial and angular information on a single sensor. Spatial super-resolution of light fields captured with a lenslet camera is thus a common application.

In this experiment, we used the extension of the LFBM5D denoising filter to spatial super-resolution presented by Alain et al. [60]. This method uses the sparse coding of the LFBM5D filter as a prior to solve the ill-posedness of super-resolution. A two-step iterative algorithm alternating between a LBM5D filtering step and a back-projection step is used to obtain the super-resolved light field.

We show results for a single SAI of the *raoul* light field in Fig. 19. The super-resolution result (right) is compared to a simple bicubic upsampling (left). Results obtained with the toolbox of Dansereau et al. [10] (*Da*) are displayed on the top row, with results for our full pipeline (*ReN*) on the bottom row. The benefit of our pipeline is clearly visible, especially in terms of hot pixels and noise removal. This is due to a general side effect of super-resolution which amplifies the high frequency corresponding to noise. This is common to all super-resolution methods, not only the one used here.

D. Light Field Editing

Light field editing is another important application in light field imaging, with works by Jarabo et al. [61] or Zhang et al. [62]. To determine whether our proposed pipeline provides any advantages for light field editing applications, we applied the recent editing technique of Frigo et al. [63] to both our processed light fields and those processed with Dansereau et al.'s method [10]. The technique proposed by Frigo et al. [63] allows the user to edit the centre SAI of the light field, either via image recolouring or inpainting, and propagates the edits to the remaining views using a structure tensor driven diffusion on the EPIs. Some light field editing results can be seen in Fig. 20.

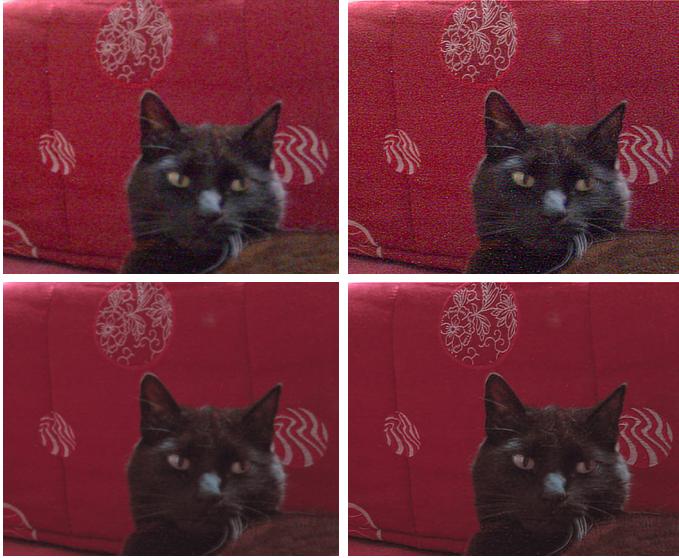


Fig. 19: Spatial super-resolution (right) of the *raoul* light field compared to a simple bicubic upsampling (left). Top: Dansereau et al. [10] (*Da*). Bottom: our full pipeline (*ReN*).

Due to the strong colour differences between the centre SAI and the external views of light fields obtained using Dansereau et al.'s method (*Da*), the tensor-driven diffusion becomes inaccurate at the edges of the light field, causing unwanted warping of the SAIs (Fig. 20, column 1). The strong colour differences between SAIs also means that when colours from the centre SAI are propagated to other SAIs, they do not blend seamlessly with the rest of the image, creating strong colour inconsistencies (Fig. 20, column 1, see inpainting results). Interestingly, we also found that when editing light fields generated using our full pipeline, including denoising (*ReN*), unwanted warping artefacts are also created (Fig. 20, column 3). As with any denoising algorithm, small image details can also be removed with noise, some of which are needed by the tensor diffusion step in the edit propagation software proposed by Frigo et al. Removing these details creates inaccuracies and causes artefacts. On the other hand, edit propagation results applied to our pipeline before denoising (*Re*) are the best (Fig. 20, column 2). The consistent colours across these light fields ensure that the edits are propagated correctly, and that no inconsistent colours can be seen in the edited SAIs, even towards the outside of the light field. This indicates that if using a similar editing approach, edit propagation should be applied after our recolouring step, with denoising applied as a final step.

E. Depth / disparity estimation

We evaluate here the performance of the proposed pipeline on depth or disparity estimation, which is one of the flagship applications for light fields. For that purpose we use 4 different methods [64] [65] [66] [34] applied after every step of the pipeline. For all methods we used the code provided by the authors. The first method estimated the depth by simply computing the slopes of the EPIs based on the light field gradient [64]. Note that the code provided by the authors implements the first step described in the paper and only outputs a sparse estimation. The second method was designed

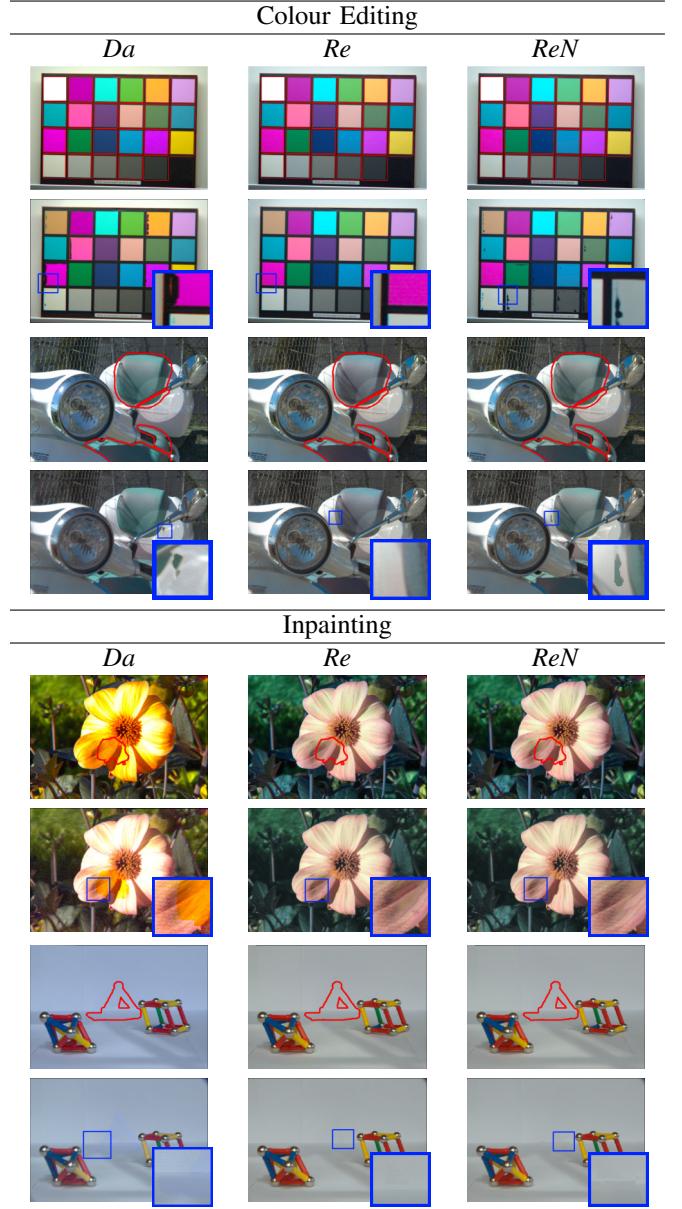


Fig. 20: Light field editing results using the edit propagation method of Frigo et al. [63]. For each light field, the top row shows the user edits made to the centre SAI of the light field, with red lines indicating the mask used during the propagation process. The second row shows a sample SAI from the light field after the edit propagation.

to be robust to occlusions by analysing the statistics of angular patches of the light field together with refocus cues [65]. The third method uses the spinning parallelogram operator to estimate the slopes of the EPIs and provide a robust depth estimate [66]. Finally, the fourth method adapted optical flow techniques to estimate the disparity on row or columns of the light field [34].

Fig. 21 shows the results for the four methods on the *bee_2* light field. Results for 7 additional light fields are available in the supplementary material. For each method, the depth or disparity was estimated for the centre SAI of the light field decoded with the toolbox of Dansereau et al. [64] without (*Da*) and with denoising (*DaN*), our demultiplexing (*De*), and our full pipeline without (*Re*) and with denoising (*ReN*). Note

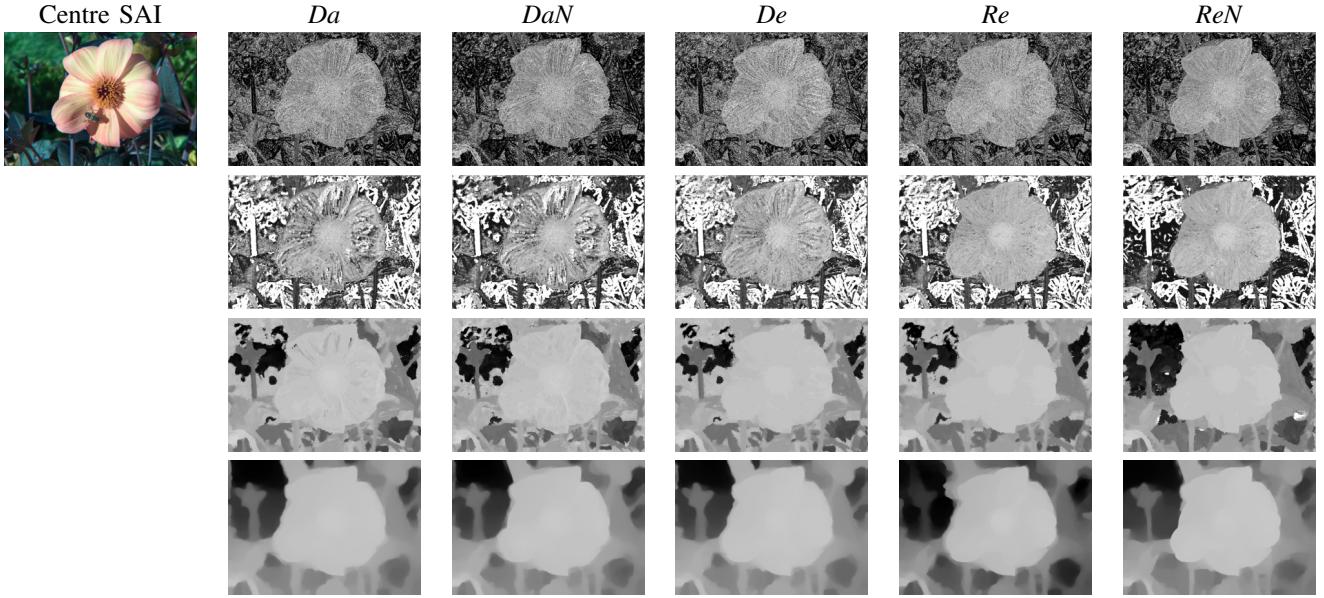


Fig. 21: Depth maps estimated for different steps of the pipeline on *bee_2* using, from top to bottom: [64], [65], [66], and [34].

that all results were colour coded so that close objects appear white, while far objects appear black.

Since no ground truth is available for the depth or disparity maps, no objective evaluation could be conducted. For each method, slight variations can be observed between the depth or disparity maps corresponding to the different steps, but no step seems to clearly deter or improve the performances. Note that this is also true after the denoising step, even though denoising is sometimes not recommended before such applications. While in general denoising may smooth images, the LFBM5D algorithm chosen in this paper can preserve edges, which are useful features for most depth or disparity estimation methods. Thus the proposed pipeline does not seem to strongly impact the performances of depth or disparity map estimation.

IX. CONCLUSION

We presented a high quality light field extraction pipeline aimed at reducing or removing the various artefacts, colour inconsistencies and noise that are prevalent in the typical output from plenoptic cameras. We provide and analyse several tools that can be used either on their own or in conjunction with each other for increased effect, and we show that each of the steps is necessary to ensure the best possible image quality. We also highlight the importance of the order in which each step is performed within the pipeline. We have proven, using a number of metrics, as well as a subjective experiment, that our results outclass those obtained from the previous state of the art tools, and finally make the entirety of the sub-aperture views usable by the user. We note that both the recolouring and denoising steps in our pipeline can be applied to light fields captured with camera arrays or gantries, and are not limited to plenoptic light fields. Finally we demonstrate that using higher quality light fields enhances the quality of the results for a number of classic light field applications, and therefore expect that this improvement will allow the research

community to be keener to use these cameras and data for their work.

REFERENCES

- [1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, 1996, pp. 31–42.
- [2] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, 1991, pp. 3–20.
- [3] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 765–776, Jul. 2005.
- [4] "The stanford light field archive," <http://lightfield.stanford.edu/lfs.html>, accessed: 05-03-2019.
- [5] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," Stanford University CSTR, Tech. Rep., Apr. 2005.
- [6] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *Proc. IEEE ICCP*, 2009, pp. 1–8.
- [7] D. Cho, M. Lee, S. Kim, and Y. W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proc. IEEE ICCV*, Dec. 2013, pp. 3280–3287.
- [8] S. Xu, Z.-L. Zhou, and N. Devaney, "Multi-view image restoration from plenoptic raw images," in *Proc. ACCV Workshops*, 2014.
- [9] M. Seifi, N. Sabater, V. Drazic, and P. Perez, "Disparity-guided demosaicking of light field images," in *Proc. IEEE ICIP*, Oct. 2014, pp. 5482–5486.
- [10] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenslet-based plenoptic cameras," in *Proc. IEEE CVPR*, 2013, pp. 1027–1034.
- [11] "JPEG Pleno call for proposals on light field coding," Chengdu, China, Oct. 2016.
- [12] P. Matysiak, M. Grogan, M. L. Pendu, M. Alain, and A. Smolic, "A pipeline for lenslet light field quality enhancement," in *Proc. IEEE ICIP*, Oct. 2018, pp. 639–643.
- [13] 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, 2017.
- [14] P. David, M. L. Pendu, and C. Guillemot, "White lenslet image guided demosaicing for plenoptic cameras," in *Proceedings of the IEEE Int. Workshop on Multimedia Signal Processing*, Oct. 2017, pp. 1–6.
- [15] H. S. Malvar, L.-W. He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *Proc. IEEE ICASSP*, 2004.

- [16] M. Alain and A. Smolic, "Light field denoising by sparse 5D transform domain collaborative filtering," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, Oct. 2017, pp. 1–6.
- [17] Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev, "An analysis of color demosaicing in plenoptic cameras," in *Proc. IEEE CVPR*, Jun. 2012, pp. 901–908.
- [18] X. Huang and O. Cossairt, "Dictionary learning based color demosaicing for plenoptic cameras," in *Proc. IEEE CVPR Workshops*, Jun. 2014, pp. 455–460.
- [19] T. Lian and K. Chiang, "Demosaicing and denoising on simulated light field images," 2016.
- [20] J. Fu, Y. Wu, X. Mou, W. Ji, and P. Wang, "Fpga-based implementation of estimating saturated pixel values in RAW image," in *Digital Photography and Mobile Imaging XII*, 2016, pp. 1–6.
- [21] M. Assefa, T. Pouli, J. Kervec, and M. Larabi, "Correction of overexposure using color channel correlations," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2014, pp. 1078–1082.
- [22] E. Elboher and M. Werman, "Recovering color and details of clipped image regions," in *Proc. CGVCVIP*, 2010.
- [23] S. Z. Masood, J. Z., and M. F. Tappen, "Automatic correction of saturated regions in photographs using cross-channel correlation," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1861–1869, Oct. 2009.
- [24] A. R. Robertson, "The CIE 1976 color-difference formulae," *Color Research & Application*, vol. 2, no. 1, pp. 7–11, 1977.
- [25] M. Grogan and R. Dahyot, "L2 divergence for robust colour transfer," *Computer Vision and Image Understanding*, 2019.
- [26] M. Oliveira, A. Sappa, and V. Santos, "A probabilistic approach for color correction in image mosaicking applications," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 508–523, Feb 2015.
- [27] J. Park, Y. Tai, S. N. Sinha, and I. S. Kweon, "Efficient and robust color consistency for community photo collections," in *Proc. IEEE CVPR*, June 2016, pp. 430–438.
- [28] M. Xia, J. Y. Renping, X. M. Zhang, and J. Xiao, "Color consistency correction based on remapping optimization for image stitching," in *Proc. IEEE ICCV Workshops*, Oct 2017, pp. 2977–2984.
- [29] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim, "Probabilistic moving least squares with spatial constraints for nonlinear color transfer between images," *Computer Vision and Image Understanding*, 2019.
- [30] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123–137, Jul. 2007.
- [31] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," in *Scale Space and Variational Methods in Computer Vision*, 2013, pp. 428–439.
- [32] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 22–45, Jan 2015.
- [33] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim, "Color transfer using probabilistic moving least squares," in *Proc. IEEE CVPR*, June 2014, pp. 3342–3349.
- [34] Y. Chen, M. Alain, and A. Smolic, "Fast and accurate optical flow based depth map estimation from light fields," in *Proceedings of the Irish Machine Vision and Image Processing Conference*, 2017.
- [35] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proc. IEEE CVPR*, 2016, pp. 5704–5712.
- [36] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM Transactions on Graphics (TOG)*, vol. 28, 2009, p. 24.
- [37] S. H. Park, H. S. Kim, S. Lansel, M. Parmar, and B. A. Wandell, "A case for denoising before demosaicing color filter array data," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Nov. 2009, pp. 860–864.
- [38] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicing and denoising," *ACM ToG*, vol. 35, no. 6, pp. 191:1–191:12, Nov. 2016.
- [39] L. Condat and S. Mosaddegh, "Joint demosaicing and denoising by total variation minimization," in *IEEE International Conference on Image Processing*, Sep. 2012, pp. 2781–2784.
- [40] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [41] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1001–1013, 2014.
- [42] P. Jain and V. Tyagi, "A survey of edge-preserving image denoising methods," *Information Systems Frontiers*, vol. 18, pp. 159–170, 2016.
- [43] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE TIP*, vol. 21, no. 9, pp. 3952–3966, 2012.
- [44] Z. Li, H. Baker, and R. Bajcsy, "Joint image denoising using light-field data," in *Proc. IEEE ICME Workshops*, 2013.
- [45] A. Sepas-Moghadam, P. L. Correia, and F. Pereira, "Light field denoising: exploiting the redundancy of an epipolar sequence representation," in *Proceedings of the 3DTV Conference*, Jul 2016, pp. 1–4.
- [46] J. Chen, J. Hou, and L. Chau, "Light field denoising via anisotropic parallax analysis in a cnn framework," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1403–1407, Sep. 2018.
- [47] Y. Liu, N. Qi, Z. Cheng, D. Liu, Q. Ling, and Z. Xiong, "Tensor-based light field denoising by integrating super-resolution," in *Proc. IEEE ICIP*, Oct 2018, pp. 3209–3213.
- [48] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proceedings of the International Conference on Quality of Multimedia Experience*, 2016.
- [49] M. L. Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1981–1993, April 2018.
- [50] D. G. Dansereau, B. Girod, and G. Wetzstein, "LiFF: Light field features in scale and depth," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019.
- [51] C. Hahne, A. Aggoun, V. Velislavljevic, S. Fiebig, and M. Pesch, "Baseline and triangulation geometry in a standard plenoptic camera," *International Journal of Computer Vision*, vol. 126, pp. 21–35, Jan 2018.
- [52] 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.
- [53] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *Journal of the Society for Information Display*, vol. 5, no. 1, pp. 61–63, 1997.
- [54] G. Chen, F. Zhu, and P. A. Heng, "An efficient statistical method for image noise level estimation," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 477–485.
- [55] ITU-R, "Methodology for the subjective assessment of the quality of television pictures," ITU-R Recommendation BT.500-13, Jan 2012.
- [56] M. Perez-Ortiz and R. K. Mantiuk, "A practical guide and software for analysing pairwise comparison experiments," arXiv:1712.03686, 2017.
- [57] H. Talebi and P. Milanfar, "NIMA: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, Aug 2018.
- [58] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *Proc. IEEE ICME Workshops*, July 2016, pp. 1–4.
- [59] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," document VCEG-M33, ITU-T VCEG Meeting, 2001.
- [60] M. Alain and A. Smolic, "Light field super-resolution via LFBM5D sparse coding," in *Proc. IEEE ICIP*, Oct 2018, pp. 2501–2505.
- [61] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?" *ACM TOG*, vol. 33, no. 4, 2014.
- [62] F. Zhang, J. Wang, E. Shechtman, Z. Zhou, J. Shi, and S. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE TVCG*, vol. 23, no. 5, pp. 1561–1573, May 2017.
- [63] O. Frigo and C. Guillemot, "Epipolar Plane Diffusion: An Efficient Approach for Light Field Editing," in *Proc. BMVC*, Sep. 2017.
- [64] D. Dansereau and L. Bruton, "Gradient-based depth estimation from 4d light fields," in *IEEE International Symposium on Circuits and Systems*, vol. 3, May 2004, pp. III–549.
- [65] T. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3487–3495.
- [66] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Computer Vision and Image Understanding*, vol. 145, pp. 148 – 159, 2016.