Light Field Inpainting Propagation via Low Rank Matrix Completion

Mikael Le Pendu, Xiaoran Jiang, Christine Guillemot

Light Field inpainting propagation via Low Rank Matrix completion

Mikael Le Pendu, Xiaoran Jiang, and Christine Guillemot

Abstract—**Building up on the advances in low rank matrix completion, this article presents a novel method for propagating the inpainting of the central view of a light field to all the other views. After generating a set of warped versions of the inpainted central view with random homographies, both the original light field views and the warped ones are vectorized and concatenated into a matrix. Because of the redundancy between the views, the matrix satisfies a low rank assumption enabling us to fill the region to inpaint with low rank matrix completion. To this end, a new matrix completion algorithm, better suited to the inpainting application than existing methods, is also developed in this paper. In its simple form, our method does not require any depth prior, unlike most existing light field inpainting algorithms. The method has then been extended to better handle the case where the area to inpaint contains depth discontinuities. In this case, a segmentation map of the different depth layers of the inpainted central view is required. This information is used to warp the depth layers with different homographies. Our experiments with natural light fields captured with plenoptic cameras demonstrate the robustness of the low rank approach to noisy data as well as large color and illumination variations between the views of the light field.**

Index Terms—**Light Field, Inpainting, Low rank matrix completion, homography.**

I. INTRODUCTION

LIGHT Field imaging is becoming increasingly popular thanks to the recent advances in Light Field capture of real scenes. While a traditional 2D camera loses a lot of information of the captured 3D scene (*e.g.* no depth information, limited depth of field), light field imaging devices are able to yield a much richer description of the scene in a 4D representation, enabling novel post-capture processing. For instance, refocusing, extended depth of field and different viewpoint rendering can be performed from a single capture [1]–[3]. A light field can be seen as an array of views of the scene (also called sub-aperture images) with densely sampled view points. For the sake of parametrization, a light field is often considered as a 4D function $L(x, y, u, v)$, where u and v represent the angular dimensions, corresponding respectively to a horizontal and vertical shift in the viewpoint. The x and y parameters are the horizontal and vertical axis in each view. Thus, for u, and v fixed respectively to u^* and v^*, $L(x, y, u^*, v^*)$ represents one sub-aperture image.

As the capture of 4D light fields from real scenes grows in popularity, the need for a generalization of traditional image editing tools is expected to rise as well. In particular,

for inpainting, which is the recovery of missing parts of an image, plenty of algorithms already exist in the 2D case. For instance, diffusion based methods [4] can efficiently recover thin holes. Exemplar based methods [5]–[7] are better suited for applications such as object removal thanks to their ability to fill larger holes using patches of texture taken from the known parts of the image. However, applying independently these methods to each sub-aperture image of a light field is very unlikely to result in a consistent inpainting in the angular direction (*i.e.* across sub-aperture images) as illustrated in Figure 1(b) and (c). In addition, in some challenging examples, where large holes in a background with complex geometry and textures need to be recovered, even advanced patch-based methods do not give satisfactory results. Therefore, a manual editing remains necessary. For light field inpainting, though, manually editing each sub-aperture image in a consistent manner is extremely tedious.

For those reasons, in our approach, only the inpainting of the central view is required and can be performed either manually or using existing 2D algorithms. Our method thus focuses on consistently propagating the inpainted region to all the other views of the light field, as in Figure 1. In a first step, a matrix is built where each column is a vectorized view. Additional images, and thus additional columns of the matrix, are constructed by warping the inpainted central view using randomly sampled homographies. In order to cope with the case where the area to inpaint contains depth discontinuities, an extension of the method is proposed where a segmentation map of the depth layers of the inpainted central view is required as additional input. It is used by our algorithm to warp each depth layer with a different random homography. Because of the redundancy existing between the different views of the light field and the warped versions of the inpainted view, the constructed matrix can be well approximated by a low rank matrix. Our algorithm then draws on the recent advances in low rank matrix completion to fill the unknown parts of the matrix corresponding to the region to inpaint in each view. Although a large number of low rank matrix completion algorithms have already been developed (*e.g.* [8]–[16]), they have been seldom used for inpainting. They are not suitable for this problem for at least one of the following reasons: either the rank must be known a priori or the method assumes the positions of the unknown matrix entries are uniformly distributed. In summary, the contributions of the paper include:

- The definition of a new low rank matrix completion algorithm suitable for completing large parts of a matrix without requiring the exact matrix rank.
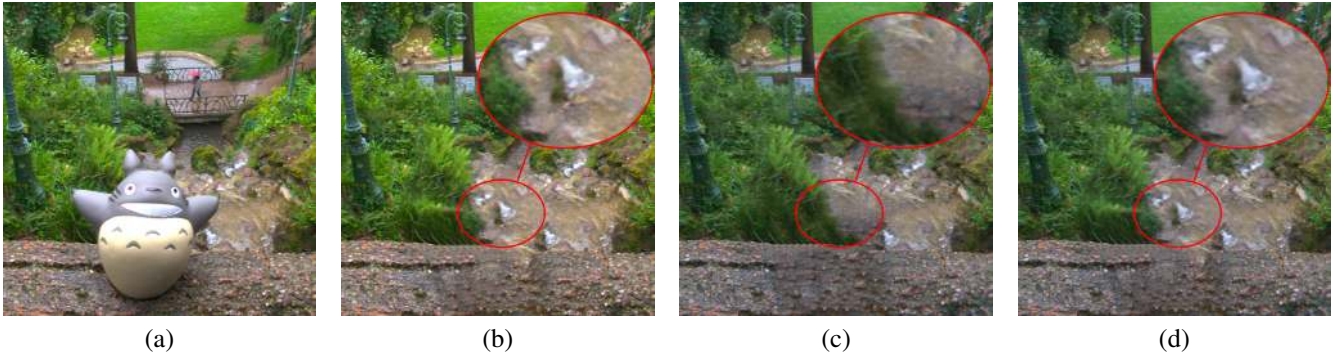
Fig. 1. Illustration of our inpainting propagation method : (a) Original central view (5,5). (b) Central view inpainted with [6]. Note that because of the short baseline, disocclusion is not available and a single image inpainting method must be used. (c) View (8,8) inpainted with [6]. (d) Propagated inpainting from central view to view (8,8). Our propagation method keeps consistent inpainting across views.

- The use of this algorithm for propagating an inpainted area from the central view to all the other views of a light field. In case of depth discontinuities within the inpainted area, an extension of the method using a depth layer segmentation map is also proposed.
- As an intermediate step of the inpainting propagation, we define a method for randomly generating homography warpings that are consistent with the displacement between views in actual light field data.

II. RELATED WORK

Light Field inpainting

Although the problem of image or video inpainting has already been extensively studied, only a few methods in the literature address the specific case of 4D light field data.

In [17], the authors propose a variational framework for solving various inverse problems including inpainting. By exploiting depth information, they defined constraints on the structure of the epipolar plane images of the light field, thus ensuring a consistent inpainting across views. This method can successfully recover missing parts of the light field as long as the regions unknown in some views are visible in several other views. However, for removing an object in all the views, the method no longer applies since a consistent inpainting of several views should already be available in order to infer depth information in the area to recover.

A different approach is used in [18], where the spatial consistency is ensured by minimizing a 4D patch consistency measure extending the bidirectional similarity measure of [19]. A depth estimation of each view of the light field is required to evaluate this measure.

In [20], the editing of the central view is first performed using a 2D patch based method, and the offsets between the filled patches and their best match in the known region are memorized. Using a layered depth map, the offsets are propagated to other views in order to fill them in a consistent manner. However, the method does not generalize to the case where the central view has been inpainted with an arbitrary method that does not provide the offset information.

Note that each of these algorithms require a depth map and are thus highly dependent on the quality of the depth estimation. Furthermore, the depth map must also be inpainted in the area to be removed. Another difference with our approach is that they do not propagate the inpainting result of a single view performed by an arbitrary method. Therefore, they do not allow a user to control the quality of the inpainting for one of the views in the complex cases where a fully automatic inpainting is not possible.

Concerning the propagation of one inpainted view (by an arbitrary method) to the rest of the light field, very few works have been conducted yet to the best of our knowledge. The authors of [21] inpaint an all-in-focus image and propagate it to the light field represented as a focal stack (*i.e.* set of refocused images) instead of processing directly the light field views. The method in [22] propagates user strokes performed on one view to edit all the light field views, but the method only applies to the re-colorization application. In [23], the authors propose a simple adaptation of the exemplar based inpainting of Criminisi et al. [5] where one of the views is inpainted first. For the other views, instead of searching a corresponding patch in the known regions of the view to inpaint, the patch is searched in the first inpainted view in order to ensure a better consistency across views. However, because of the greedy nature of the algorithm and the fact that the views are processed separately, the inpainting results may still contain severe angular inconsistencies.

Other methods have been proposed in the related topic of multiview image inpainting [24], [25]. These methods were designed for images with large differences between viewpoints. Experiments from the authors have shown that the generalization to densely sampled light fields with short baselines is not straightforward because their camera calibration step failed with such data. The complexity in [24] is also an issue considering the large number of views of a light field. Similarly, the method in [26] processes light fields with a sparse set of views and a large baseline, which makes it possible to recover occluded areas of an image from different viewpoints where the corresponding area is visible. However, this method hardly applies to densely sampled light fields because of its computational cost and the fact that very few disoccluded areas can be exploited.

Low rank matrix completion

The low rank matrix completion problem has received a great deal of attention in the past few years. For a matrix $M \in \mathbb{R}^{m \times n}$ of sufficiently low rank r $(r \ll \min(m, n))$ and

given only a subset of its entries, the goal is to recover the entire matrix by exploiting the low rank prior. The optimization problem is then:

$$\min_{X} \quad \text{rank}(X)$$
$$\text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M), \tag{1}$$

where Ω is the set of indices of the known elements of M, and $\mathcal{P}_\Omega$ is such that the element (i, j) of $\mathcal{P}_\Omega(X)$ is equal to $X_{i,j}$ if $(i, j) \in \Omega$ and zero otherwise.

Candes and Recht [27] have determined conditions under which the problem has a unique solution. Furthermore, although the rank is not a convex function, replacing it by the trace norm $\|X\|_*$ leads to a convex problem with the same unique solution, assuming the conditions in [27] are satisfied. An algorithm based on singular value soft thresholding has then been developed in [8] to solve this trace norm minimization problem. The authors in [10] accelerate the latter algorithm by using the alternating direction method of multipliers (ADMM) [28]. In [29], a similar algorithm, HaLRTC, have been used for filling holes in an image by considering the 2D image directly as a matrix to complete. Note that in this context of spatial 2D inpainting, the low rank prior is only satisfied for images containing essentially horizontal and vertical repetitions. Our application, though, is not concerned by this limitation since the 4D light field is inpainted in the angular dimensions given the complete central view (previously spatially inpainted with an arbitrary method).

Another important issue must be taken into consideration when performing inpainting using low rank completion algorithms. The theoretical results in [27] are based on the assumption that the known elements Ω of the matrix are selected uniformly at random. However, this is not the case in the context of object removal in an image or a light field. Therefore, replacing the rank by the trace norm in equation (1) does not necessarily yield optimal results. In a second type of approach, the true rank is considered. For a given target rank r, the r-rank matrix that best approximates the known entries of the input matrix is sought:

$$\min_{X} \quad \frac{1}{2} \|\mathcal{P}_\Omega(M - X)\|_F^2$$
$$\text{s.t.} \quad \text{rank}(X) \le r, \tag{2}$$

where $\|.\|_F$ is the Frobenius norm. Many algorithms have been developed to solve this problem such as ADMiRA [11], OptSpace [12], GoDec [13], NIHT [14]. They are based on singular value decomposition (SVD) and hard thresholding. In the weighted non negative matrix factorization (WNMF) [15] and in LMaFit [16], a similar rank constraint is enforced by expressing X as a product of two matrices $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$. These methods are however impractical for inpainting because they require an accurate rank estimate which is in itself difficult to obtain.

III. PROPOSED LOW RANK COMPLETION ALGORITHM

In order to solve the problem of Equation (1), we use the alternating direction method of multipliers (ADMM) [28], similarly to several low rank approximation algorithms [10],

[29]–[31]. Unlike those methods however, we directly want to solve the problem of rank minimization instead of its convex relaxation formed with the trace norm. Furthermore, in order to increase the robustness of the method in the case where the matrix is only approximately low rank, we propose to relax the equality constraint of Equation (1) by the inequality $\|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(M)\|_F^2 / \|\mathcal{P}_\Omega(M)\|_F^2 \le \epsilon$, given a positive value ϵ. The problem can be equivalently written:

$$\min_{X} \quad \text{rank}(X)$$
$$\text{s.t.} \quad X = Z \tag{3}$$
$$Z \in \mathcal{C},$$

with $\mathcal{C} = \left\{ Z \in \mathbb{R}^{m \times n} \mid \frac{\|\mathcal{P}_\Omega(Z) - \mathcal{P}_\Omega(M)\|_F^2}{\|\mathcal{P}_\Omega(M)\|_F^2} \le \epsilon \right\}$.

The constraint $X = Z$ can be removed by defining the following augmented Lagrangian function:

$$\mathcal{L}(X, Z, \Lambda, \rho) = \text{rank}(X) + \langle \Lambda, X - Z \rangle + \frac{\rho}{2} \|X - Z\|_F^2 \tag{4}$$

where Λ is a matrix of lagrangian multipliers, ρ is a positive scalar, and $\langle ., . \rangle$ is the scalar product (*i.e.* $\langle A, B \rangle = \text{Tr}(A^\top B)$). The ADMM method then alternatively solves a sub-problem for each of the matrices X, Z and Λ using the update rules given in Algorithm 1. The value of ρ is also increased at each iteration in order to accelerate the convergence.

Algorithm 1 Low rank completion with ADMM

Input: Ω, $\mathcal{P}_\Omega(M)$, ϵ, ρ, t.
1: Initialize : $Z = \mathcal{P}_\Omega(M)$, $\Lambda = 0$,
2: **repeat**
3:      $X \leftarrow \arg\min_{X} \mathcal{L}(X, Z, \Lambda, \rho)$ (see Equation 5)
4:      $Z \leftarrow \arg\min_{Z \in \mathcal{C}} \mathcal{L}(X, Z, \Lambda, \rho)$ (see Equation 6)
5:      $\Lambda \leftarrow \Lambda + \rho(X - Z)$
6:      $\rho \leftarrow t \cdot \rho$ (with $t > 1$)
7: **until** $\|\mathcal{P}_\Omega(Z) - \mathcal{P}_\Omega(M)\|_F^2 / \|\mathcal{P}_\Omega(M)\|_F^2 \le \epsilon$
Output: X

In subsections III-A and III-B, we will demonstrate that the X and Z sub-problems, (lines 3 and 4 of Algorithm 1 respectively), have the following closed form solutions:

$$\arg\min_{X} \mathcal{L}(X, Z, \Lambda, \rho) = \mathbb{H}_{\sqrt{\frac{2}{\rho}}} \left(Z - \frac{\Lambda}{\rho} \right), \tag{5}$$

where $\mathbb{H}_s$ is the singular value hard thresholding operator defined by, $\mathbb{H}_s(A) = U \text{diag}(\{\mathcal{H}_s(\sigma_i)\}) V^\top$, for a matrix A of singular vectors U and V, and singular values $\{\sigma_i\}$, and where $\mathcal{H}_s(\sigma_i)$ is equal to σ_i if $\sigma_i \ge s$, and 0 otherwise.

The solution of the Z sub-problem is given by:

$$\mathcal{P}_\Omega(Z) = P_\Omega \left((N + \lambda M)/(1 + \lambda) \right), \tag{6a}$$
$$\mathcal{P}_{\overline{\Omega}}(Z) = P_{\overline{\Omega}}(N), \tag{6b}$$

with $N = X + \frac{\Lambda}{\rho}$, and $\lambda = \max\left(\frac{\|P_\Omega(M - N)\|_F}{\sqrt{\epsilon} \cdot \|\mathcal{P}_\Omega(M)\|_F} - 1, 0 \right)$.

We can note from equation (5), that the singular values threshold depends on ρ. As ρ increases with iterations, the threshold decreases. Therefore the rank is likely to increase.

(a) incomplete image   (b) trace norm minimized ($\epsilon = 0$)

(c) rank minimized ($\epsilon = 0$)   (d) rank minimized ($\epsilon = 0.03$)
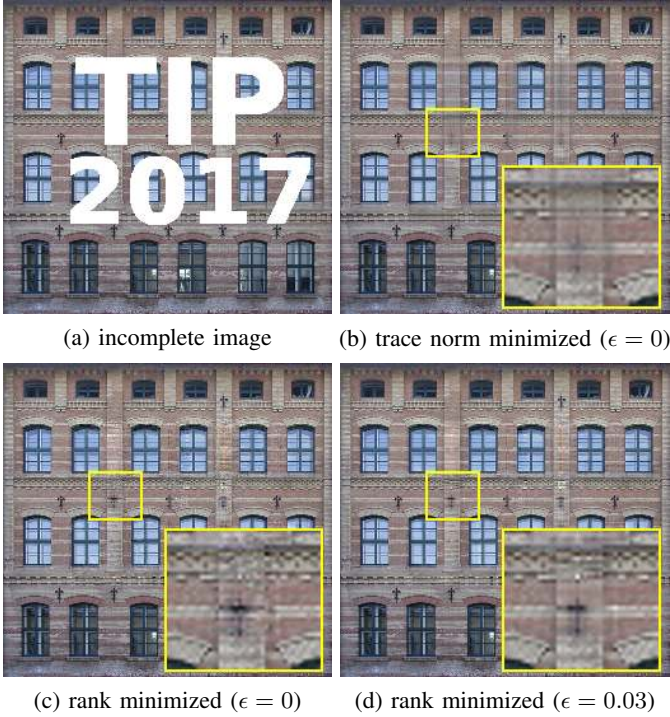
Fig. 2. Example of low rank matrix completion. For this test, each column of the matrix to complete contains the R,G and B values of the pixels of one column of the image. The rank minimization results in (c) and (d) are sharper than in (b) where the trace norm is minimized. For (d), ϵ has a non-zero value resulting in less noisy completion. The simulations (b), (c), (d) give respective ranks of 233, 119, and 54 after respectively 16, 26 and 49 iterations.

For that reason we initialize ρ as $\rho = 8/(\sigma_1 + \sigma_2)^2$ (with σ_1 and σ_2 the two largest singular values of the initial Z) so that the threshold $\sqrt{(2/\rho)}$ at the first iteration is between σ_1 and σ_2. This way the algorithm starts with a rank-1 estimate of X and increases the rank progressively. Although the general ADMM method was designed for convex optimization, convergence is observed in practice in our non-convex rank minimization thanks to the update of ρ. From our experiments, satisfactory results were obtained by multiplying ρ by the parameter $t = 1.45$ at each iteration. A trade-off between convergence speed and accuracy can be obtained by increasing this value.

Note that replacing the rank by the trace norm in the problem formulation (3) would lead to a very similar expression of the solution where singular value soft thresholding is used instead of hard thresholding (*e.g.* [8], [29]) in the X subproblem. The value of the threshold is also different ($1/\rho$ for trace norm minimization instead of $\sqrt{(2/\rho)}$ in our algorithm). Therefore, in order to compare both approaches, the initialization of ρ was adapted in our implementation of the trace norm minimization as $\rho = 2/(\sigma_1 + \sigma_2)$, so that the threshold at the first iteration is the same as the one used in our method. Figure 2 shows the results obtained using either rank or trace norm minimization for matrix completion in a simple 2D inpainting example. Sharper details are recovered in Figure 2(c) and (d) (rank minimization), than in Figure 2(b) (trace norm minimization). Note, for example, how the repetition of the black crosses is preserved in our method while these detail have almost vanished in the inpainted area

with trace norm minimization. In addition, we can observe in Figure 2(d) that the introduction of the tolerance parameter ϵ successfully removes noise in the inpainted area. Intuitively, the sharper results obtained with the rank minimization can be understood by the fact that the hard thresholding only removes the lowest singular values which are associated to the singular vectors containing the highest frequencies. But it does not affect the singular vectors with medium frequencies associated to singular values just above the threshold. Soft thresholding also attenuates those medium frequencies as it reduces all the singular values.

A. X sub-problem

One can verify that the X sub-problem is equivalent to determining X that minimizes the following expressions:

$$\min_X \quad \mathrm{rank}(X) + \frac{\rho}{2}\left\| X - Z + \frac{\Lambda}{\rho} \right\|_F^2 \tag{7a}$$

$$= \min_r \left(r + \min_{X \text{ s.t. } \mathrm{rank}(X) \le r} \frac{\rho}{2}\left\| X - Z + \frac{\Lambda}{\rho} \right\|_F^2 \right). \tag{7b}$$

Let us denote by $U\Sigma V^\top$ the SVD of $Z - \frac{\Lambda}{\rho}$ (where Σ is a diagonal matrix with non-increasing diagonal entries). We define $\sigma_i := \Sigma(i,i)$ and $\Sigma_r \in \mathbb{R}^{m \times n}$ by:

$$\Sigma_r(i,j) := \begin{cases} \sigma_i & \text{if } i = j, \ i \le r \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

From the Eckart-Young theorem, we know that given r, Equation (7b) is minimized for $X = U\Sigma_r V^\top$. We can then rewrite Equation (7b) as follows:

$$\min_r \left(r + \frac{\rho}{2}\left\| U\Sigma_r V^\top - U\Sigma V^\top \right\|_F^2 \right) \tag{9a}$$

$$= \min_r \left(\frac{\rho}{2}\left(\frac{2}{\rho} \cdot r + \left\|\Sigma_r - \Sigma\right\|_F^2 \right) \right) \tag{9b}$$

$$= \min_r \left(\frac{\rho}{2}\left(\sum_{i=1}^{r} \frac{2}{\rho} + \sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \right) \right). \tag{9c}$$

This expression is minimized by choosing r such that $\sigma_i \ge \sqrt{\frac{2}{\rho}}$ if and only if $i \le r$. Therefore, the solution of the X sub-problem is $X = U\Sigma_r V^\top = \mathbb{H}_{\sqrt{\frac{2}{\rho}}}(Z - \frac{\Lambda}{\rho})$. $\square$

B. Z sub-problem

Similarly to subsection III-A, we can express the Z sub-problem as:

$$\min_Z \|N - Z\|_F^2 \\ \text{s.t. } \|\mathcal{P}_\Omega(Z) - \mathcal{P}_\Omega(M)\|_F^2 / \|\mathcal{P}_\Omega(M)\|_F^2 \le \epsilon \tag{10}$$

where $N = X + \frac{\Lambda}{\rho}$. The problems for $\mathcal{P}_\Omega(Z)$ and $\mathcal{P}_{\overline{\Omega}}(Z)$ can be solved independently. For $\mathcal{P}_{\overline{\Omega}}(Z)$, the constraint does not apply. Then, the solution is $\mathcal{P}_{\overline{\Omega}}(Z) = \mathcal{P}_{\overline{\Omega}}(N)$.

For $\mathcal{P}_\Omega(Z)$, let us define the Lagrangian function:

$$\mathcal{L}_z(\mathbf{z}, \lambda) = \|\mathbf{n} - \mathbf{z}\|_F^2 + \lambda\left(\|\mathbf{m} - \mathbf{z}\|_2^2 - \epsilon \cdot \|\mathbf{m}\|_2^2 \right). \tag{11}$$

For convenience, we define vectors $\mathbf{z}$, $\mathbf{n}$, and $\mathbf{m}$ as the vectorized entries within Ω of respectively Z, N, and M.

Since the Frobenious norm is convex, the solution only needs to satisfy the first order Kuhn-Tucker conditions:

- Stationary condition (*i.e.* $\frac{\partial \mathcal{L}_z}{\partial \mathbf{z}}(\mathbf{z}) = 0$):

$$\frac{\partial \mathcal{L}_z}{\partial \mathbf{z}}(\mathbf{z}) = 0 \Leftrightarrow 2(\mathbf{z} - \mathbf{n}) + 2\lambda(\mathbf{z} - \mathbf{m}) = 0, \quad (12a)$$

$$\Leftrightarrow \mathbf{z} = \frac{\mathbf{n} + \lambda\mathbf{m}}{1 + \lambda}. \quad (12b)$$

- Complementary slackness:

$$\begin{cases} \lambda \geq 0 \\ \lambda = 0 \text{ or } \|\mathbf{m} - \mathbf{z}\|_2^2 - \epsilon \cdot \|\mathbf{m}\|_2^2 = 0. \end{cases} \quad (13)$$

Using Equation (12b), $\mathbf{z}$ can be replaced by its expression. We obtain: $\|\mathbf{m} - \mathbf{z}\|_2^2 = \|\mathbf{m} - \mathbf{n}\|_2^2/(1 + \lambda)^2$. The condition (13) thus becomes:

$$\lambda = \max\left(\frac{\|\mathbf{m} - \mathbf{n}\|_2}{\sqrt{\epsilon} \cdot \|\mathbf{m}\|_2} - 1, 0\right). \quad (14)$$

This concludes the proof of equation (6).

IV. APPLICATION TO LIGHT FIELD INPAINTING

In this section, we show how the inpainting of one light field view (*e.g.* the central view) can be consistently propagated to the rest of the light field by means of low rank matrix completion. Figure 3 illustrates how the light field data is organized in a matrix that can be completed by our algorithm. The method is based on the premise that the views are highly correlated. As a consequence, the matrix formed by vectorizing each view and by concatenating the resulting column vectors, can be assumed to have a low rank (with respect to the number of views). Here, a row of the matrix contains the pixels' values in all the views at a fixed (x,y) coordinate. Unfortunately, when the area to be removed has roughly the same position in all the views, many rows of the matrix only contain one known entry, corresponding to the central view. In this configuration, low rank completion is not able to recover reliably the unknown entries. Thus, in a preliminary step, we generate several additional views by warping the inpainted central view with a set of homography projections, as shown in Figure 3. Instead of computing the homographies that globally compensate for the disparities between the central view and the other views, we prefer to generate a set of random homographies. This approach allows a more uniform sampling of all the possible displacements of each region of the image.

Note that the low rank property is relative to the matrix size. Therefore, a light field with large amount of parallax may still satisfy this property as long as it contains a sufficient number of views, including the additional warped views. However, for practical reasons, this application essentially targets light fields with a limited baseline such as those captured by plenoptic cameras, for which a dense view sampling can be achieved with a limited number of views.
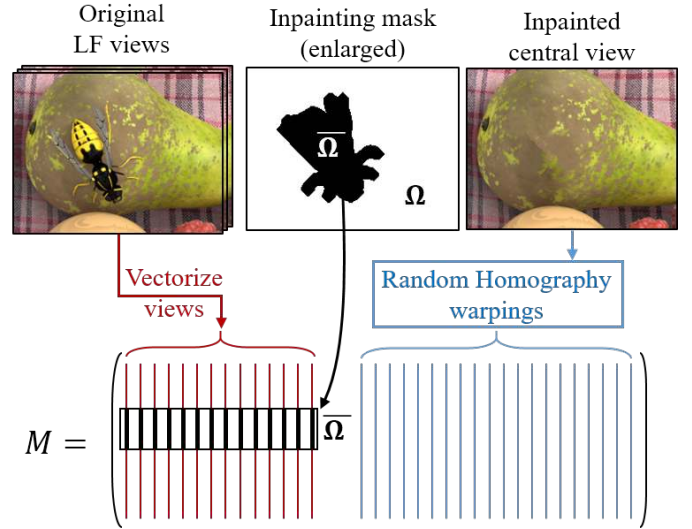

Fig. 3. Construction of the matrix to complete.

A. Random generation of homographies

A homography transformation is defined by a 3×3 matrix H, transforming 2D coordinates (x, y) into (x', y') as:

$$w \cdot [x', y', 1]^\top = H \cdot [x, y, 1]^\top. \quad (15)$$

Note that the element $(3, 3)$ of H can be fixed to 1 without loss of generality. Therefore, eight parameters are needed to determine one homography. In our method, we need to carefully generate those values so that the resulting homographies are within a reasonable range of rotation angle, translation, shear *etc*. In order to determine a suitable probability distribution function (pdf) for the eight parameters, we have built a dataset of homographies from a set of 62 light field images captured with a Lytro camera. For all the images, we have matched one homography between the central view and each of the other views.

Note that the homography matrix depends on the image size. For instance, applying the same homography transformation to two images of different definition will result in a larger translation for the lower definition image relatively to its size. In order to remove the influence of the image size, one must determine the relation between a homography matrix H transforming any point (x, y) into (x', y') and the corresponding scaled matrix $H^{a,b}$ transforming (ax, by) into (ax', by'), given scaling values $a, b > 0$. By noting $h_{i,j}$ the element (i, j) of H, one can find that $H^{a,b}$ is expressed as (see Appendix):

$$H^{a,b} = \begin{pmatrix} h_{1,1} & h_{1,2}\frac{b}{a} & \frac{h_{1,3}}{a} \\ h_{2,1}\frac{a}{b} & h_{2,2} & \frac{h_{2,3}}{b} \\ h_{3,1} \cdot a & h_{3,2} \cdot b & 1 \end{pmatrix}. \quad (16)$$

For building our dataset, each homography H was then replaced by its scaled version $H^{a,b}$ where a and b are the inverse of the image width and height respectively. This simulates the homography one would obtain for images of size 1×1.

Figure 4 shows the marginal distributions of all the homography parameters for our dataset. We observe that the eight distributions have a narrow peak around the mean that is
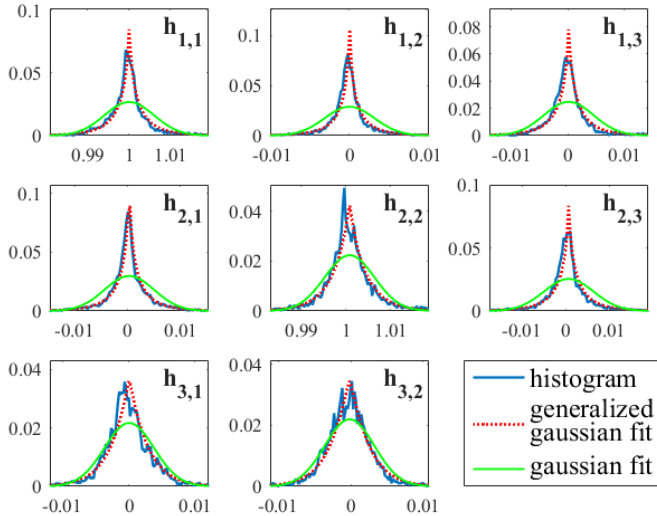
Fig. 4. Marginal distributions of the eight homography parameters (each corresponding to an entry of the homography matrix).
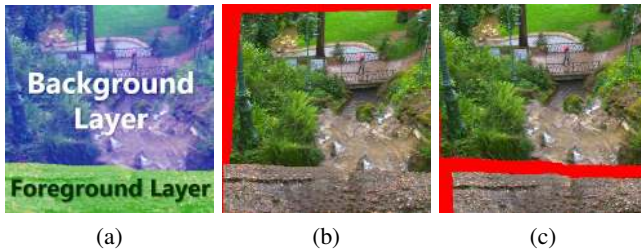

Fig. 5. (a) Input inpainted central view with layer segmentation. (b) and (c) are two examples of random warpings. Red areas correspond to pixels with no color information after the warping. In (b), the foreground layer partly occludes the background.

better modeled by a generalized gaussian distribution (GGD) than a simple gaussian distribution. In order to take into account the dependencies between the parameters, we fit a multivariate GGD (MGGD) to the data using the method in [32]. For each homography to generate, a 8×1 vector is first generated randomly with our MGGD distribution as described in [33]. The vector is re-arranged into a homography matrix H. Finally, in order to warp an image, the scaled version $H^{a,b}$ of the matrix must be used, where a and b are the image width and height respectively.

For our experiments, we considered light field images with a limited baseline (e.g. captured with a plenoptic Lytro camera). In this situation, generating 400 warpings of the central view using this procedure was found to give a sufficiently dense sampling of the homographies for our inpainting application.

B. Extension to multiple depth layers

In the case where the area to recover contains several layers with different depths, the global homography warping is not sufficient to represent the depth discontinuity. The method has then been extended to better handle this case. For that purpose, a segmentation map of the different depth layers of the inpainted central view must be provided additionally, and the depth order of the segments must be known.

This information is used to warp the layers with different homographies (each homography being chosen randomly as
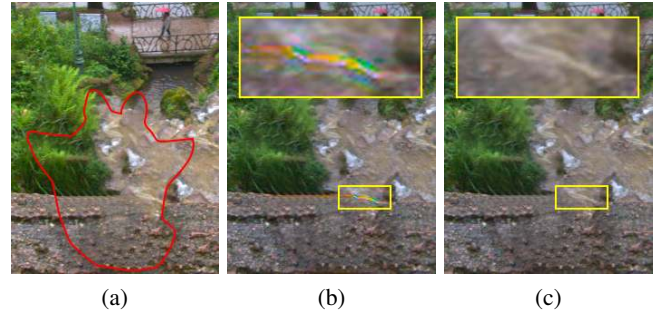

Fig. 6. (a) Central view inpainted with [6] within the red boundary. (b) Inpainting propagated to another view without filling the invalid regions of the warped views (they are set to zero). (c) Result when the invalid regions are set into $\overline{\Omega}$ (*i.e.* filled by low rank matrix completion).

described in the previous subsection). The warped regions corresponding to each layer are combined in one image. In case of overlap between several warped layers, the layer labelled with the lowest depth should occlude the other ones. Several images are generated randomly with this method to augment the matrix to complete as shown in Figure 3. Two examples of warpings obtained with this method are shown in Figure 5(b) and (c).

Note that depth cannot be directly estimated in the area to be inpainted since only the central view is fully available. Therefore, a certain level of user interaction is required in order to segment the different depth layers. State-of-the art segmentation tools with user interaction such as grab-cut [34] can be used for that purpose. In this method, a user can paint strokes associated to different labels to coarsely define the different regions. The labels are then propagated to the rest of the image to generate the segmentation map. Since this segmentation is only needed for the central view, the user input required for this step is limited.

C. Determination of the unknown pixels

Once the matrix M has been defined, the set of unknown entries $\overline{\Omega}$ must be determined in order to perform matrix completion. $\overline{\Omega}$ essentially contains the entries corresponding to the area to be removed in the original light field views, as shown in Figure 3. For simplicity, the area to be removed in each view is defined by enlarging the mask that was used for inpainting the central view. For all the light field images considered in our experiments (*i.e.* essentially plenoptic capture), enlarging the mask by 10 pixels was sufficient to completely cover the object to be removed in each view.

In addition to the original views, the warped versions of the central views may also contain unknown pixels. When warping the central view, some pixels at the border of the warped image are undetermined if their projection in the central view is outside of the image area (*e.g.* red areas in Figure 5(b) and (c)). In the case where several depth layers are considered, a disocclusion area between the warped layers may also be unknown as shown in red in Figure 5(c). All those unknown regions should be included in $\overline{\Omega}$ so that they are filled jointly with the original views by the low rank completion algorithm. Figure 6 shows the benefit of filling the borders of the warped views by matrix completion.
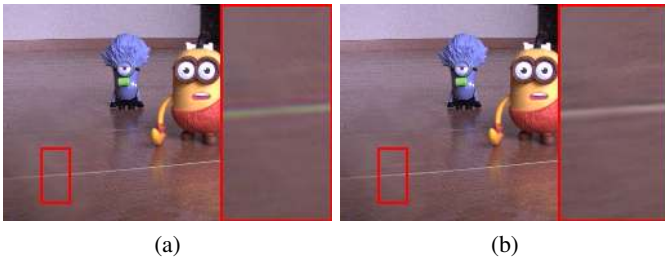
(a)   (b)

Fig. 7. Inpainting results for one view of the light field 'Figurine' using (a) RGB colorspace (b) Yu'v' colorspace. The zoomed detail is included in the inpainted area.

D. Color Processing

In our method, the color components are treated separately. This is more computationally and memory efficient than building one matrix containing all the color components. However, instead of processing the light field directly in the RGB colorspace, we compute the luminance channel Y, and the two chromaticity components CIE u'v'. The choice of this Yu'v' colorspace is due to its perceptual uniformity. Since the three components are treated separately, there may be small difference between the completed regions of each component (e.g. slightly shifted textures). If the RGB colorspace was used, these differences would be perceived more easily than in the Yu'v' colorspace which decouples the luminance from the chromaticity components that are less important perceptually.

Additionally, since the human eye is less sensitive to chromaticity than luminance, the memory and computation time can be further reduced by using less homography warpings for the u' and v' components (100 in our experiments, instead of 400) without significantly degrading the perceived result. Similarly, for the chromaticity components, we use a higher value for the parameter t controlling the convergence speed ($t = 5$ for u' and v' instead of $t = 1.45$), which significantly reduces the number of iterations without noticeable difference. Figure 7 shows an example of inpainting results obtained using either the Yu'v' or RGB colorspaces. Some color artifacts can be seen around the white line when the images are processed in RGB (Figure 7(a)). Note that the matrix completion algorithm was also faster with Yu'v' (19 seconds instead of 45 with RGB) because of the reduced computation time for the u' and v' components.

E. Limitations

The main limitation of our method is the memory requirement, since a large set of views (400 for the luminance) are added to the original light field views to form the matrix M. In order to alleviate this problem, in our implementation, only the pixels within an area surrounding the object to be removed are used to build the matrix M. The number of lines of M is then the number of pixels in this area.

Our method has been designed and validated for light fields captured by plenoptic cameras or synthetic light fields with a limited baseline. Other types of capture devices such as a camera array or a camera mounted on a gantry can produce light fields with larger baselines. Such data would require a different probability distribution for the homography sampling, and a larger number of warped views.

TABLE I
RUNNING TIMES AND NUMBER OF ITERATIONS OF ALGORITHM 1 FOR THE TESTED LIGHT FIELDS. THE NUMBER OF ITERATION IS GIVEN FOR EACH COLOR COMPONENT YU'V'. THE RUNNING TIME IS THE SUM ON THE THREE COMPONENTS.

Light field (#views)	Central view inpainting	Running Time	#iterations		
			Y	u'	v'
butterfly (9x9)	manual	45.0 s	30	8	8
stillife (9x9)	Turkan et al. [7]	13.2 s	32	9	9
TotoroWaterfall (7x7)	Daisy et al. [6]	9.1 s	30	8	9
TotoroAlley (7x7)	Daisy et al. [6]	9.2 s	34	8	10
Bee2 (11x11)	manual	4.2 s	29	7	9
Figurines (11x11)	manual	19.3 s	34	9	9

V. EXPERIMENTAL RESULTS

Our method has been evaluated on both synthetic and real world light field images presented in Table I. For some of the tested light fields, manual inpainting of the central view was necessary. This task was performed with the open source GIMP software [35], essentially using the cloning tool.

The synthetic light fields *'butterfly'* and *'stillife'* are taken from the HCI database [36]. The other ones were captured with a Lytro camera and the sub-aperture images were extracted with the toolbox of Dansereau *et al.* [37]. This produces challenging data with noise and color and illumination variations between views. The images shown in subsections V-A, V-B, and V-C do not present depth discontinuity in the area to inpaint. Hence, no depth layer segmentation was used for these experiments. We show in subsection V-D how our method is improved by the use of a depth layer segmentation for light fields with a depth discontinuity.

For all the experiments, the parameters of the Algorithm 1 are fixed as: $\epsilon = 0.003$, $t = 1.45$ for the luminance and $t = 5$ for the chromaticity components. The parameter ρ is initialized automatically as explained in section III.

The experiments were performed with our matlab implementation using a PC with an Intel Core i7-7700 CPU with 4 cores (8 threads) at 3.6GHz and 32GB RAM. The number of iterations and running times of Algorithm 1 are given in Table I for each tested light field.

A. Rank vs trace norm minimization

First, in order to confirm the interest of minimizing the rank instead of the trace norm (*e.g.* [10], [29]), a comparison of both approaches is shown in Figure 8. In our implementation of the trace norm based matrix completion [10], the same values were used for the parameters ϵ and t. The only differences with our rank minimization are the thresholding step and the initialization of the parameter ρ as explained in section III. This example illustrates that our low rank matrix completion better preserves the details when propagating the inpainted area. This observation is consistent with the 2D inpainting result of Figure 2 which is sharper with our algorithm based on rank minimization. Note that in the case of the trace norm minimization, the effect of the noise tolerance parameter ϵ is very limited. Taking $\epsilon = 0$ does not produce sharper results.

(a) Central view with mask   (b) Central view inpainted by [7]

(c) Propagation to view (8,8) with trace norm minimization   (d) Propagation to view (8,8) with our rank minimization
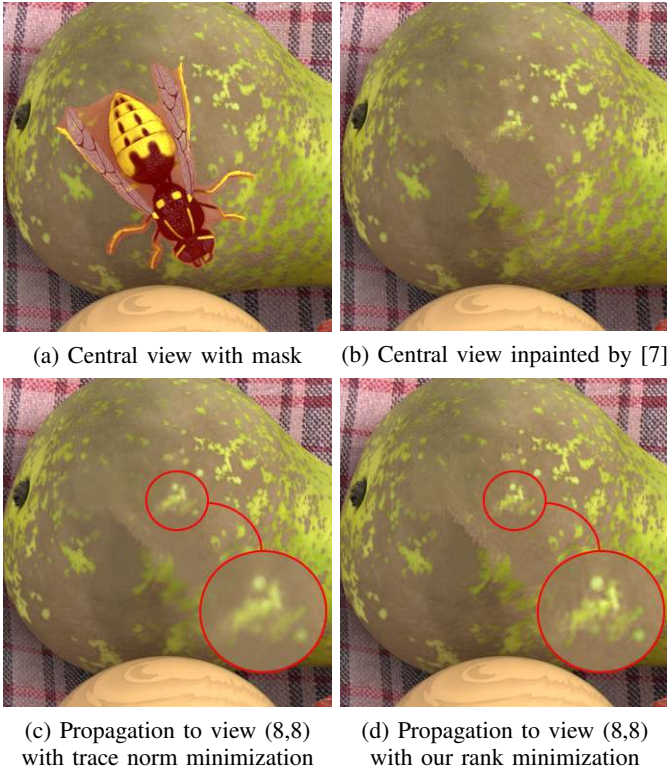
Fig. 8. Light field consistent inpainting of the synthetic lightfield "stilllife" with low rank matrix completion algorithm using either rank or trace norm minimization. For (c), our implementation of the trace norm based matrix completion [10] was used. More results can be found in the supplementary materials.

B. Comparison with patch-based methods

Additionally, we have implemented the method described in [23] that propagates an inpainted reference view to the other views of the light field. Each view is inpainted independently using a modified version of the patch based algorithm of Criminisi *et al.* [5] that searches patches from the inpainted reference view. Note that in their method, an extended focus image is used as a reference in place of the central sub-aperture image. However, for the comparison, we have used the central view as a reference in both methods. Figure 9 shows that the inpainting in [23] may yield inconsistent results. This is caused by the greedy procedure which fills patches of the unknown region one by one. Moreover, since the patches are selected from the central view, it is not robust to color and light intensity differences between the views.
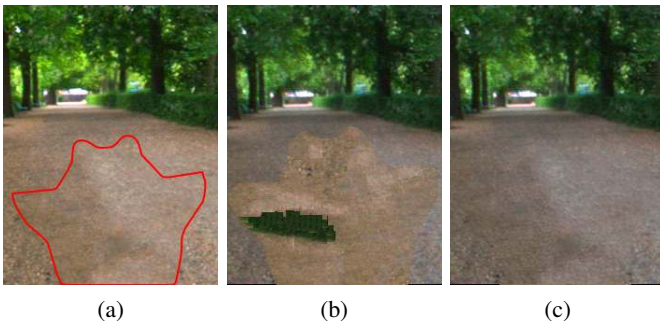

(a)   (b)   (c)

Fig. 9. (a) Central view inpainted with [6] within the red boundary. (b) Propagation to view (8,8) with the method described in [23]. (c) Propagation to view (8,8) with our method.

Chen et al. [18] and Zhang et al. [20] also proposed more advanced patch-based methods taking advantage of the 4D structure of the light field. Unlike our method, they cannot be used to propagate a view inpainted with an arbitrary technique to the rest of the light field. Although in PlenoPatch [20], the central view is inpainted first and then propagated to the other views, the propagation requires additional information retrieved from the initial 2D inpainting step. This information (i.e. patch offsets) is specific to the patch-based 2D inpainting method used in their implementation. Therefore, the method may not apply to complex cases such as the light field "butterfly" in Figure 11 for which a patch-based inpainting of the central view is unlikely to provide satisfactory results.

Nevertheless, a comparison between these patch-based methods and our approach can still be led by evaluating the consistency between the inpainted views. Section III of the supplementary materials shows that inconsistencies appear in the results of [18] and [23] (e.g flickering of texture details between views in [18], texture not following the geometry of the object in [23]), while our method does not produce such artifacts.

Visual comparisons with PlenoPatch [20] could not be performed. However, it should be noted that the depth layer estimation step used in this algorithm was not designed to cope with the amount of noise and color variations between views in the data extracted from plenoptic captures with [37]. Furthermore, regardless of the accuracy of the depth layers as well as the quality of the initial 2D inpainting of the central view, the authors of [20] have also reported that their depth layer based patch synthesis does not accurately model geometry with slanted surfaces. This is not the case of our method thanks to the use of homography warpings (e.g. light field "figurine" in figure 10).

C. Comparison with direct homography projection

Additional results of our method are presented in Figure 10. It also shows the results obtained when the inpainted region of the central view is directly projected onto the other views with an homography. For this experiment the homographies were determined by matching SIFT descriptors [38] between the central view and each of the other views, and outliers were eliminated with RANSAC [39].

In the light field "figurines" (first example in Figure 10), the input top left view has severe distortion (noise, ghosting effect) caused by the capture and inaccuracies in the decoding process. This issue makes it difficult to find accurate correspondences between the central view and the other ones. It explains the discontinuities observed at the boundary of the inpainted region propagated with direct projection. These inaccuracies consequently impair the images generated by refocusing the inpainted light field. For example, in the light field "figurine" refocused on the foreground, the inpainted area appears blurred in the foreground with the direct projection method. In contrast, our method can cope with this type of data and ensures a consistent inpainting propagation to all the views. As a result, the refocused image has a realistic depth of field, where only the out-of-focus areas appear blurred. Note

however that the inpainted area may still have a different appearance (less noise) than the rest of the image for the views that are more noisy than the central inpainted view. This is visible in the inpainted top left view shown in Figure 10 because the outer views of the light fields captured with plenoptic cameras are more noisy than the central views. However, this effect is no longer visible after the refocusing step which reduces the noise by averaging all the views.

Another type of distortion introduced by plenoptic capture can be observed in "Bee2" and "TotoroAlley" (second and third examples of Figure 10) : a global variation of color and light intensity appears between the central and the outer views. While our low rank matrix completion approach is able to adjust the overall color and brightness of the inpainted area to each view, the direct projection causes a color discontinuity. Note however that this artifact is less perceptible after refocusing the light field, since only a few of the views are concerned by this issue.

D. Extension with depth layer segmentation

Finally, Figure 11 shows the results obtained for two light fields containing a depth discontinuity, either using the simple version of our method (without segmentation) or with the extended version which takes a segmentation map of depth layers as input. In both cases, the results are improved by the use of the segmentation. It is particularly visible in the refocused image where we can see a clearer separation between the in-focus foreground and the out-of-focus background.

E. Convergence and complexity

Because of the non-convexity of the rank minimization problem, the use of the ADMM method does not provide theoretical guarantees of convergence. Studying the theoretical convergence is particularly challenging for this problem and is out of the scope of this paper. However, in our experiments, convergence was obtained in approximately 30 iterations for the luminance and less than 10 for the chromaticity which requires less accuracy (see Table I).

The running time also depends on the size of the matrix to complete. For a $m \times n$ matrix, the complexity of the SVD is $O(\min(n^2 m, nm^2))$, which is the complexity bottleneck for each iteration of our algorithm. In the light field inpainting application, the number of columns n is equal to the number of views including the additional warped views. The number of rows m is equal to the number of pixels within an area surrounding the inpainting mask, which is generally much higher than n. The complexity is then $O(n^2 m)$ per iteration.

Therefore, the complexity depends on the square of the number of added warped views. For that reason and for the memory limitations, this parameter should not be taken higher than necessary. An example showing the effect of the number of warped views on the inpainting quality is shown in the supplementary materials. Increasing this parameter improves the quality of the inpainted area (e.g. reduce blur and ghosting artifacts), but the results tend to stabilize for sufficiently high numbers of added warped images. From our experiments, we have considered that setting this parameter to 400 for the luminance component and 100 for the chrominance was a reasonable tradeoff between the result quality and the computational load since higher values did not produce significantly better results in most cases.

In order to reduce unnecessary memory consumption and computing time, single precision arithmetic was used in the implementation. However, further speed optimizations could be considered. In practice, less than a third of the matrix completion running time is spent on the singular value thresholding step that includes the most complex operations of the algorithm (i.e. SVD computation and matrix multiplications). More than half of the time is used for internal matlab computations which may include unnecessary duplications of large matrices in memory. Therefore, significant gains may still be obtained. Furthermore, our algorithm may also benefit from accelerations with parallel computing since there exist parallel algorithms for the singular value decomposition of large matrices. For example, significant speed improvements were reported by the authors of [40] for their SVD implementation on GPU compared to that of matlab.

VI. Conclusion

In this paper we proposed a novel light field inpainting algorithm where the central view is inpainted first using a classical automatic or semi-supervised 2D image inpainting method. Our focus was on the consistent propagation of the inpainted area to the rest of the light field. For that purpose we first generate a set of warped versions of the inpainted central view with random homographies, using a carefully chosen random distribution. If the area to inpaint contains depth discontinuities, a segmentation map of the depth layers of the inpainted central view may be given as additional input. It is used by our algorithm to warp the depth layers with different homographies. Then, a matrix is formed by vectorizing and concatenating the warped images and the original light field views. Thanks to the high correlation between views, the matrix can be assumed to have a low rank, which enables the use of a low rank matrix completion algorithm.

The current matrix completion methods have been designed essentially for the case where the positions of the unknown entries are uniformly distributed in the matrix. Therefore, we also proposed a new matrix completion algorithm that is better suited to the inpainting application where a large region is unknown instead of random entries. We have demonstrated the effectiveness of our matrix completion scheme for inpainting all the views of a light field. In particular, the method is well suited for real world light fields captured by plenoptic cameras, that have a short baseline but present challenging features such as noise and color and illumination variations between the views. Furthermore, thanks to the consistency of the inpainting across views, the resulting light field can be used to produce refocused images with realistic depth of field, which is an important application of plenoptic photography.

APPENDIX
Proof of the Homography scaling Equation (16)

Given a homography matrix H transforming a 2D point (x, y) into (x', y'), we want to find the matrix $H^{a,b}$ that
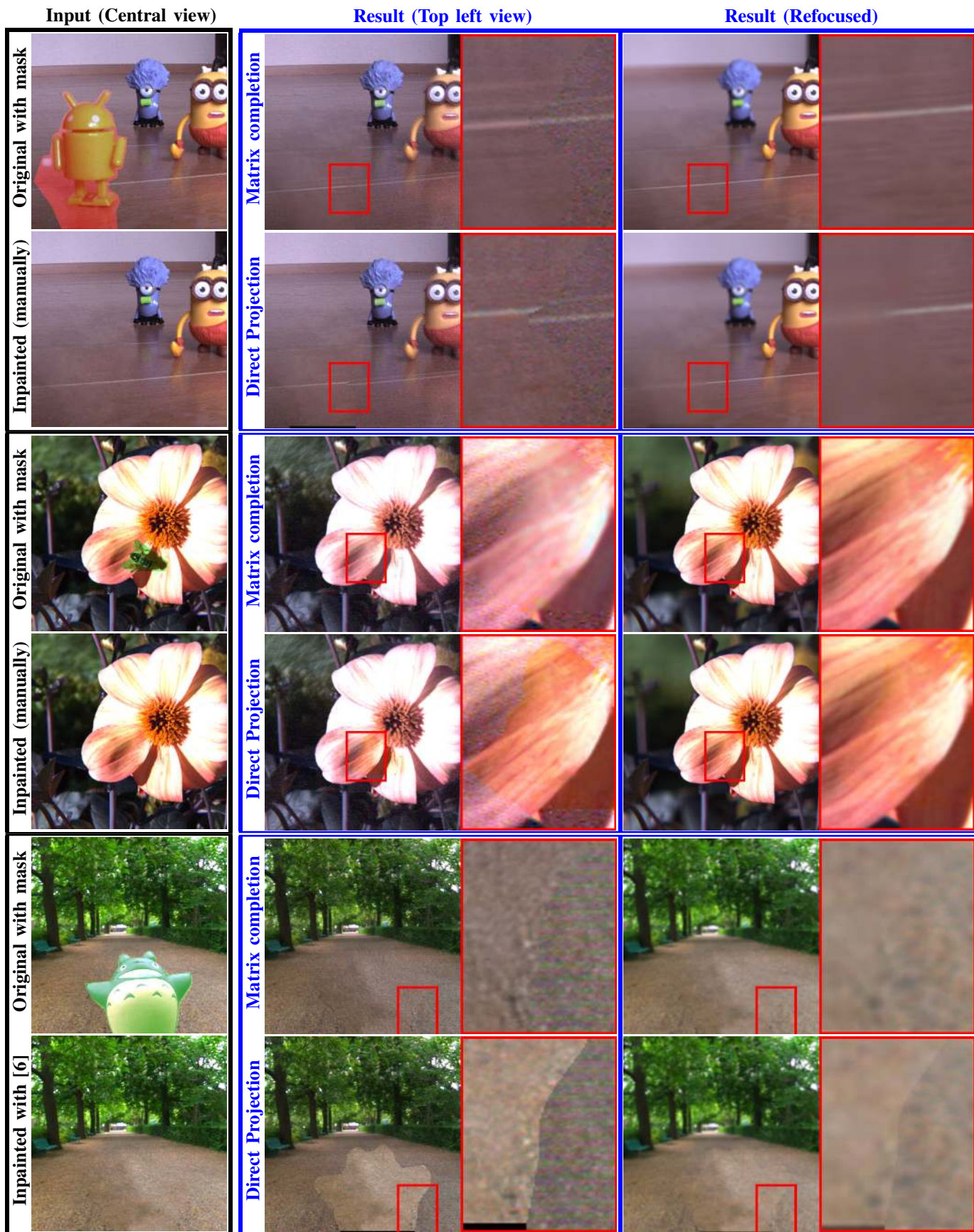
Fig. 10. Results for the three light fields "figurine", "Bee2" and "TotoroAlley" (from top to bottom) captured with a Lytro Illum camera. Our method based on matrix completion is compared to the direct projection of the inpainted region using homographies computed by matching SIFT points between the central view and the other views. The last column shows an example of refocused image generated by shifting and averaging all the views. For 'figurine" and "Bee2", the inpainting of the central view was performed manually since traditional 2D inpainting methods did not produce satisfactory results. More results including videos displaying all the views of the light fields can be found in the supplementary materials and on our web page: https://www.irisa.fr/temics/demos/lightField/InpaintMC/LFinpaintMC.html

Fig. 11. Results of our method for the light fields "butterfly" (synthetic) and "TotoroWaterfall" (captured with a Lytro Illum camera). Those light fields contain a depth discontinuity, and the result are shown either with or without using a segmentation of the depth layers as input (segmentation map shown on top of the inpainted central view). The last column shows an example of refocused image generated by shifting and averaging all the views. For 'butterfly', the inpainting of the central view was performed manually since traditional 2D inpainting methods did not produce satisfactory results. More results including videos displaying all the views of the light fields can be found in the supplementary materials and on our web page: https://www.irisa.fr/temics/demos/lightField/InpaintMC/LFinpaintMC.html

transforms (ax, by) into (ax', by') for arbitrary scaling values $a, b > 0$, and any point (x, y).

For a given a pair of coordinates (x, y), its projection (x', y') with the homography defined by $H = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix}$, is expressed as:

$$w \cdot [x', y', 1] = [x, y, 1] \cdot H \tag{17}$$

Therefore,

$$\begin{cases} wx' = h_{1,1} \cdot x + h_{2,1} \cdot y + h_{3,1} & \text{(18a)} \\ wy' = h_{1,2} \cdot x + h_{2,2} \cdot y + h_{3,2} & \text{(18b)} \\ w = h_{1,3} \cdot x + h_{2,3} \cdot y + 1 & \text{(18c)} \end{cases}$$

Then, by replacing w by its expression in Equations (18a) and (18b), we obtain:

$$\begin{cases} h_{1,3} \cdot xx' + h_{2,3} \cdot yx' + x' \\ \qquad = h_{1,1} \cdot x + h_{2,1} \cdot y + h_{3,1} & \text{(19a)} \\ h_{1,3} \cdot xy' + h_{2,3} \cdot yy' + y' \\ \qquad = h_{1,2} \cdot x + h_{2,2} \cdot y + h_{3,2} & \text{(19b)} \end{cases}$$

Similarly, $H^{a,b}$ being defined as the homography matrix

projecting (ax, by) to (ax', by'), we obtain:

$$\begin{cases} h_{1,3}^{a,b} \cdot a^2 xx' + h_{2,3}^{a,b} \cdot abyx' + ax' \\ \qquad = h_{1,1}^{a,b} \cdot ax + h_{2,1}^{a,b} \cdot by + h_{3,1}^{a,b} \quad (20a) \\ h_{1,3}^{a,b} \cdot abxy' + h_{2,3}^{a,b} \cdot b^2 yy' + by' \\ \qquad = h_{1,2}^{a,b} \cdot ax + h_{2,2}^{a,b} \cdot by + h_{3,2}^{a,b} \quad (20b) \end{cases}$$

where the element (i, j) of $H^{a,b}$ is noted $h_{i,j}^{a,b}$. By dividing Equation (20a) by a and Equation (20b) by b, we obtain:

$$\begin{cases} a h_{1,3}^{a,b} \cdot xx' + b h_{2,3}^{a,b} \cdot yx' + x' \\ \qquad = h_{1,1}^{a,b} \cdot x + h_{2,1}^{a,b} \cdot \dfrac{b}{a} \cdot y + \dfrac{h_{3,1}^{a,b}}{a} \quad (21a) \\ a h_{1,3}^{a,b} \cdot xy' + b h_{2,3}^{a,b} \cdot yy' + y' \\ \qquad = h_{1,2}^{a,b} \cdot \dfrac{a}{b} \cdot x + h_{2,2}^{a,b} \cdot y + \dfrac{h_{3,2}^{a,b}}{b} \quad (21b) \end{cases}$$

We note that the systems of equations (19) and (21) are identical by taking:

$$H^{a,b} = \begin{pmatrix} h_{1,1} & h_{1,2}\frac{b}{a} & \frac{h_{1,3}}{a} \\ h_{2,1}\frac{a}{b} & h_{2,2} & \frac{h_{2,3}}{b} \\ h_{3,1} \cdot a & h_{3,2} \cdot b & 1 \end{pmatrix}. \qquad (22)$$

Therefore, for any point (x, y) and its projection (x', y') with the homography defined by H, the homography matrix projecting (ax, by) to (ax', by') is given by equation (22).

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. ACM, Jul. 1996, pp. 31–42.

[2] R. Ng, "Light field photography," Ph.D. dissertation, Stanford University, 2006.

[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 Trans. Graph.*, vol. 24, no. 3, pp. 765–776, Jul. 2005.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00, Jul. 2000, pp. 417–424.

[5] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[6] M. Daisy, D. Tschumperlé, and O. Lézoray, "A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting," in *SIGGRAPH Asia 2013 Technical Briefs*, Nov. 2013, pp. 8:1–8:4.

[7] M. Turkan and C. Guillemot, "Locally linear embedding based texture synthesis for image prediction and error concealment," in *IEEE Int. Conf. Image Processing*, Sep. 2012, pp. 3009–3012.

[8] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[9] P. Jain, R. Meka, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection." in *NIPS*, Dec. 2010, pp. 937–945.

[10] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," University of Illinois at Urbana-Champaign, Tech. Rep., Oct. 2009.

[11] K. Lee and Y. Bresler, "ADMiRA: Atomic decomposition for minimum rank approximation," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4402–4416, Sep. 2010.

[12] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[13] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *International Conference on Machine Learning*, Jul. 2011.

[14] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion." *SIAM J. Scientific Computing*, vol. 35, no. 5, 2013.

[15] Y. D. Kim and S. Choi, "Weighted nonnegative matrix factorization," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2009, pp. 1541–1544.

[16] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, Dec. 2012.

[17] B. Goldluecke and S. Wanner, "The variational structure of disparity and regularization of 4d light fields," in *Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 1003–1010.

[18] K. W. Chen, M. H. Chang, and Y. Y. Chuang, "Light field image editing by 4d patch synthesis," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Jun. 2015, pp. 1–6.

[19] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Computer Vision and Pattern Recognition (CVPR)*, Jun. 2008, pp. 1–8.

[20] F. L. Zhang, J. Wang, E. Shechtman, Z. Y. Zhou, J. X. Shi, and S. M. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, Feb. 2016.

[21] T. Broad and M. Grierson, "Light field completion using focal stack propagation," in *ACM SIGGRAPH 2016 Posters*, ser. SIGGRAPH '16, Jul. 2016, pp. 54:1–54:2.

[22] A. Jarabo, B. Masia, and D. Gutierrez, "Efficient propagation of light field edits," in *Proceedings of the V Ibero-American Symposium in Computer Graphics*, ser. SIACG 2011, 2011, pp. 75–80.

[23] Williem, K. W. Shon, and I. K. Park, "Spatio-angular consistent editing framework for 4d light field images," *Multimedia Tools and Applications*, vol. 75, no. 23, pp. 16615–16631, Dec. 2016.

[24] S.-H. Baek, I. Choi, and M. H. Kim, "Multiview image completion with space structure propagation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[25] T. Thonat, E. Shechtman, S. Paris, and G. Drettakis, "Multi-view inpainting for image-based scene editing and rendering," in *Fourth International Conference on 3D Vision (3DV)*, Oct. 2016, pp. 351–359.

[26] L. Yatziv, G. Sapiro, and M. Levoy, "Lightfield completion," in *International Conference on Image Processing (ICIP)*, vol. 3, Oct. 2004, pp. 1787–1790.

[27] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Commun. ACM*, vol. 55, no. 6, pp. 111–119, Jun. 2012.

[28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[29] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[30] X. Yuan and J. Yang, "Sparse and Low-Rank Matrix Decomposition Via Alternating Direction Methods," *optimization-online.org*, Nov. 2009.

[31] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, "SLRMA: Sparse low-rank matrix approximation for data compression," *IEEE Trans. Circuits Syst. Video Technol.*, Dec. 2015.

[32] Z. Boukouvalas, S. Said, L. Bombrun, Y. Berthoumieu, and T. Adal, "A new riemannian averaged fixed-point algorithm for MGGD parameter estimation," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2314–2318, Dec. 2015.

[33] E. Gómez, M. Gomez-Viilegas, and J. Marn, "A multivariate generalization of the power exponential family of distributions," *Communications in Statistics - Theory and Methods*, vol. 27, no. 3, pp. 589–600, 1998.

[34] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[35] "GIMP," https://www.gimp.org/.

[36] S. Wanner, S. Meister, and B. Goldlücke, "Datasets and benchmarks for densely sampled 4d light fields," in *Vision, Modeling & Visualization*, Sep. 2013, pp. 225–226.

[37] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2013.

[38] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[39] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[40] S. Lahabar and P. J. Narayanan, "Singular value decomposition on GPU using CUDA," in *IEEE International Symposium on Parallel Distributed Processing*, May 2009, pp. 1–10.