Photo Realistic Image Completion via Dense Correspondence

Jun-Jie Huang, *Student Member, IEEE,* and Pier Luigi Dragotti, *Fellow, IEEE*

***Abstract*—In this paper, we propose an image completion algorithm based on dense correspondence between the input image and an exemplar image retrieved from Internet. Contrary to traditional methods which register two images according to sparse correspondence, in this paper we propose a hierarchical PatchMatch method that progressively estimates a dense correspondence which is able to capture small deformations between images. The estimated dense correspondence has usually large occlusion areas that correspond to the regions to be completed. A nearest neighbor field (NNF) interpolation algorithm interpolates a smooth and accurate NNF over the occluded region. Given the calculated NNF, the correct image content from the exemplar image is transferred to the input image. Finally, as there could be a color difference between the completed content and the input image, a color correction algorithm is applied to remove the visual artifacts. Numerical results show that our proposed image completion method can achieve photo realistic image completion results.**

***Index Terms*—Image Completion, Dense Correspondence, Image Registration, EM Algorithm**

I. Introduction

IMAGE completion [1] tries to meet the increasing demand of editing personal photos, in particular by replacing an undesired image region (such as strangers, and construction sites) with a natural looking background which should be as close as possible to the real scene. The difficulty of this problem is directly related to the size of the region-of-interest (ROI) or “hole” to be completed which is assumed to be specified by users. The larger the “hole”, the higher the probability the missing content is non-stationary and this fact makes the image completion problem harder.

Single image completion algorithms [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] exploit the self-similarity of image regions (usually small image patches) within the input image and cover the ROI with similar image content. Impressive completion results are demonstrated by recent works when ROI is relatively small and a sufficient number of repetitive patterns exists. The main limitation of the single image based approaches is that the completion results tend to deviate from the real scene and become less realistic as the size of ROI grows. This happens because the self-similarity assumption is less satisfied. Recent single image completion methods are mainly example based [3], [4], [8], [7], [9], [11], [12] rather than diffusion based [13], [14]. The example based methods exploit redundancy within the input image itself and transfer image patches from the image regions outside the “hole” to the “hole” region. The central issue here is to establish accurate patch correspondence in order to find similar patches within the input image. Fast patch matching methods [5], [6], [15], [16], [17] play an important role in these algorithms. In particular, PatchMatch [5], [6] is a patch-based fast algorithm for dense correspondence estimation. The space-time completion method [3] searches similar patches and replaces the patches on the boundary of the “hole”. By iteratively searching and updating, the “hole” gradually shrinks. The image melding method [9], which generates natural looking results, further extends the search space of the generalized PatchMatch method [6] with reflection, gain and bias. He and Sun [11] proposed to utilize the statistics of the offsets between similar patches returned by PatchMatch to accelerate the matching process. Since image completion can be considered as a multi-label discrete optimization problem, other discrete optimization techniques, such as Graph Cuts [18], [19] and Belief Propagation [20], are also employed to address the single image completion problem. Pathak *et al.* [21] and Iizuka *et al.* [22] take the data driven approach for image completion by learning from an external dataset using deep neural networks.

Internet-based image completion algorithms [23], [24], [25], [26], [27] instead search for suitable image content from existing similar images available on Internet and can, in this way, overcome the limitations of the self-similarity assumption. An image is called an exemplar image if it has been taken from a similar viewpoint as the input image but may differ in camera parameters, illumination conditions, and with possible occlusions. With the help of exemplar images from the Internet, suitable image regions can be transferred to the input image, and under these conditions, accurate and photo realistic restoration becomes possible. In the pioneering work of Hays and Efros [24], they proposed to find images which are semantically similar to the input image and perform context matching and blending using a graphical model. However, their retrieved images could be taken from a distinct location and not satisfy our definition of exemplar image. The resultant image may not be faithful to the real scene and the incorrect reconstructions may lead to unrealistic images. There are many papers e.g. [23], [25], [27] that search and apply exemplar images for image completion. Amirshahi and Kondo [23] proposed to find a single homography correspondence between two images from obtained sparse correspondence (i.e. the matched SIFT keypoints [28] between images) and transfer patches with respect to the locations specified by the homography model. The limitation of the single homography model is that

Jun-Jie Huang is with the Department of Electrical and Electronic Engineering, Imperial College London, UK, e-mail: j.huang15@imperial.ac.uk.

Pier Luigi Dragotti is with the Department of Electrical and Electronic Engineering, Imperial College London, UK, e-mail: p.dragotti@imperial.ac.uk

the relationship between matched SIFT keypoints may not be well modeled if two images do not share the same camera center or contain piece-wise planar scenes. Whyte *et al.* [25] use a geometrical registration and a color registration for image completion. The geometrical registration is performed using multiple planes to approximate matched SIFT keypoints correspondence. Affine transformation is then applied to each color channel for color registration. A recent work [27] proposed by Zhu *et al.* automatically finds 20 exemplar images from Internet through SIFT keypoints matching with multiple homographies and line segments matching. Each exemplar image is warped to the input image by a mesh-based warping which is assisted with both point and line constraints. A scoring algorithm helps to select as completed image the one with the highest score. Moving from the single homography model [23] to multiple homographies [25], [27], a more general scene correspondence can be more accurately approximated. However, the sparse correspondence they relied on is still a discretized representation of the continuous image structure and may not be able to fully capture the correspondence between images.

For this reason, dense correspondence has recently been attracting more and more interests since it is able to discover pixel-wise correspondence of the shared content in two images. Dense correspondence algorithms can be divided into three main categories: PatchMatch-based approaches [9], [29], [30], [31], [32], [33], pyramid-based approaches [34], [35], [36], [37], and deep-learning-based approaches [38], [39], [40]. PatchMatch [5], [6] can also be applied for dense correspondence estimation between images. Non-rigid dense correspondence (NRDC) method [29] adopted a coarse-to-fine scheme where the operations on each scale consist of nearest neighbor search, consistent region aggregation, color mapping and search range adjustment. PatchMatch filter (PMF) method [30] proposed to apply superpixel to link the PatchMatch method and edge aware filtering techniques (e.g. guided filter). PMF method gets rid of the runtime dependency on patch size and improves the speed of the PatchMatch method. Pyramid-based approaches [34], [35], [36], [37] use a pyramid graph model for dense correspondence. SIFT flow [36] is a computational framework which extracts fixed scale and orientation SIFT descriptor [28] at each pixel for matching and produces dense and pixel-to-pixel correspondence. Different from SIFT flow where the graph model only connects neighboring pixels at the same level, the edges of the graph model in the deformable spatial pyramid (DSP) method [35] links neighboring cells as well as parent-child node across adjacent layers. Deep-learning-based approaches [38], [39], [40] apply a multi-layer network to discover shared contents between images. DeepMatching method [38], [39] matches two images by computing patch correlations in a bottom-up, then top-down fashion. There is no feature representation learned in DeepMatching. Patches from the first image are used as convolutional filters for the second image. Flownet [40] utilizes an end-to-end trained convolutional neural network (CNN) for optical flow estimation.

In this paper, we propose to tackle the Internet-based image completion problem using a robustly estimated dense correspondence. The advantage of utilizing dense correspondence for image completion is two-fold. First, dense correspondence gives us a complete landscape of the correspondence between images, which captures some local deformations that might be missed by sparse correspondence. With the dense correspondence, the Internet-based image completion problem becomes a surface fitting problem with a segment of the original surface missing. Therefore, our objective becomes finding a surface which is consistent with the acquired correspondence outside the "hole" region under a smoothness constraint. Second, dense correspondence provides a rich set of color correspondence between the input image and the retrieved exemplar image. This helps remove color discrepancy on the completed region. This is in contrast with the color correspondence provided by sparse correspondence which is usually not sufficient for robust color correction.

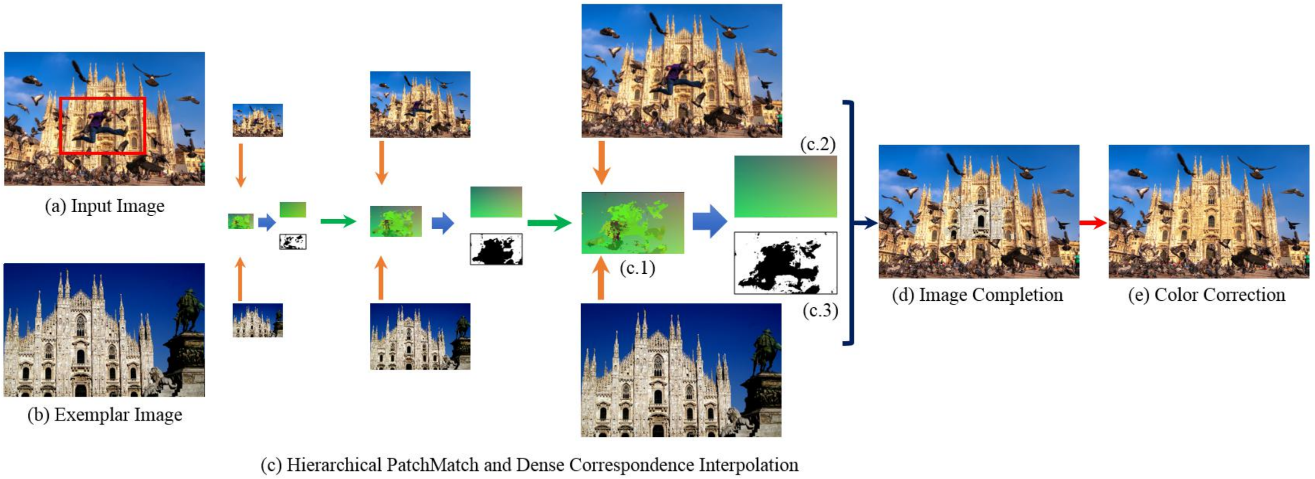
A hierarchical framework is proposed to progressively achieve image completion based on dense correspondence. The hierarchical structured PatchMatch imposes smoothness constraint on the estimated dense correspondence. However, the obtained dense correspondence is usually noisy and contains outliers. We propose to use an Expectation-Maximization (EM) based approach with kernel ridge regression to jointly denoise the obtained correspondence and interpolate the correspondence in the "hole" region at each hierarchical level. Within the hierarchical framework, the EM model parameters of the current level are used to initialize the parameters of the next level and this leads to a faster convergence. At the final level, a completed image is obtained by transferring image content with respect to the interpolated dense correspondence and performing color correction to remove the possible color differences between two images.

The rest of the paper is organized as follows. Section II gives an overview of our proposed image completion method. Section III introduces the proposed image completion using dense correspondence framework, and Section IV describes color correction based on estimated dense correspondence. Section V presents the results of our extensive experimental work and Section VI draws conclusions.

II. Overview

Our algorithm makes the same assumptions as other Internet-based image completion algorithms [23], [25], [27]. Specifically, we assume that there are many images on Internet similar to the image that needs to be completed and some similar exemplar images are already available in that we rely on existing searching engines to find exemplar images. This is particularly true for images of famous landmarks. Moreover, we assume that a region-of-interest (ROI) is given by the user indicating a region that requires completion.

The overview of the proposed image completion method is illustrated in Fig. 1. The required ROI is a rectangle region rather than a detailed contour. However, we are not trying to replace the whole ROI but only the occluded part. Gaussian image pyramids are built for both the input image and the retrieved exemplar image. We then progressively estimate dense correspondence from coarse-to-fine pyramid levels. An



(a) Input Image

(b) Exemplar Image

(c.1)

(c.2)

(c.3)

(d) Image Completion

(e) Color Correction

(c) Hierarchical PatchMatch and Dense Correspondence Interpolation

Fig. 1. Overview of the proposed image completion algorithm: (a) The input image with a region-of-interest. (b) A retrieved exemplar image. (c) Our proposed framework progressively estimates dense correspondence (e.g. (c.1)) from coarse-to-fine pyramid levels using a hierarchical PatchMatch and interpolates a smooth dense correspondence (e.g. (c.2)) over the occluded region based on the estimated inlier correspondences (e.g. (c.3)). (d) Image completion based on the interpolated dense correspondence. (e) Color correction based on the dense correspondence on the estimated inlier region. (Please note that the dense correspondence has four dimensions at each location, the color representation only shows the first three dimensions.)

Expectation-Maximization (EM) algorithm jointly performs inliers/outliers estimation and dense correspondence interpolation (Fig. 1 (c)) to obtain a smooth dense correspondence over the “hole” region. The current level dense correspondence and model parameters of the EM algorithm will be passed to the next pyramid level. The “hole” can then be filled using the image content from the exemplar image. More precisely, the interpolated dense correspondence indicates the correct pixel location on the exemplar image of a pixel on the input image. By transferring the corresponding pixel values from the exemplar image, the undesired image content in the input image can be replaced (Fig. 1 (d)). However, it is very likely that the color between the input image and the retrieved exemplar image is incompatible (for example, in Fig. 1 (a) and (b)). Based on the established dense correspondence, our color correction algorithm is applied to fit a smooth B-spline color transfer function for each color channel in RGB color space. It corrects the color differences and improves visual quality as shown in Fig. 1 (e).

III. Image Completion using Dense Correspondence

Let us denote the input image with $\mathcal{I}^1$ and the exemplar image with $\mathcal{E}^1$. We generate from $\mathcal{I}^1$ an image pyramid with decreasing resolutions $\{\mathcal{I}^k\}_{k=1}^K$, scaled down by a factor 2^{k-1}. Similarly, an image pyramid $\{\mathcal{E}^k\}_{k=1}^K$ is generated from $\mathcal{E}^1$.

The k^{th} level nearest neighbor field (NNF) $\mathcal{F}^k$ relates every image patch on $\mathcal{I}^k$ with its nearest neighbor (NN) patch on $\mathcal{E}^k$. As the same scene on $\mathcal{I}^k$ and $\mathcal{E}^k$ may differ in viewpoint, we assume there are $D = 4$ degrees of freedom for the NN patch, i.e. position (u, v), scale s, and orientation θ. Thus, $\mathcal{F}^k$ has the same size as $\mathcal{I}^k$ and has 4 matching parameters at each location. We denote with $\mathcal{I}_r^k(\boldsymbol{p})$ the $(2r+1) \times (2r+1)$ patch centered at $\boldsymbol{p} = (x, y)$ on $\mathcal{I}^k$, with $\mathcal{F}^k(\boldsymbol{p})$ the matching parameter (u, v, s, θ) of the NN patch for patch $\mathcal{I}_r^k(\boldsymbol{p})$, and with $\mathcal{E}_r^k(\mathcal{F}^k(\boldsymbol{p}))$ the NN patch on $\mathcal{E}^k$ at location (u, v) with patch radius $s \times r$, and orientation θ.

A. Basic PatchMatch

In this section, we briefly review PatchMatch and the variations we have introduced to make it more suitable to our problem. For a detailed description of the classical PatchMatch, please refer to [5], [6].

There are three key steps in PatchMatch, i.e. random initialization, neighboring propagation, and random search. As natural images are highly structured, good matching parameters in a randomly initialized NNF can be propagated to its spatial neighbors with minor adjustment. For example, a patch $\mathcal{I}_r^1(\boldsymbol{p})$ can update its matching parameter $\mathcal{F}^1(\boldsymbol{p})$ by using those of its spatial neighbors $\Psi_{\boldsymbol{p}}$:

$$\mathcal{F}^1(\boldsymbol{p}) = \arg \min_{\mathcal{F}^1(\boldsymbol{p}_i)} \left\{ S\left(\mathcal{F}^1(\boldsymbol{p}_i)\right) | \boldsymbol{p}_i \in \boldsymbol{p} \cup \Psi_{\boldsymbol{p}} \right\}, \quad (1)$$

where $S(\mathcal{F}^1(\boldsymbol{p}_i))$ is the matching cost between $\mathcal{I}_r^1(\boldsymbol{p})$ and $\mathcal{E}_r^1(\mathcal{F}^1(\boldsymbol{p}_i) + \boldsymbol{\omega})$ with $\boldsymbol{\omega}$ being the affine adjustment.

Propagation is specified by how the spatial neighbors $\Psi_{\boldsymbol{p}}$ is defined. There are two kinds of propagation in PatchMatch [5], i.e. type 1: $\Psi_{\boldsymbol{p}} = \{\boldsymbol{p} - \mathbf{1}_h, \boldsymbol{p} - \mathbf{1}_v\}$ and type 2: $\Psi_{\boldsymbol{p}} = \{\boldsymbol{p} + \mathbf{1}_h, \boldsymbol{p} + \mathbf{1}_v\}$, where $\mathbf{1}_h = (1, 0)$ and $\mathbf{1}_v = (0, 1)$. Bailer *et al.* [41] proposed that propagation should be performed in two more directions (i.e. type 3: $\Psi_{\boldsymbol{p}} = \{\boldsymbol{p} - \mathbf{1}_h, \boldsymbol{p} + \mathbf{1}_v\}$ and type 4: $\Psi_{\boldsymbol{p}} = \{\boldsymbol{p} + \mathbf{1}_h, \boldsymbol{p} - \mathbf{1}_v\}$) so that good matching parameters can be propagated to any position on the input image. This is the strategy adopted in this paper.

After propagation has been performed at each position, random search is applied to avoid matching parameters being trapped in local minima. For a location $\boldsymbol{p}$, a small number of matching parameters will be randomly sampled near $\mathcal{F}^1(\boldsymbol{p})$ in the parameter space within a predefined maximum random search range. The sampling radius shrinks by 2 until the range is smaller than 1. If one of them can provide lower matching cost, $\mathcal{F}^1(\boldsymbol{p})$ will be updated with it. The problem of the random search in classical PatchMatch is that random search is intensively applied in every location with the same sampling radius. This introduces a huge number of ineffective computation, since a non-adaptive sampling radius will generate many irrelevant matching parameters with high matching costs. In Section III.*D*, we propose an adaptive random search which adaptively selects candidate matching parameters based on their estimated reliability.

B. Feature Representation and Distance Metric

In optical flow estimation, two images are temporally adjacent and have no obvious color difference. However, in our setting the exemplar images are normally different from the input image in viewpoint, illumination and color condition.

The commonly used patch feature descriptors in optical flow include RGB, intensity invariant color representation, gradients, SIFT descriptor [28], and Census transform [42]. Matching patches using RGB will result in very noisy NNF as RGB is vulnerable to the aforementioned image variations. SIFT descriptor can provide better performance, however, this is a computationally expensive solution as a huge number of patches need to be evaluated in PatchMatch. Census transform is a binary descriptor and is robust to illumination changes. The advantage of binary descriptors [43], [44], [45] is that a patch is represented by a binary string and matching is easy to compute, since Hamming distance instead of l_2 distance is applied for feature distance evaluation.

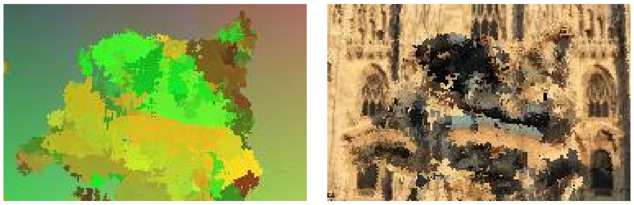
In this paper, we utilize BRIEF descriptor [44] as feature representation. It is one of the most widespread binary descriptors and has comparable performance with floating-point descriptors [46]. Census transform [42] is a special case of BRIEF. According to BRIEF descriptor [44], a small number of binary tests with randomly generated test locations can yield good discrimination power. A binary test τ for a patch is defined as:

$$\tau(c, \boldsymbol{p}, \boldsymbol{q}) = \begin{cases} 1 & \text{if } c(\boldsymbol{p}) < c(\boldsymbol{q}), \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where c represents one of the feature channels, $\boldsymbol{p}$ and $\boldsymbol{q}$ are two positions on the patch.

In order to gain color invariant property, RGB color image is converted to CIE $\mathrm{La^*b^*}$ color space where L is the illuminance channel, and $\mathrm{a^*}$ and $\mathrm{b^*}$ are chrominance channels. Two gradient channels $\nabla_{\mathrm{x}}\mathrm{L}$ and $\nabla_{\mathrm{y}}\mathrm{L}$ have also been included to account for edges. Therefore the final feature channel is $\mathcal{C} = (\mathrm{L}, \mathrm{a^*}, \mathrm{b^*}, \nabla_{\mathrm{x}}\mathrm{L}, \nabla_{\mathrm{y}}\mathrm{L})$.

There are n_b binary tests that are generated $\{\tau(c_i, \boldsymbol{p}_i, \boldsymbol{q}_i)\}_{i=1}^{n_b}$. For each binary test $\tau(c, \boldsymbol{p}, \boldsymbol{q})$, c is randomly selected from feature channel set $\mathcal{C}$, and $\boldsymbol{p}$ and $\boldsymbol{q}$ are randomly sampled from discrete locations on a $(2r+1) \times (2r+1)$ patch. After the n_b binary tests are determined, all the patches will use these tests to construct the BRIEF descriptor. The BRIEF descriptor of a patch can be expressed as a binary string being the binary counterpart of $\sum_{i=1}^{n_\mathrm{b}} 2^{i-1}\tau(c_i, \boldsymbol{p}_i, \boldsymbol{q}_i)$. Thus, the distance metric $S(\cdot)$ in Eqn. (1) measures the Hamming distance between the BRIEF descriptors of two patches.


(a) Estimated NNF.

(b) Reconstructed image.

Fig. 2. (a) An example of the estimated nearest neighbour field. (b) The reconstructed image based on the estimated NNF. The estimated NNF is noisy and with a large region with occlusions. The quality of the NNF can be reflected in the reconstructed image.

C. Nearest Neighbor Field Interpolation

NNF interpolation is necessary to produce a smooth and accurate NNF over the occluded region. With the basic PatchMatch using BRIEF descriptor, the acquired NNF is still relatively noisy and with a large outlier region in the occluded part (see Fig. 2 for an example). Only with a well refined NNF, the corresponding pixel values on the exemplar image can be faithfully transferred to the input image.

We apply an Expectation-Maximization (EM) approach similar to [47], [48], [49] to jointly estimate inliers/outliers in the observed NNF and interpolate NNF within ROI using kernel ridge regression. There are two reasons for that. First, traditional approaches, such as affine transform, and thin-plate spline model, are vulnerable to noise and outliers. The EM algorithm can be used to identify the noisy data and outliers from the observed data. In turn, a reliable NNF interpolation model can be constructed using only the inlier data points. Second, the kernel ridge regression can flexibly adapt to the variations in NNF. In this way, the flexibility gained through dense correspondence is preserved.

1) Likelihood Formulation: We assume that there are N matched pixel-wise correspondences $\{(\boldsymbol{p}_i, \mathcal{F}^k(\boldsymbol{p}_i))\}_{i=1}^{N}$ within the ROI of pyramid level k. To simplify notation, we denote $(\boldsymbol{x}_i, \boldsymbol{y}_i) = (\boldsymbol{p}_i, \mathcal{F}^k(\boldsymbol{p}_i))$ for $i = 1, 2, ..., N$, with $\mathbf{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)^{\mathrm{T}} \in \mathbb{R}^{N\times 2}$ and $\mathbf{Y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_N)^{\mathrm{T}} \in \mathbb{R}^{N\times D}$.

The NNF generated by PatchMatch is assumed to be a mixture of Gaussian distributed inliers and uniformly distributed outliers. Moreover the components of the NNF are assumed to be independent. An indicator $z_i \in \{0, 1\}$ is defined for each data pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ to indicate it being an inlier ($z_i = 1$) or an outlier ($z_i = 0$). The fitting error of inliers are assumed to follow a Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with zero mean and variance $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$. The outliers are the data pairs which cannot be well described by NNF interpolation function $\boldsymbol{f}$. We assume the outliers are uniformly distributed among the parameter space. Under the above assumptions, the likelihood function is given as follows:

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = & \prod_{i=1}^{N} \sum_{z_i} p(\boldsymbol{y}_i, z_i | \boldsymbol{x}_i, \boldsymbol{\theta}) \\ = & \prod_{i=1}^{N} \left(\gamma \frac{\exp(-d_i)}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} + \frac{1-\gamma}{V} \right), \end{aligned} \tag{3}$$

where model parameter $\boldsymbol{\theta} = \{\boldsymbol{f}, \gamma, \boldsymbol{\Sigma}\}$, γ represents the percentage of inliers, $d_i = \frac{1}{2}\left(\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i)\right)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i)\right)$, $\det(\cdot)$ is the determinant of a matrix, and V is the volume of the NNF parameter space.

The underlying interpolation function should be smooth. Let us assume a smooth prior on the NNF interpolation function as $p(\boldsymbol{f}) \propto \exp\left(-\frac{\lambda}{2}\phi(\boldsymbol{f})\right)$ where $\phi(\boldsymbol{f})$ is a smoothness function, and λ is a regularization parameter. With the likelihood defined in Eqn. (3), the smooth prior on $\boldsymbol{f}$, and assuming a uniform prior on parameter γ and $\boldsymbol{\Sigma}$, the posterior distribution of the model parameter can be estimated via Bayes rule as $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{f})$. The optimal model parameter $\boldsymbol{\theta}^{*}$ is then estimated from a Maximum A Posteriori (MAP) of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{*} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{f}). \tag{4}$$

2) EM Algorithm: Let us define $p_i = p(z_i = 1|\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta})$ and $P = \sum_{i=1}^{N} p_i$. Replacing Eqn. (3) into Eqn. (4), taking a negative logarithm of it and removing the terms independent of $\boldsymbol{\theta}$ yields the negative log-likelihood function:

$$\begin{aligned} Q(\boldsymbol{\theta}) \;\; = \;\; & \sum_{i=1}^{N} p_i d_i + \frac{DP}{2}\ln\det(\boldsymbol{\Sigma}) - P\ln\gamma \\ & - (N-P)\ln(1-\gamma) + \frac{\lambda}{2}\phi(\boldsymbol{f}). \end{aligned} \tag{5}$$

We apply an EM algorithm to iteratively estimate model parameter $\boldsymbol{\theta} = \{\boldsymbol{f}, \gamma, \boldsymbol{\Sigma}\}$. At iteration $t+1$, the EM algorithm iterates between an Expectation step (E-step), which computes the posteriori probability of the latent variable $p(z_i|\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta}_t)$ using the parameter $\boldsymbol{\theta}_t = \{\boldsymbol{f}_t, \gamma_t, \boldsymbol{\Sigma}_t\}$ from previous iteration, and a Maximization step (M-step), which estimates a new parameter $\boldsymbol{\theta}_{t+1}$ that minimizes the negative log-likelihood $Q(\boldsymbol{\theta})$.

During the E-step, the posteriori probability of the latent variable z_i can be estimated using Bayes rule with the model parameter $\boldsymbol{\theta}_t$:

$$p_i = \frac{\gamma_t \exp\left(-d_{i,t}\right)}{\gamma_t \exp\left(-d_{i,t}\right) + \frac{1-\gamma_t}{V}\sqrt{\det(2\pi\boldsymbol{\Sigma}_t)}}, \tag{6}$$

where $d_{i,t} = \frac{1}{2}\left(\boldsymbol{y}_i - \boldsymbol{f}_t(\boldsymbol{x}_i)\right)^{\mathrm{T}} \boldsymbol{\Sigma}_t^{-1} \left(\boldsymbol{y}_i - \boldsymbol{f}_t(\boldsymbol{x}_i)\right)$.

In the M-step, model parameter is updated so as to minimize the negative log-likelihood. To exclude the influence of the outliers, we binarize the posteriori probability such that $p_i = 1$ if $p_i > \varphi$ ($\varphi = 0.5$), $p_i = 0$ otherwise. We take the derivative of Eqn. (5) with respect to γ and $\boldsymbol{\Sigma}$, respectively, and set them to zero. The updated parameters are then given by:

$$\begin{cases} \gamma_{t+1} \;\; = \;\; \frac{\mathrm{tr}(\mathbf{P})}{N}, \\ \boldsymbol{\Sigma}_{t+1}(i,i) \;\; = \;\; \frac{(\mathbf{Y}(:,i)-\mathbf{Z}(:,i))^{\mathrm{T}}\mathbf{P}(\mathbf{Y}(:,i)-\mathbf{Z}(:,i))}{\mathrm{tr}(\mathbf{P})}, \end{cases} \tag{7}$$

where $\mathbf{P} = \mathrm{diag}(p_1, ..., p_N)$ is a $N \times N$ diagonal matrix with diagonal values specified by $p_1, ..., p_N$, $\mathbf{Y}(:,i)$ is the i^{th} column of $\mathbf{Y}$, $\mathrm{tr}(\cdot)$ returns the trace of a matrix, and $\mathbf{Z} = (\boldsymbol{f}_t(\boldsymbol{x}_1), ..., \boldsymbol{f}_t(\boldsymbol{x}_N))^{\mathrm{T}} \in \mathbb{R}^{N\times D}$ are the estimated NNF using $\boldsymbol{f}_t$.

We group the terms in Eqn. (5) related to the NNF interpolation function $\boldsymbol{f}$ and define an energy function $E(\boldsymbol{f})$ function as:

$$E(\boldsymbol{f}) = \frac{1}{2}\sum_{i=1}^{N} p_i \left(\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i)\right)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i)\right) + \frac{\lambda}{2}\phi(\boldsymbol{f}). \tag{8}$$

Eqn. (8) is a special form of Tikhonov regularization. The interpolation function $\boldsymbol{f}_{t+1}$ is updated by minimizing the energy function $E(\boldsymbol{f})$ with respect to $\boldsymbol{f}$. Let the smoothness function be defined as $\phi(\boldsymbol{f}) = ||\boldsymbol{f}||_{\mathcal{H}}^2$ where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS). In this paper, the reproducing kernel is selected as Gaussian. According to the representer theorem [50], the NNF interpolation function $\boldsymbol{f}$ is in the form of weighted sum of kernel products:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n)\boldsymbol{w}_n, \tag{9}$$

where $k(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\frac{||\boldsymbol{a}-\boldsymbol{b}||^2}{\beta}\right)$ is a Gaussian reproducing kernel with filter range defined by β, and $\boldsymbol{w}_n \in \mathbb{R}^D$ is the weight associated with $k(\cdot, \boldsymbol{x}_n)$.

The norm in RKHS $\mathcal{H}$ can be expressed as $||\boldsymbol{f}||_{\mathcal{H}}^2 = \widetilde{\mathbf{W}}^{\mathrm{T}}\mathbf{K}\widetilde{\mathbf{W}}$ with $\widetilde{\mathbf{W}} = \left(\boldsymbol{w}_1^{\mathrm{T}}, ..., \boldsymbol{w}_N^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}^{ND\times 1}$ being the coefficient column vector and $\mathbf{K} \in \mathbb{R}^{ND\times ND}$ being a $N \times N$ block matrix where the $(i,j)^{th}$ block has size $D \times D$ and its entries are with value $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By replacing $\boldsymbol{f}$ in Eqn. (8) with the form in Eqn. (9), the energy function can be expressed in matrix form as follows:

$$E(\boldsymbol{f}) = \frac{1}{2}\left(\widetilde{\mathbf{Y}} - \mathbf{K}\widetilde{\mathbf{W}}\right)^{\mathrm{T}} \widetilde{\mathbf{P}} \left(\widetilde{\mathbf{Y}} - \mathbf{K}\widetilde{\mathbf{W}}\right) + \frac{\lambda}{2}\widetilde{\mathbf{W}}^{\mathrm{T}}\mathbf{K}\widetilde{\mathbf{W}}, \tag{10}$$

where $\widetilde{\mathbf{Y}} = \left(\boldsymbol{y}_1^{\mathrm{T}}, ..., \boldsymbol{y}_N^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}^{ND\times 1}$ is a column vector, $\widetilde{\mathbf{P}} = \mathbf{P} \otimes \boldsymbol{\Sigma}^{-1}$ with $\otimes$ being Kronecker product.

Taking the derivative of Eqn. (10) with respect to $\widetilde{\mathbf{W}}$ and setting it to zero leads to the following expression for the coefficient matrix $\widetilde{\mathbf{W}}$:

$$\widetilde{\mathbf{W}} = (\mathbf{K} + \lambda\widetilde{\mathbf{P}}^{-1})^{-1}\widetilde{\mathbf{Y}}. \tag{11}$$

Though a closed form solution exists for $\widetilde{\mathbf{W}}$, it is too expensive to compute the inverse of $(\mathbf{K} + \lambda\widetilde{\mathbf{P}}^{-1})$ which can be decomposed into D matrix inverses of size $N \times N$. In order to reduce the computational complexity, a fast approximation method [48], [51] suggests that we can use a subset of the observed data as control points $\{\widetilde{\boldsymbol{x}}_m\}_{m=1}^{M}$ with $M \ll N$ and this reduces the size of the matrix to $M \times M$. With this fast approximation, the computational complexity is reduced from $O(N^3)$ to $O(M^3)$. The NNF interpolation function is then only represented by M kernel products:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{m=1}^{M} k(\boldsymbol{x}, \widetilde{\boldsymbol{x}}_m)\boldsymbol{w}_m. \tag{12}$$

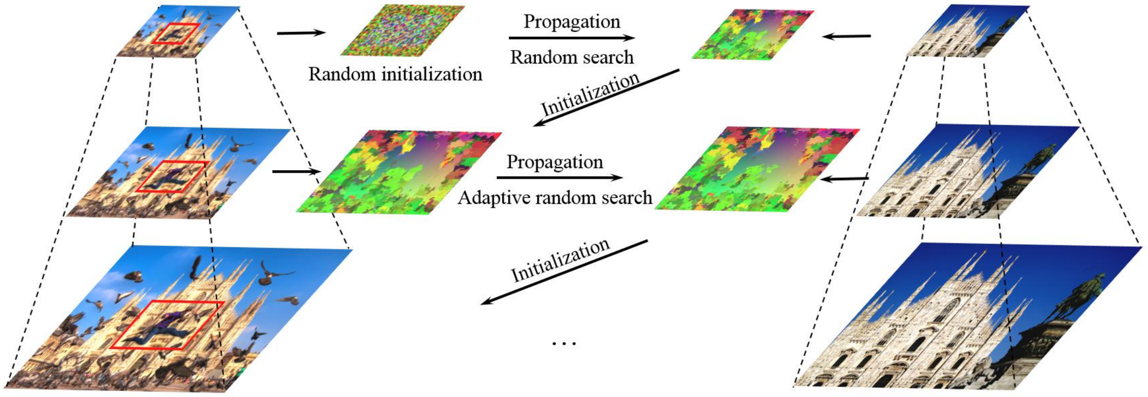


Fig. 3. Hierarchical PatchMatch flow diagram. Image pyramids for the input image and the exemplar image are built. The dense correspondence is estimated from the higher to the lower pyramid level. Dense correspondence at lower pyramid level is initialized using the result from previous level. Once a reference NNF predicted by previous level interpolation function is available, random search is performed in an adaptive way to reduce complexity. (The hole region is marked with red rectangle.)

By using M control points $\{\widetilde{\boldsymbol{x}}_m\}_{m=1}^{M}$, the coefficient matrix $\widetilde{\mathbf{W}} \in \mathbb{R}^{MD\times 1}$ are determined using the closed form expression:

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{U}}^{\mathrm{T}}\widetilde{\mathbf{P}}_{\mathbf{I}}\widetilde{\mathbf{U}} + \lambda\widetilde{\mathbf{Q}})^{-1}\widetilde{\mathbf{U}}^{\mathrm{T}}\widetilde{\mathbf{P}}_{\mathbf{I}}\widetilde{\mathbf{Y}}, \qquad (13)$$

where $\widetilde{\mathbf{U}} \in \mathbb{R}^{ND\times MD}$ is a $N \times M$ block matrix with the $(i, j)^{th}$ block being a $D \times D$ matrix in which its entries are with value $k(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_j)$, $\widetilde{\mathbf{P}}_{\mathbf{I}} = \mathbf{P} \otimes \mathbf{I}_D$ ($\mathbf{I}_D$ is a $D \times D$ identity matrix), and $\widetilde{\mathbf{Q}} = \mathbf{Q} \otimes \mathbf{\Sigma}$ with $\mathbf{Q} \in \mathbb{R}^{M\times M}$ being the intra Gram matrix and $\mathbf{Q}(i, j) = k(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)$.

The NNF interpolation function $\boldsymbol{f}$ is thus defined by the selected M control points $\{\widetilde{\boldsymbol{x}}_m\}_{m=1}^{M}$ as well as the coefficient matrix $\widetilde{\mathbf{W}}$. If the control points are fixed through the EM algorithm, the NNF interpolation function $\boldsymbol{f}$ is solely determined by $\widetilde{\mathbf{W}}$. The interpolation function $\boldsymbol{f}_{t+1}$ at iteration $t + 1$ is thus obtained through Eqn. (13) by using the covariance matrix $\mathbf{\Sigma}_{t+1}$ from Eqn. (7) and the updated probability matrix $\mathbf{P}$. One may argue that it is only necessary to find the interpolation function for position parameters (u, v). This can save half computation. However, the other two parameters (s, θ) provide important information for posteriori probability estimation and a well estimated probability matrix $\mathbf{P}$ leads to a faster convergence of the EM algorithm.

The stopping criterion of the EM algorithm is activated when the negative log-likelihood converges or the maximum iteration $\mathcal{T}_{\mathrm{EM}}$ has been reached.

D. Hierarchical PatchMatch with NNF Interpolation

We propose to progressively estimate a smooth NNF within the ROI using a hierarchical PatchMatch and interpolate an accurate NNF over the occluded region. The flow diagram of the hierarchical PatchMatch is shown in Fig. 3. The two main reasons for using the hierarchical framework are as follows:

- First, the basic PatchMatch generally produces noisy NNF as its objective is to find nearest neighbor patches rather than a smooth NNF. The coarse-to-fine scheme can impose smoothness constraint on NNF. With a fixed patch size, a patch in a higher pyramid level has relatively larger spatial range and imposes stronger spatial smoothness constraints. Proceeding to lower pyramid levels, the NNF is refined with a relatively smaller patch size and has higher matching accuracy. The lower level NNF is initialized by up-sampling previous level NNF. Good matching parameters can be passed through the hierarchy, while erroneous matching parameters are unlikely to be retained on NNF at different pyramid levels.
- Second, NNF interpolation can benefit from the hierarchical PatchMatch. From Eqn. (13), the estimation of $\mathbf{P}$ and $\mathbf{\Sigma}$ is essential to obtain a reliable $\widetilde{\mathbf{W}}$ which corresponds to a smooth and accurate NNF. The selection of control points can also affect the quality of NNF interpolation. As the overall problem is non-convex, a good initialization plays a key role in the EM algorithm. The EM algorithm can have a "warm" start by using the model parameters of the previous level for initialization. This gives a faster convergence. Moreover, the control points can be selected based on their probability of being inliers using the model parameters estimated in the previous level. A better control points selection scheme will improve NNF interpolation accuracy and make full use of the limited number of control points.

With the $(k + 1)^{th}$ level NNF $\mathcal{F}^{k+1}$, the k^{th} level NNF $\mathcal{F}^k$ is initialized by up-sampling $\mathcal{F}^k$ by a factor 2. The spatial range of $\mathcal{F}^k$ has been doubled compared to $\mathcal{F}^{k+1}$, while the scale and the rotation range are kept the same. In order to faithfully pass the matching parameters between different pyramid levels, the up-sampling method applied is the nearest neighbor interpolation:

$$\mathcal{F}^k(2\boldsymbol{p} - (i, j)) = (2u, 2v, s, \theta), \qquad (14)$$

where $\mathcal{F}^{k+1}(\boldsymbol{p}) = (u, v, s, \theta)$, and $i, j = \{0, 1\}$.

The complexity of PatchMatch algorithm is proportional to the number of checked matching parameters from neighboring patches and randomly sampled parameters. With the initialized NNF $\mathcal{F}^k$, propagation and random search are performed in a way similar to the one described in Section III.*A*. With the existence of the ROI, only the pixels in ROI will be processed. Moreover two approaches are proposed to reduce the complexity of random search. First, random search can be performed in an adaptive way. Once the interpolation function of the previous level is successfully estimated, it can be used to predict a reference NNF $\mathcal{F}^k_{\mathrm{ref}}$ to guide the random search. For a randomly generated parameter $\mathcal{F}^k_{\mathrm{rnd}}(\boldsymbol{p})$ around $\mathcal{F}^k(\boldsymbol{p})$, if it is too far from the reference NNF (i.e. $\exists i \in \{1, 2, ..., D\}$: $\boldsymbol{d}_{\mathrm{rnd}}\,(i) > \boldsymbol{\tau}_{\mathrm{rnd}}\,(i)$ where $\boldsymbol{d}_{\mathrm{rnd}} = \left|\mathcal{F}^k_{\mathrm{rnd}}(\boldsymbol{p}) - \mathcal{F}^k_{\mathrm{ref}}(\boldsymbol{p})\right|$ and $\boldsymbol{\tau}_{\mathrm{rnd}} \in \mathbb{R}^D$ is the adaptive search threshold), it will unlikely be a good candidate and thus is abandoned. This leads to an early termination for the unreliable matching parameters and saves computation. Second, random search is conducted for the patches with edges which are determined via Canny edge detection. A random search is then performed only on patches which contains pixels with edge value larger than a threshold τ_{edge}. For the patches extracted from smooth region, random search would have a high probability of introducing wrong matching parameters which have even lower matching cost than the correct one. This can reduce complexity as well as avoid incorrect matching parameters being adopted.

The model parameters at level k is initialized using the model parameter from level $k+1$. Instead of using the exact value of $\boldsymbol{\Sigma}$ at previous level, it is re-scaled by a factor $\kappa > 1$ for initialization to avoid being trapped in a local minimum. The random selection of control points directly affects the results of the NNF interpolation algorithm. If outliers are picked as control points, the NNF interpolation could become less effective. When the EM algorithm of the previous level produces a reliable NNF interpolation function, the probability p_i of a data pair being inlier at current level can be predicted from Eqn. (6). Control points are selected from the data pairs with inlier probability $p_i > \rho$.

E. Image Completion

A smooth NNF $\mathcal{F}$ over the occluded region can be obtained through our proposed hierarchical PatchMatch and NNF interpolation:

$$\mathcal{F} = \mathbf{W}\mathbf{U}^{\mathrm{T}}, \tag{15}$$

where $\mathbf{W} = (\boldsymbol{w}_1, ..., \boldsymbol{w}_M) \in \mathbb{R}^{D\times M}$ is the interpolation coefficient matrix, and $\mathbf{U} \in \mathbb{R}^{N\times M}$ is the inter Gram matrix with $\mathbf{U}(i,j) = k(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_j)$ and $\widetilde{\boldsymbol{x}}_j$ being the control points from pyramid level 1.

Let us define a mask $\mathcal{M}$ within the ROI which has value 1 on the locations with $p_i \geq \tau_p$ and value 0 otherwise. As the BRIEF descriptor is robust to small occlusions, the boundary of $\mathcal{M}$ may still contain image content that need to be replaced. To include all the occluded image region, the mask $\mathcal{M}$ is eroded a few times. Let denote with $\mathcal{M}_i$ the region with mask value i for $i = 0, 1$. The mask region $\mathcal{M}_0$ indicates the image content on the input image which should be replaced. Every pixel $\boldsymbol{p} \in \mathcal{M}_0$ on the input image is replaced by the corresponding pixel value on location $\mathcal{E}^1(\mathcal{F}(\boldsymbol{p}))$ from the exemplar image. The completed input image is denoted by $\mathcal{I}^c$.

Algorithm 1 summarizes our proposed image completion method based on dense correspondence.

Algorithm 1 Image Completion via Dense Correspondence

```
Input: Input image $\mathcal{I}^1$ with a ROI, exemplar image $\mathcal{E}^1$;
Output: Completed image $\mathcal{I}^c$;
Build image pyramids $\{\mathcal{I}^k\}_{k=1}^K$ and $\{\mathcal{E}^k\}_{k=1}^K$;
Randomly initialize the coarsest level NNF $\mathcal{F}^K$;
for $i = 1 : K$
    Perform PatchMatch between $\mathcal{I}^{K+1-i}$ and $\mathcal{E}^{K+1-i}$ within ROI;
    Interpolate NNF within ROI using the EM algorithm in Section III.C;
    if $i < K$
        Initialize the next level NNF $\mathcal{F}^{K-i}$ and EM algorithm model parameter $\boldsymbol{\theta}^{K-i}$;
    end if
end for
Perform image completion to $\mathcal{I}^1$as in Section III.E.
```

IV. Color Correction

Though images are taken near the same landmark, different camera parameters, shooting time, and shooting angles could result in significant color difference between the input image and the selected exemplar images. In order to have a visually pleasant result, color correction needs to be performed.

Different from the color transfer methods [52], [53] where color correspondences are not available, pixel-wise color correspondences $\mathcal{D} = \{(x_{ci}, y_{ci})\}_{i=1}^{|\mathcal{M}_1|}$ have been established by the computed dense correspondence $\mathcal{F}^1$ within $\mathcal{M}_1$ with (x_{ci}, y_{ci}) being the corresponding color values on input image and exemplar image. A global color correction model can be used to minimize the color differences in a way similar to NRDC [29]. For each RGB color channel, a color transfer curve can be fitted using color correspondence $\mathcal{D}$ and applied to $\mathcal{I}^c$ within $\mathcal{M}_0$. The color transfer curve $\boldsymbol{f}_c$ is modeled as a piece-wise cubic spline with L knots:

$$\boldsymbol{f}_c(x) = \sum_{i=1}^{L} c(i)B(x-i), \tag{16}$$

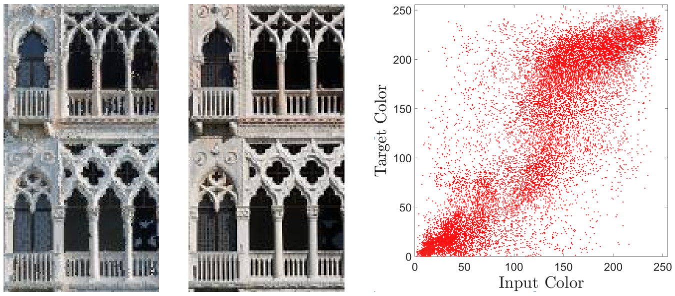
where $B(x)$ is the cubic B-spline basis function, and $c(i)$ are the B-spline coefficients.

The B-spline coefficient matrix $\mathbf{C} = (c(1), c(2), ..., c(L))^{\mathrm{T}}$ can be obtained in a least square manner:

$$\mathbf{C} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{T}, \tag{17}$$

where $\mathbf{B} \in \mathbb{R}^{|\mathcal{M}_1|\times L}$ is the B-spline basis matrix constructed using the input color values and $\mathbf{T} \in \mathbb{R}^{|\mathcal{M}_1|}$ is the target color value matrix containing the target color values in $\mathcal{M}_1$.

However, color correspondence is generally noisy as shown in Fig. 4, even when the estimated dense correspondence is accurate. The noise could be due to subtle differences in the images (for example, shadows) or intrinsically noisy image (i.e. the same input color corresponds to a few different colors on the captured image). If a B-spline curve is applied to fit the noisy color correspondence, the transfer curve will be distorted with reduced dynamic range. This will greatly affect the color transfer curve estimation.



(a) Input. (b) Target. (c) Color correspondence.

Fig. 4. An example of color correspondence: (a) Reconstructed image segment with the estimated NNF. (b) The corresponding original image segment. (c) Their color correspondence (red channel) is noisy and with many outliers. The objective is to fit a color transfer curve which can faithfully map the color in (a) to that in (b).

To further improve the global model, we adopt an EM-based algorithm for color correction as in the case of NNF interpolation. The color correspondence can be re-arranged as $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup ... \cup \mathcal{D}_{255}$ with $\mathcal{D}_m = \{(m, y_{m,i})\}_{i=1}^{N_m}$ for input color value $m = 0, ..., 255$. From the observation, $\mathcal{D}_m$ can also be modeled as a mixture of Gaussian distributed inliers and uniformly distributed outliers. We need to estimate the percentage and variance of inliers $\boldsymbol{\theta}_m = \left(\gamma_m, \sigma_m^2\right)$ for $\mathcal{D}_m$.

During the E-step, the probability of color pair $(m, y_{m,i})$ being an inlier can be estimated using:

$$p_{m,i} = \frac{\gamma_m \exp\left(-\frac{\|y_{m,i} - \boldsymbol{f}_c(m)\|^2}{2\sigma_m^2}\right)}{\gamma_m \exp\left(-\frac{\|y_{m,i} - \boldsymbol{f}_c(m)\|^2}{2\sigma_m^2}\right) + \frac{1-\gamma_m}{256}\left(2\pi\sigma_m^2\right)^{\frac{1}{2}}}. \quad (18)$$

During the M-step, $\boldsymbol{\theta}_m = \left(\gamma_m, \sigma_m^2\right)$ is updated as:

$$\begin{cases} \gamma_m & = \frac{\mathrm{tr}(\mathbf{P}_m)}{N_m}, \\ \sigma_m^2 & = \frac{(\mathbf{Y}_m - \mathbf{Z}_m)^{\mathrm{T}} \mathbf{P}_m (\mathbf{Y}_m - \mathbf{Z}_m)}{\mathrm{tr}(\mathbf{P}_m)}, \end{cases} \quad (19)$$

where $\mathbf{P}_m = \mathrm{diag}(p_{m,1}, ..., p_{m,N_m})$ is a diagonal matrix, $\mathbf{Y}_m = \left(y_{m,1}, ..., y_{m,N_m}\right)^{\mathrm{T}}$ and $\mathbf{Z}_m = \left(\boldsymbol{f}_c(m), ..., \boldsymbol{f}_c(m)\right)^{\mathrm{T}} \in \mathbb{R}^{N_m}$ are column vectors.

For each input color value $m \in \{0, ..., 255\}$, the EM algorithm is applied for $\mathcal{T}_{\text{in}}$ iterations and followed by removing the input and target color pairs with low probability being inliers (i.e. $p_{m,i} < \tau_{pc}$) from $\mathcal{D}$. When the outliers in every input color values have been removed, the color transfer curve is re-estimated with the updated color correspondence $\mathcal{D}$. After the whole process has been iterated for $\mathcal{T}_{\text{out}}$ iterations, the color transfer function is applied to image regions on $\mathcal{I}^c$ within $\mathcal{M}_0$. **Algorithm 2** summarizes our proposed color correction algorithm.

V. Numerical Results

In this section, we report the implementation details and numerical results of our proposed image completion method and compare them to other commonly used methods. Testing images are from [27].

Algorithm 2 Color Correction

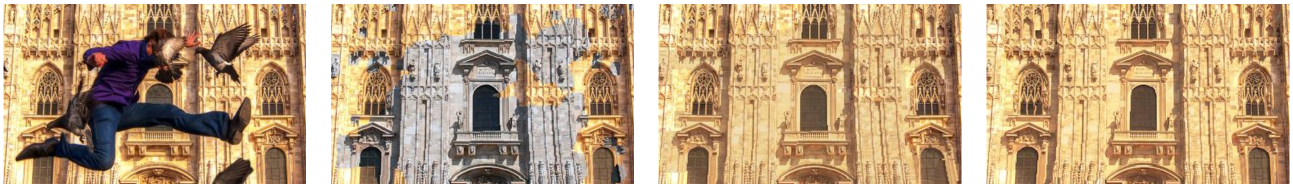
```
Input: Completed image $\mathcal{I}^c$ and masks $\{\mathcal{M}_i\}_{i=0}^{1}$ ;
Output: Color corrected image $\mathcal{O}$;
for each RGB color channel
    Fit a B-spline curve with color correspondence $\mathcal{D}$;
    for $i = 1 : \mathcal{T}_{\text{out}}$
        for $m = 0 : 255$
            for $j = 1 : \mathcal{T}_{\text{in}}$
                E-step: update posterior probability as for Eqn.(18);
                M-step: update model parameter as for Eqn.(19);
            end for
            Remove color pairs with low probability from $\mathcal{D}$;
        end for
        Fit a B-spline curve with updated $\mathcal{D}$;
    end for
    Apply color transfer curve to image region on $\mathcal{I}^c$ within $\mathcal{M}_0$ for image correction.
end for
```

A. Image Completion via Dense Correspondence

For PatchMatch, the NNF parameters (u, v, s, θ) take discrete values. The scale parameter s has been discretized into 256 discrete scales. The minimum and maximum scale is 0.33 and 3.00, respectively. The scale ratio between consecutive scales is fixed. Similarly, there are 90 discrete orientations for the orientation parameter θ. The start angle is $-45°$, and the end angle is $45°$ with the angle step being $1°$. In random search, the adaptive search threshold is set to $\boldsymbol{\tau}_{\text{rnd}} = (10, 10, 25, 9)^{\mathrm{T}}$, the edge threshold to determine edge patches is set to $\tau_{\text{edge}} = 60$. The size of the image pyramid K is the largest integer such that $2^{-K} \times \max(m, n) \geq 32$ for $\mathcal{I}^1 \in \mathbb{R}^{m \times n}$. For example, if the image size is 800×1024, then $K = 5$.

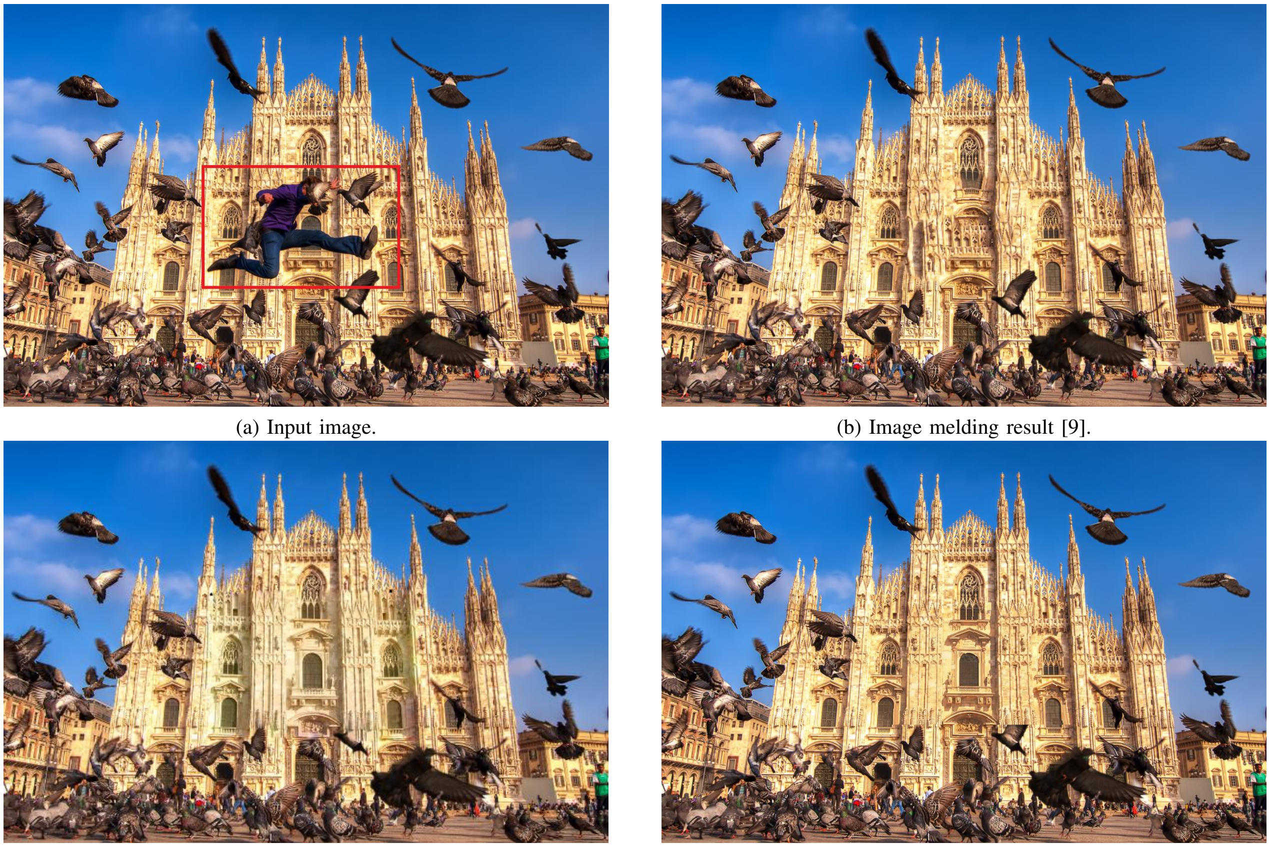
The patch size is selected as 7×7 (i.e. patch radius is $r = 3$). The length of the BRIEF descriptor is set to be $n_b = 128$. When the pixels on patch $\mathcal{E}_r^k(\mathcal{F}^k(\boldsymbol{p}))$ does not fall on a regular grid of image $\mathcal{E}^k$, bi-linear interpolation is applied to compute its pixel values. Its BRIEF descriptor is then extracted from the interpolated pixel values.

Data normalization has been applied to each column dimension of $\mathbf{X}$ and $\mathbf{Y}$ so that each parameter has zero mean and unit variance. It enables us to fix the Gaussian kernel filter range β in NNF interpolation for different "hole" sizes. The EM algorithm will stop if it does not converge within $\mathcal{T}_{\text{EM}} = 50$ iterations. When the model parameter of the previous level is not available, the percentage of inliers is initialized as $\gamma = 0.5$, otherwise the model parameter is initialized using the parameter of the previous level. The re-scaling factor for the initialization of the model parameter across pyramid level was set to $\kappa = 2$. The regularization parameter λ is set to 10. A data pair will be selected as a control point only if it is very likely an inlier (i.e. its inlier probability estimated using the previous level interpolation function exceeds $\rho = 0.99$). The threshold value for identifying mask region is set to $\tau_p = 0.7$ through cross-validation. A data point with the confidence score larger



(a) Input image segment. (b) Image completion result. (c) Color correction with B-spline. (d) Proposed color correction.

Fig. 5. Color correction comparison. (a) Image segment within the ROI. (b) Image completion result by our method before color correction. (c) Color correction result by directly fitting a B-spline curve for each color channel using the obtained color correspondence. (d) Color correction result of our proposed method. (Better view in electronic version.)



(a) Input image. (b) Image melding result [9].

(c) Zhu *et al.*'s result [27]. (d) Our completion result.

Fig. 6. Comparison with state-of-the-art image completion methods for the image taken near the Duomo of Milan. (a) Input image with a labeled ROI (red rectangles). (b) Image melding result [9]. (c) Image completion result from [27]. (d) our proposed image completion via dense correspondence result. (Better view in electronic version.)

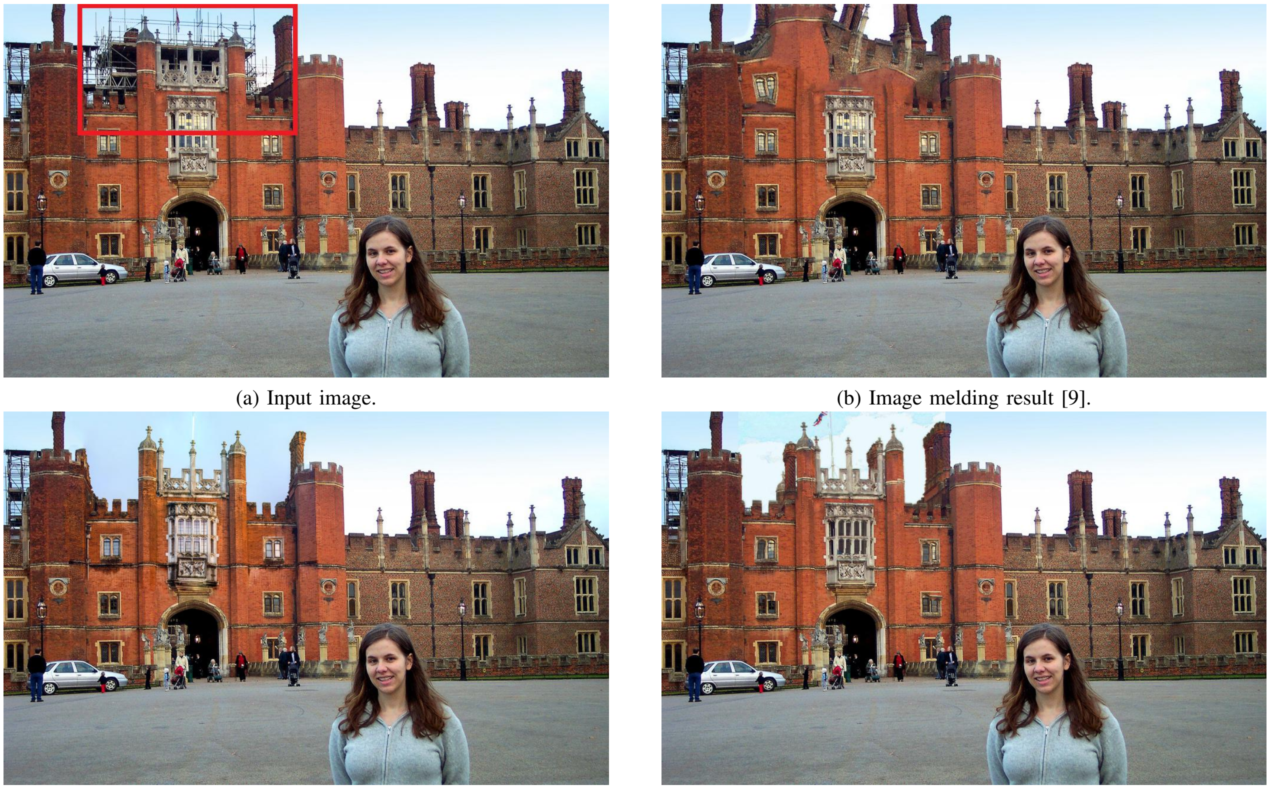
than 0.7 is considered to be reliably estimated inlier.

It is essential to have some good matching parameters found at the initial pyramid level so that these good parameters can be propagated at consecutive pyramid levels. With more iterations, a larger number of random searches will be applied to exploit the parameter space in order to increase the probability of obtaining some good matches. The total number of iterations applied at the first level was selected as 16. Each type of propagation is performed 4 times. For the other levels, the number of iteration for propagation is reduced to 4, i.e. each type propagation is performed only once.

For NNF interpolation, the number of control points M is around 0.5% of the total number of points N within the ROI. The filter range of the Gaussian kernel is selected as $\beta = 100$. There are two reasons to set the filter range to such a large value. First, the filter range β controls the smoothness of the interpolation result. As the NNF is generally noisy, a large β can remove the noisy high frequency components. Second, a small filter range is not able to interpolate the NNF within a large occluded region as the Gaussian kernel will return a very small value when the input location is far from the control points.

B. Color Correction

The B-spline for color correction has $L = 7$ knots. The maximum inner $\mathcal{T}_{\text{in}}$ and outer $\mathcal{T}_{\text{out}}$ iterations for the EM algorithm is 3 and 5, respectively. The threshold value for defining inliers is set to $\tau_{pc} = 0.7$ via cross-validation.



(a) Input image.

(b) Image melding result [9].

(c) Zhu *et al.*'s result [27].

(d) Our completion result.

Fig. 7. Comparison with state-of-the-art image completion methods for the image taken near the Hampton Palace. (a) Input image with a labeled ROI (red rectangles). (b) Image melding result [9]. (c) Image completion result from [27]. (d) our proposed image completion via dense correspondence result. (Better view in electronic version.)

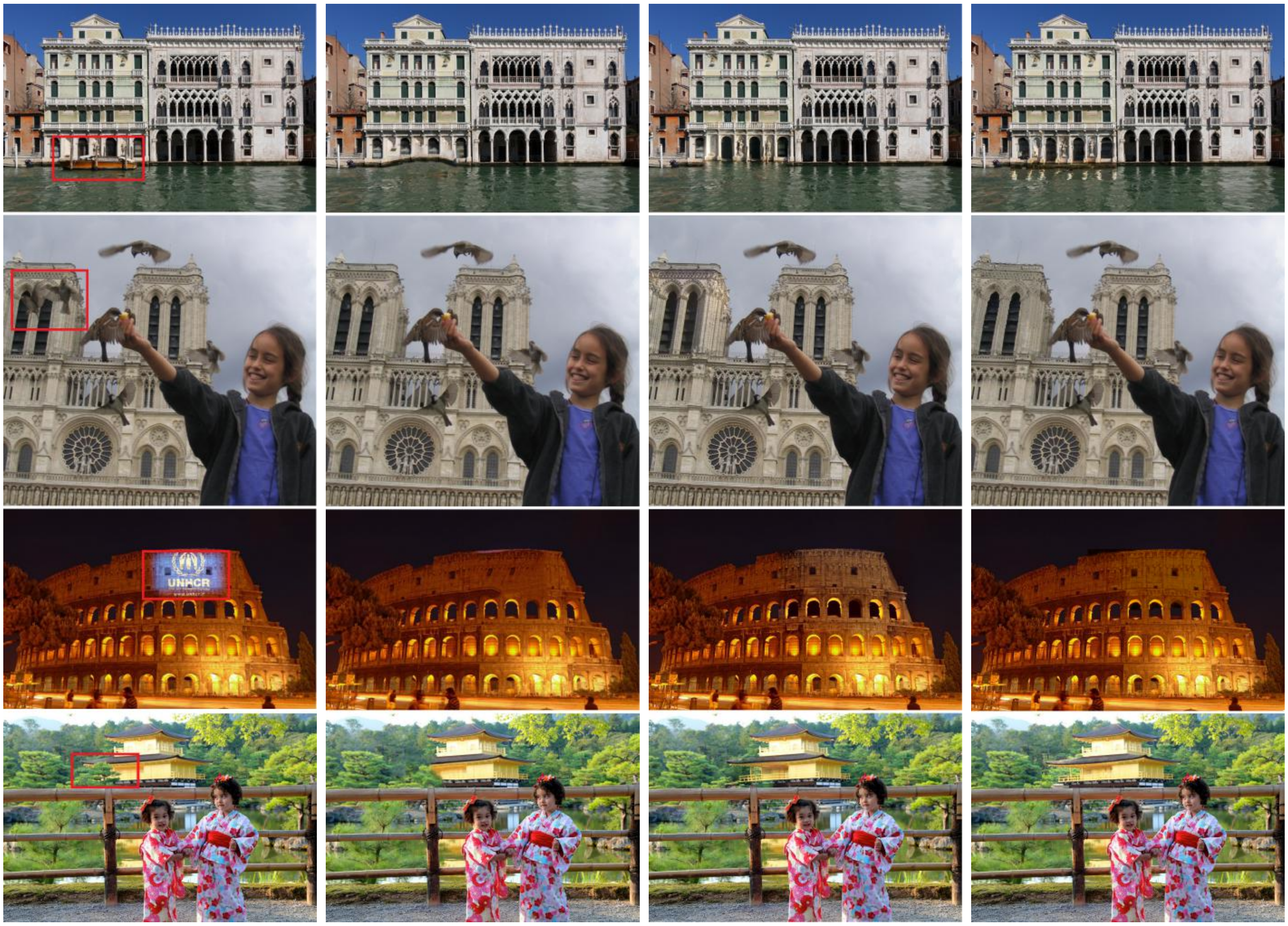
In this section, we make visual comparisons between the result without color correction, with color correction by direct B-spline fitting, and with our proposed color correction method. As shown in Fig. 5 (b), though the completed texture is highly coherent with the original image, there is an apparent color difference between the completed region and the original image in the image completion result. If a B-spline curve is fitted for every color channel using the obtained color correspondence, the color difference in 5 (c) is alleviated, however, the completed region is still a bit darker and can be identified. This is caused by using a noisy color correspondence for color correction. Fig 5 (d) shows our proposed color correction result where the completed region has no visible color difference compared to the input image in Fig 5 (a). This validates the effectiveness of our proposed color correction method for image completion.

C. Comparison with State-of-the-Art Methods

In this section, we compare our proposed image completion via dense correspondence method with state-of-the-art image completion methods [9], [27]. Image melding [9] is a single image completion method based on patch correspondence within the input image itself. Zhu *et al.*'s method [27] is a recently proposed Internet-based image completion method based on sparse and line correspondence. Image melding method requires a detailed mask to identify the unwanted image region, while Zhu *et al.*'s method and our proposed method only need a rectangle to mark ROI. The Matlab realization of image melding is from authors' website. The image completion results of [27] are provided by the authors.

Fig. 6 and Fig. 7 present two set of image completion results for detailed comparison. Fig. 8 and Fig. 9 show more comparison results. For completeness, we also show, in Fig. 11, the exemplar images used by our image completion method.

Fig. 6 (a) shows an image which is relatively easy to complete since the background is a large plane where the correspondence can be well modeled by a perspective transform. We would like to remove the stranger from the photo. In Fig. 6 (b), image melding method reconstructs a distorted image region since there are not enough repetitive image patches. This reveals the shortcoming of image completion methods based on a single image. In Fig. 6 (c), Zhu *et al.*'s method utilizes exemplar images from Internet for image completion. The completed image region is essentially coherent with the original image content in the input image. However, there is a visible miss-alignment on the top boundary between completed region and the input image. The completed image region also has a slightly different color style compared with the input image as there is no color correction performed in [27]. Fig. 6 (d) shows the image completion result by our proposed method. With dense correspondence and NNF interpolation, the completed image content is highly consistent with input image. Though the exemplar has a distinct color style as shown in Fig. 5 (b), our final image completion result



(a) Input image.

(b) Image melding [9].

(c) Zhu *et al.* [27].

(d) Proposed method.

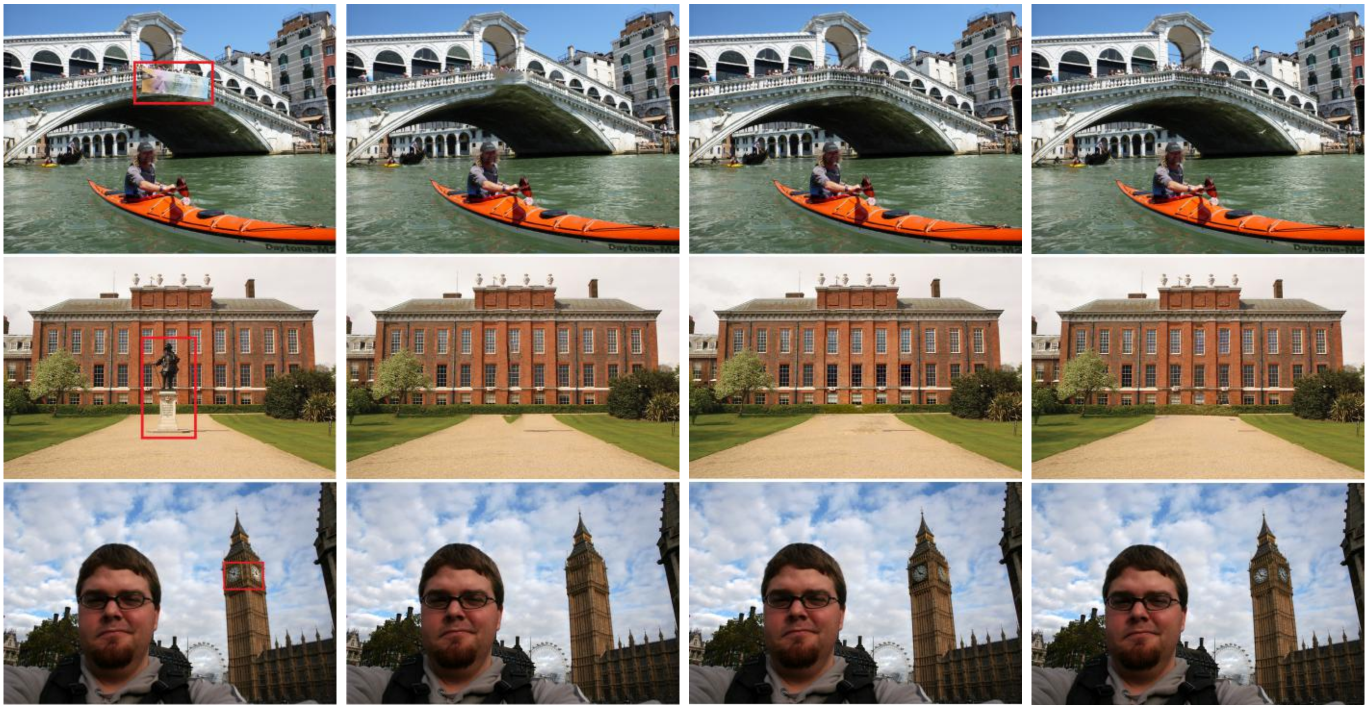
Fig. 8. Comparison with state-of-the-art image completion methods. (a) Input images with a labeled ROI (red rectangles). (b) Image melding results [9]. (c) Image completion results from [27]. (d) our proposed image completion via dense correspondence results. The input images from top to bottom are taken near Palazzo Santa Sofia, Notre-Dame de Paris, Colosseum, and Kinkaku-ji, respectively. (Better view in electronic version.)

has no visible color difference thanks to our color correction algorithm.

In Fig. 7 (a), there is a construction site on the Hampton Palace. We would like to restore its normal look. As shown in Fig. 7 (b), image melding is not able to reconstruct a natural-looking building due to the large size of the "hole" and non-stationary image content. In Fig. 7 (c), Zhu *et al.*'s method faithfully reconstructs a natural looking image. In Fig. 7 (d), our image completion result is also faithful. When making a comparison between the result in Fig. 7 (c) and Fig. 7 (d), we can find that our result has an indistinguishable color while there is a slight color difference in the result of [27]. The building in our completed result (shown in Fig. 7 (d)) has almost the same orientation as that in the input image (Fig. 7 (a)), while the building in Fig. 7 (c) is slightly tilted.

In the first row of Fig. 8, image melding still fails to reconstruct a natural looking image. Zhu *et al.*'s method faithfully reconstructs the image content behind the boat. However, the reconstructed image region seems to have a larger scale compared to the input image and has a slight miss-alignment. The image completion result by our proposed method feels more realistic. It is interesting to find that image melding method achieves a better completion result for the second input image in Fig. 8 (the image taken near the Notre-Dame de Paris). This is because the Notre-Dame de Paris is symmetric and image melding method makes use of image content from the right side. Single image based method also has the advantage of consistent color across the reconstructed image. For the input image taken near the Colosseum, image melding method completes the "hole" with slight distortion. The result by Zhu *et al.*'s method has a visible inconsistency in the color rendering. Our proposed method reconstructs the content within the ROI realistically. In the last row, Zhu *et al.*'s method and our method have similar completion results. Three further examples are shown in Fig. 9 where we can appreciate a behaviour of the three algorithms similar to what seen in Fig. 8. Image melding tends to struggle when the region to be completed is large and there is a lack of similar patches in the image. It performs well otherwise. The other two methods, i.e. [27] and ours, perform consistently well with our method often providing a more consistent rendering.

The data-driven methods of [21], [22] are powerful and



(a) Input image. (b) Image melding [9]. (c) Zhu *et al.* [27]. (d) Proposed method.

Fig. 9. Comparison with state-of-the-art image completion methods. (a) Input images with labeled ROI (red rectangles). (b) Image melding results [9]. (c) Image completion results from [27]. (d) our proposed image completion via dense correspondence results. The input images from top to bottom are taken near Rialto Bridge, Kensington Palace, and Big Ben, respectively. (Better view in electronic version.)

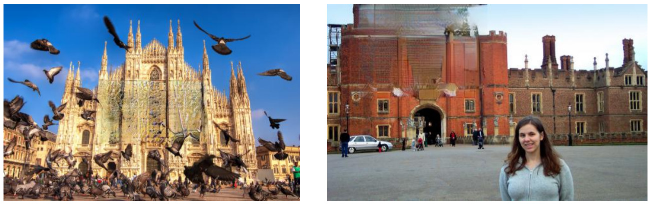


Fig. 10. Sample image completion results by [22].

Fig. 11. The exemplar images used for image completion.

are able to learn a single and generally effective model for image completion from a large dataset. In Fig 10, sample image completion results using [22] are shown. We conjecture that the unsatisfactory results can be due to the inconsistency between the training dataset and the testing image. Our proposed method is a model-based approach so it does not suffer from potentional training/testing mismatches and the prior information (i.e. the smoothness of the dense correspondence and the color transfer curve) is important to provide the much better image completion results shown in the paper. We also note that our proposed method relies on the use of "side information" provided by the retrieved exemplar images, whereas the other two approaches do not need an exemplar image. This also clarify why we perform much better.

D. Subjective Evaluation for Image Completion

In order to perform subjective evaluation, we have surveyed 34 people about their preference over different image completion results. Each respondent was asked to make a pairwise comparison for 9 pairs of image completion results. The 9 pairs of image completion results are randomly selected from the results of image melding (method 1), Zhu *et al.*'s method (method 2), and our proposed method (method 3) shown in Fig. 6 - Fig. 9. For method i and j, with $i, j = 1, 2, 3$ and $i \neq j$, we have obtained 102 pairwise comparison results. This gives us a winning matrix $\Xi \in \mathbb{R}^{3 \times 3}$:

$$\Xi = \begin{bmatrix} 0 & 0.95 & 0.83 \\ 0.05 & 0 & 0.75 \\ 0.17 & 0.25 & 0 \end{bmatrix}, \tag{20}$$

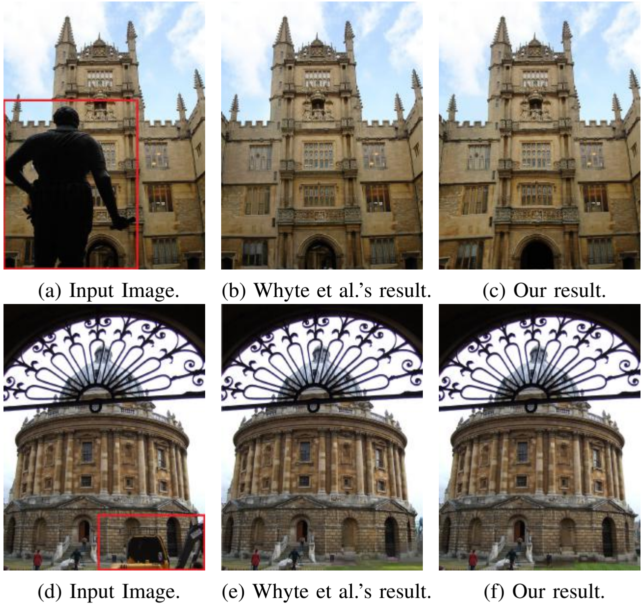
where $\Xi(i, j)$ indicates the fraction of respondents that think the result of method j is better than that of method i with $i, j \in \{1, 2, 3\}$.

From the winning matrix Ξ, we can find that about 83% and 75% of the respondents think the image completion results of our proposed method are better than those of image melding and Zhu *et al.*'s method, respectively. There are also some interesting results from the survey. We thought that

Test Image	Duomo	Hampton Palace	Palazzo Santa Sofia	Notre Dame	Colosseum	Kinkaku-ji	Rialto Bridge	Kensington Palace	Big Ben
ROI Size	81.4	68.0	35.1	19.5	48.0	30.5	27.9	82.5	3.2
Time (s)	27.58	64.34	13.28	10.91	44.19	10.08	9.01	31.57	5.76

TABLE I

COMPUTATION TIME (S) OF THE PROPOSED ALGORITHM. THE ROI SIZE IS THE NUMBER OF PIXELS IN THE RECTANGLE ROI IN 10^3 PIXELS.


(a) Input Image. (b) Whyte et al.'s result. (c) Our result.

(d) Input Image. (e) Whyte et al.'s result. (f) Our result.

Fig. 12. Further image completion results evaluated on the images from [25].

image melding achieves the best result for the Notre-Dame de Paris case, while most respondents think Zhu *et al.*'s method produces a better result. This could be due to a brighter color and more detailed structure in the result of Zhu *et al.*'s method. Another example is that most respondents prefer our result in the Big Ben case over Zhu *et al.*'s. This may be due to a slightly miss-alignment on the Zhu *et al.*'s result.

E. Computation Complexity and Further Examples

Table I shows the computation time of our proposed image completion method. We have implemented our method using C++. The evaluation is performed on a PC with Intel Core i7 3.4 GHz CPU. From Table I, the computation time of our image completion algorithm is generally linear with the number of pixels in the ROI region, while may fluctuate depending on the complexity of the image content. To show that our method works well also in different conditions, we have also performed image completion on two images from [25]. The results are shown in Fig. 12.

VI. CONCLUSIONS

In this paper, we have proposed a novel Internet-based image completion method. Instead of correlating the input image with the retrieved exemplar image based on sparse correspondence, we propose to make use of dense correspondence to relate every pixel within the ROI to a pixel on the exemplar image. This enables accurate image content transfer from the exemplar image and reliable color correction to remove color difference between the completed part and the input image.

A hierarchical framework is built to perform dense correspondence estimation, image completion, and color correction jointly. The dense correspondence estimation is based on a hierarchical variation of the PatchMatch method. Contrary to other approaches, BRIEF descriptor is adopted to handle the inter image differences. As the estimated NNF is in general noisy and with large occlusion within the "hole", an EM-based method is applied to identify reliable inliers from the estimated dense correspondence and interpolate a smooth NNF over the occluded region. A global model is applied for color correction. As a noisy color correspondence is observed, color correction is also based on a similar EM algorithm to remove outlier color correspondences. To demonstrate our method, we made a comparison with the state-of-the-art image completion methods. From the numerical results, we can see that our proposed image completion method achieves photo realistic results on a wide range of images.

ACKNOWLEGMENT

The authors would like to thank the anonymous reviewers who helped improve the quality of this manuscript.

REFERENCES

[1] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, 2014.

[2] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[3] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[4] C.-W. Fang and J.-J. J. Lien, "Rapid image completion system using multiresolution patch-based directional and nondirectional approaches," *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2769–2779, 2009.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[6] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *European Conference on Computer Vision*. Springer, 2010, pp. 29–43.

[7] T.-H. Kwok, H. Sheung, and C. C. Wang, "Fast query for exemplar-based image completion," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3106–3115, 2010.

[8] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153–1165, 2010.

[9] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image melding: Combining inconsistent images using patch-based synthesis." *ACM Trans. Graph.*, vol. 31, no. 4, pp. 82–1, 2012.

[10] O. Le Meur, M. Ebdelli, and C. Guillemot, "Hierarchical super-resolution-based inpainting," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3779–3790, 2013.

[11] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2423–2435, 2014.

[12] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 129, 2014.

[13] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[14] S. Gepshtein and Y. Keller, "Image completion by diffusion maps and spectral relaxation," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 2983–2994, 2013.

[15] C. Barnes, F.-L. Zhang, L. Lou, X. Wu, and S.-M. Hu, "Patchtable: efficient patch queries for large datasets and applications," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 97, 2015.

[16] K. He and J. Sun, "Computing nearest-neighbor fields via propagation-assisted kd-trees," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 111–118.

[17] S. Korman and S. Avidan, "Coherency sensitive hashing," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1607–1614.

[18] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Transactions on Graphics (ToG)*, vol. 24, no. 3, pp. 861–868, 2005.

[19] Y. Liu and V. Caselles, "Exemplar-based image inpainting using multiscale graph cuts," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1699–1711, 2013.

[20] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[22] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 107, 2017.

[23] H. Amirshahi and S. Kondo, "An image completion algorithm using occlusion-free images from internet photo sharing sites," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 91, no. 10, pp. 2918–2927, 2008.

[24] J. Hays and A. A. Efros, "Scene completion using millions of photographs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 4.

[25] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! internet-based inpainting," in *Proceedings of the 20th British Machine Vision Conference, London*, 2009.

[26] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Photo uncrop," in *European Conference on Computer Vision*. Springer, 2014, pp. 16–31.

[27] Z. Zhu, H.-Z. Huang, Z.-P. Tan, K. Xu, and S.-M. Hu, "Faithful completion of images of scenic landmarks using internet images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 8, pp. 1945–1958, 2016.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM transactions on Graphics (TOG)*, vol. 30, no. 4, p. 70, 2011.

[30] J. Lu, H. Yang, D. Min, and M. N. Do, "Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1854–1861.

[31] H. Yang, W.-Y. Lin, and J. Lu, "Daisy filter flow: A generalized discrete approach to dense correspondences," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 3406–3413.

[32] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5704–5712.

[33] C. Barnes and F.-L. Zhang, "A survey of the state-of-the-art in patch-based synthesis," *Computational Visual Media*, vol. 3, no. 1, pp. 3–20, 2017.

[34] J. Hur, H. Lim, C. Park, and S. C. Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1392–1400.

[35] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2307–2314.

[36] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[37] C. Zhang, C. Shen, and T. Shen, "Unsupervised feature learning for dense correspondences across scenes," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 90–107, 2016.

[38] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1385–1392.

[39] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Deepmatching: Hierarchical deformable dense matching," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323, 2016.

[40] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *arXiv preprint arXiv:1504.06852*, 2015.

[41] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4015–4023.

[42] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European conference on computer vision*. Springer, 1994, pp. 151–158.

[43] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[44] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[45] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE conference on*. Ieee, 2012, pp. 510–517.

[46] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[47] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[48] J. Ma, J. Zhao, J. Tian, X. Bai, and Z. Tu, "Regularized vector field learning with sparse approximation for mismatch removal," *Pattern Recognition*, vol. 46, no. 12, pp. 3519–3532, 2013.

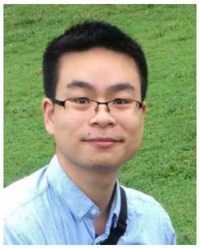
[49] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1706–1721, 2014.

[50] C. A. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural computation*, vol. 17, no. 1, pp. 177–204, 2005.

[51] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *European conference on computer vision*. Springer, 2002, pp. 21–31.

[52] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 123–137, 2007.

[53] O. Frigo, N. Sabater, V. Demoulin, and P. Hellier, "Optimal transportation for example-guided color transfer," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 655–670.



Jun-Jie Huang (S'17) received the B.Eng. (Hons.) degree with First Class Honours in Electronic Engineering from The Hong Kong Polytechnic University in 2013, and the M.Phil degree in Electronic and Information Engineering from the same university in 2015. He is currently pursuing the Ph.D. degree with Electrical and Electronic Engineering Department, Imperial College London, U.K. His research interests include signal and image processing, inverse problems, dictionary learning and deep learning.

Pier Luigi Dragotti (M'02–SM'11–F'17) is Professor of Signal Processing in the Electrical and Electronic Engineering Department at Imperial College London and a Fellow of the IEEE. He received the Laurea Degree (summa cum laude) in Electronic Engineering from the University Federico II, Naples, Italy, in 1997; the Master degree in Communications Systems from the Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland in 1998; and PhD degree from EPFL in 2002. He has held several visiting positions. In particular, he was a visiting student at Stanford University, Stanford, CA in 1996, a summer researcher at Bell Labs, Lucent Technologies, NJ in 2000 and a visiting scientist at Massachusetts Institute of Technology (MIT) in 2011. Before joining Imperial College in November 2002, he was a senior researcher at EPFL working on distributed signal processing for sensor networks for the Swiss National Competence Center in Research on Mobile Information and Communication Systems.

Prof. Dragotti was Technical Co-Chair for the European Signal Processing Conference in 2012, Associate Editor of the IEEE Transactions on Image Processing from 2006 to 2009 and an Elected Member of the IEEE Image, Video and Multidimensional Signal Processing Technical Committee. He was also the recipient of an ERC Consolidator Award for the project RecoSamp. He is currently Editor-in-Chief of the IEEE Transactions on Signal Processing, a member of the IEEE Signal Processing Theory and Methods Technical Committee and a member of the IEEE Special Interest Group on Computational Imaging.

His research interests include sampling theory, wavelet theory and its applications, sparsity-driven signal processing with applications in image enhancement, neuroscience and fields estimation using sensor networks.