AUTOMATIC IMAGE COMPLETION WITH STRUCTURE PROPAGATION AND TEXTURE SYNTHESIS

XIAOWU CHEN*, BIN ZHOU†, FANG XU and QINPING ZHAO

State Key Laboratory of Virtual Reality Technology and Systems
School of Computer Science and Engineering
Beihang University, Beijing, P. R. China
*chen@buaa.edu.cn
†zhoubin@vrlab.buaa.edu.cn

In this paper, we present a novel automatic image completion solution in a greedy manner inspired by a primal sketch representation model. Firstly, an image is divided into structure (sketchable) components and texture (non-sketchable) components, and the missing structures, such as curves and corners, are predicted by tensor voting. Secondly, the textures along structural sketches are synthesized with the sampled patches of some known structure components. Then, using the texture completion priorities decided by the confidence term, data term and distance term, the similar image patches of some known texture components are found by selecting a point with the maximum priority on the boundary of hole region. Finally, these image patches inpaint the missing textures of hole region seamlessly through graph cuts. The characteristics of this solution include: (1) introducing the primal sketch representation model to guide completion for visual consistency; (2) achieving fully automatic completion. The experiments on natural images illustrate satisfying image completion results.

Keywords: Image processing; image completion; image inpainting; texture synthesis.

1. Introduction

1.1. Objectives

Every once in a while, we hope to erase something from our daily photographs, such as a garbage truck right in the middle of charming Beijing National Stadium, an ex-boyfriend in a family photo, a political ally in a group portrait who has fallen out of favor. Other times, there is simply missing data in some areas of the images, such as an aged corner of an old photograph, a hole in an image-based 3D reconstruction due to occlusion, a bug on the camera lens. Image completion, also called image inpainting, which is a well-know and well-studied topic in computer vision and graphics realm, is the task of filling in or replacing the hole region mentioned in above scenarios with the available information from their surroundings. The aim

†Corresponding author.
of this task is to ensure that the modification of completed image cannot be perceptible. Let $\Lambda$ be the region of a given input image $I$ and $\Lambda_u$ be the region marked to be filled, called the hole (unknown or missing or mask) region. Image completion is to fill in $\Lambda_u$ using the information from known region $\Psi = \Lambda - \Lambda_u$ [1, 2].

Based on our observation, the key to image completion issue is to employ a suitable image representation, image prior model and efficient inference algorithm. Thus, the completion process needs a suitable image representation framework to guide the rendering process of hole regions. Most image completion methods treat the hole region as a Markov Random Field (MRF) and fill it using standard texture synthesis techniques. Textures can be modeled as a realization of a local and stationary random process. Thus, each pixel of a texture image can be characterized by a small set of spatially neighboring pixels. However, for a general image, each pixel cannot simply be predicted from a small set of local neighboring pixels, and we cannot assume each pixel is independent of the rest of the image. So, to achieve visually consistent results, this paper presents an integrated framework to complete images based on the primal sketch representation model [3], which guided the image completion process.

1.2. Related work

In the literature, there were generally three main approaches dealing with the image completion issue: structure-based methods, exemplar-based texture synthesis methods, and methods combining texture synthesis and structure reconstruction.

Through representing the image as piecewise continuous functions, the structure-based methods converted the filling of hole region to an estimation process of explicit functions. Some structure-based methods, inspired by the partial derivation equation (PDE), tried to diffuse progressively the information from the boundary of hole region [4–7], and filled in the hole region by propagating linear structures via diffusion. Bertalmio et al. [4] first introduced image inpainting, and filled the hole region by propagating image Laplacians continuously in isophote directions using PDE, which called BSCB algorithm. However, these PDE-based methods often resulted in oversmooth synthesis when the filling pixels are full of texture. Some other structure-based methods based on the variation model, which involved a criterion of regularity of the reconstruction. These variational methods [8–13] usually derived the Euler-Lagrange equation for image and solved the implicit equation via Level Set [14] or tensor voting [15] after constructing the extremum of energy function with the variation model. In [16], a method called Curvature-Driven Diffusions (CDD) was proposed to handle curve structures by incorporating Euler’s elastica. With the total variation (TV) algorithm raised by Rudin et al. [7], Chan et al. [8, 9] presented a TV image inpainting method, which defined a TV image model and its energy through bounded variation (BV). Thus the method computed the Euler-Lagrange equation of energy function by variational principle, and solved the equation in three image channels using Level Set [14], respectively. This method was suitable
for non-texture noise image. The tensor voting algorithm [15] estimated the missing curves of hole region according to the structural information of existing curves, which kept the continuity in tangent directions, and completed the junctions of multiple curves.

These structure-based methods worked at the pixel level, and performed well for small or thin gap filling, such as speckles, scratches, and text. However, when the hole region is large, blurring artifacts introduced by the diffusion process are noticeable. On the other hand, PDE based methods interpolated structure parts of image by defusing local operators or minimizing global curve smooth energies, whereas the minimization of variational energies was time-consuming and often got stuck into local minimal. Nevertheless Euler elastica energies only completed curves of circle shapes but failed at square shapes, and the tensor voting algorithm took into account the information in the neighborhood of hole regions and also curvature information, so it implicitly embeds general shape priors in its energy terms.

Texture synthesis, such as FRAME [17, 18], Heeger [19], is very important to computer vision and image processing realms. The aim of texture synthesis is to synthesize the texture with the given sample texture under certain similarity of statistical significance. Through finding the pixel sets that have the nearest neighborhoods to the source pixel and constructing the conditional probability density, Efros [20] present a non-parametric texture synthesis algorithm based on pixel sampling. Considering the time complexity, Liang [21] proposed the texture synthesis algorithm based on patch (a rectangle area of image) sampling. Inspired by texture synthesis techniques, another group of approaches performed inpainting based on pure texture synthesis to solve the oversmooth problem of structure-based methods. Amongst these approaches, the exemplar-based methods, such as Criminisi et al. [22, 23], were more effective, which generated new textures by sampling and copying texture patterns from the source [1]. These exemplar-based methods formulated the inpainting as a problem of sampling from a non-parametric one dimensional conditional MRF (FRAME or Gibbs) density, and propagated hole region by copy-and-paste from a single image [20, 22, 23, 1], or a large dataset [24].

Recent exemplar-based methods also incorporated structure information [5, 25, 26, 2]. Several papers have been concerned with the combination of a PDE model (or a variational model) for the reconstruction of structure together with a greedy exemplar-based algorithm for the reconstruction of texture. In [5, 2], the input image was decomposed into texture components and structure components and carried out image completion in the two components. The output was the sum of the two completed components. With the user manually drawing structural lines as guidance in hole region, Sun et al. [25] present an interactive system to perform exemplar based inpainting along the edges and the rest of hole region, and achieved the state-of-the-art results.

Most exemplar-based methods mainly took a greedy way to fill the hole region, finding the best patches locally. Contrary to the greedy approaches, some papers preferred globally approaches [27], which formulated the image completion as
discrete global optimization problem. The global optimization problem usually can be solved using belief propagation (BP) \[25, 28–30\]. A patch transform algorithm was performed by Cho \[28\], which modeled the image as higher-order MRF and solved the MRF using BP with restraint among patches. In \[30\] image completion was automatically solved using an efficient BP. Wexler’s global optimization algorithm \[31\] can achieve better fusion performance in boundary between hole region and known region using expectation maximization (EM) like optimization scheme. But it is sensitive to initial solution and easy to get local optimal solution. The ShiftMap algorithm \[32\] transformed the image completion into a graph labeling problem, and used graph cuts \[33\] to obtain global optimization, whose limitation is computational time.

1.3. Contributions

This paper presents a novel automatic image completion solution in a greedy manner based on a primal sketch representation model \[3\], which automatically completes salient structures and texture of the hole region. Firstly, given an input image, its primal sketch graph is computed using the primal sketch representation model \[3\], and the given image is divided into structure (sketchable) components and texture (non-sketchable) components. Afterwards, 3-degree-junctions such as T-junctions, Y-junctions around the hole region are detected by Wu’s algorithm \[34\]. Secondly, based on testing the compatibility between any two terminators of the 3-degree-junctions with Elastica \[35\], the missing structures, such as curves and corners, are predicted by tensor voting \[36\], and the structural sketch completion is completed. Thirdly, the textures along structural sketches are synthesized with the optimal sampled patches of some known structure components, which are computed by Belief Propagation algorithm \[37\]. The filled textures along structural sketches can guide the texture synthesis process of the remaining hole region. Fourthly, after structure completion, using the texture completion priorities decided by the confidence term, data term and distance term, the similar image patches of some known texture components are found by selecting a point with the maximum priority on the boundary of hole region. Finally, these image patches inpaint the remaining missing textures of hole region seamlessly through exemplar-based texture synthesis. Instead of copying the whole selected patch \[23\], the proposed solution only pastes an optimal portion of a patch to the hole region, and the optimal seam is determined by graph cuts algorithm \[38\]. The characteristics of this solution include: (1) introducing the primal sketch representation model to guide completion for visual consistency; (2) achieving fully automatic completion. The experiments on natural images illustrate satisfying image completion results and show above characteristics.

1.4. Organization

The remainder of this paper is organized as following. Section 2 formulates the image completion and introduces the overall framework of our solution. Sections 3
2. Formulation and Framework Overview

2.1. Problem formulation

Given a defined image lattice $\Lambda$ with hole region $\Lambda_u$, the goal of image completion is to infer an idea image $I$ based on the partial observation $I_{o|\Lambda\setminus\Lambda_u}$.

According to Bayesian framework, image completion can be defined as maximize the posterior probability of $p(I|I_o, \Lambda)$:

$$p(I|I_o, \Lambda) = \frac{p(I_o|I, \Lambda)p(I, \Lambda)}{p(I_o, \Lambda)} \propto p(I_o|I, \Lambda)p(I, \Lambda).$$

Essentially, the key to the image completion issue is to employ a suitable image prior model $p(I, \Lambda)$ or define a set of energy functions which are the logarithm likelihood functions of the posterior probability $p(I|I_o, \Lambda)$. This paper adopts the primal sketch model [3] as image prior, which is defined as follows:

$$p(I, S) = \frac{1}{Z} \exp \left\{ -\sum_{i=1}^{n} \sum_{(x,y) \in \Lambda_{str}, i} \frac{1}{2\sigma^2} (I(x,y) - B_i(x,y|\theta_i))^2 - \gamma_{str}(S_{str}) ight. \\
- \sum_{j=1}^{m} \sum_{(x,y) \in \Lambda_{tex}, j} \sum_{k=1}^{K} \phi_{j,k}(F_k*I(x,y)) - \gamma_{tex}(S_{tex}) \right\},$$

where $S$ is a set of pixels of discontinuity regions of given image which contains $S_{str}$ and $S_{tex}$, $B_i(x,y|\theta_i), i = 1, \ldots, n$ is a set of sparse coding functions representing edge and ridge segments $\Lambda_{str}$ in the image, $\theta_i$ are geometric and photometric parameters of these coding functions, and $F_k, k = 1, \ldots, K$ are texture filters applied on the pixels in $\Lambda_{tex}$.

The computed primal sketch graph divides the image lattice into structure domain $S_{str}$ and texture domain $S_{tex}$, respectively. Image intensities on the structure domain $S_{str}$ are represented by sparse coding functions with explicit geometric parameters (such as normal directions) and photometric parameters. These parameters provide prior information for structure completion of the image completion process. The MRF model is assumed to model image intensities on the texture domain $S_{tex}$, and the texture completion can be achieved by finding similar neighborhoods and synthesizing the missing pixels.

The energy function $E(X)$ is defined for searching optimal image patches from known regions to fill the structure and texture components of the hole region. The optimal sampled labels $X = \{x_i\}^L_i=1$ are obtained by minimizing $E(X)$:

$$E(X) = E_{structure} + E_{texture},$$

and 4 present the detailed framework implementation of our solution from structure completion and texture completion aspects. Experiment results and conclusions are shown in Secs. 5 and 6, respectively.
where the energy $E$ consists of structure energy value $E_{\text{structure}}$ and texture energy value $E_{\text{texture}}$. Meanwhile the structure energy value $E_{\text{structure}}$ is defined as:

$$E_{\text{structure}} = E_{\text{elastic}} + \sum_{i \in \Lambda_{u,sk}} \{ E_1(x_i) + E_2(x_i) \}$$

where $E_{\text{elastic}}$ constrains the curve connection process, $\nu$ and $\alpha$ are constant, $\Gamma$ is the curve, $\kappa$ is the curvature function $\kappa(s), s \in [0, L]$ and $L$ are the arc-length. $\Lambda_{u,sk}$ is the structure part of hole region $\Lambda_u$, $d(c_i, c_{x_i})$ is the sum of shortest distance between all points in segment $c_i$ and $c_{x_i}$, $c_i$ is located on the connected curves of the hole region, and $c_{x_i}$ is located on the detected curves of the known region. $E_2(x_i)$ constrains the synthesized patches on the boundary of hole region to match well with the known pixels, which is the normalized squared differences of the overlap region between image patches and known region.

In Eq. (3), the texture energy value $E_{\text{texture}}$ is:

$$E_{\text{texture}} = \sum_{(i,j) \in \Lambda_{u,nstk}} E_3(x_i + x_j),$$

where $\Lambda_{u,nstk}$ is the texture part of the hole region $\Lambda_u$, and $E_3(x_i + x_j)$ encodes the texture consistence constraint between two adjacent pixels $i$ and $j$.

### 2.2. Framework overview

According to the formulation of image completion issue, we present a novel automatic image completion solution in a greedy manner based on a generative image model [2], inspired by primal sketch representation. The framework of the solution includes automatic structure propagation and texture synthesis for completion process, as illustrated in Fig. 1.

Proposed in Marr’s book [39], primal sketch is supposed to be a symbolic and perceptually lossless representation of the input image. In the framework, given an input image $I$ with hole region, its primal sketch graph $G$ whose vertices are image primitives is computed by a mathematical model for primal sketch [3], as shown in Fig. 1(b). This sketch graph $G$ divides the image lattice into the sketchable (structure) components in Fig. 1(c) and the non-sketchable (texture) components in Fig. 1(e). The sketchable components are modeled using sparse coding, and the non-sketchable components are stochastic texture modeled by MRF model. The pixels on the sketchable components are reconstructed through the primitives under some transformations. Meanwhile the pixels on the non-sketchable components are synthesized from the MRF model regarded the sketchable components as boundary condition. Based on the division above, the solution needs to complete the structure parts and texture parts of hole region both.

As Fig. 1(d) shows, 3-degree-junctions such as T-junctions, Y-junctions around the hole region are firstly detected [34]. Based on the compatibility between any
Fig. 1: (Color online) The framework overview of automatic image completion. (a) The input image and marked hole region. (b) Sketch graph computed by primal sketch algorithm. (c) Structure part of the input image. (d) Structure completion result. (e) Texture part of the input image. (f) Texture completion process. (g) The final completion result.

two terminators of the 3-degree-junctions [35], the tangents of the missing structure lines are explicitly predicted through a tensor field diffusion process [36]. With the predicted missing structures, such as curves and corners, the structural sketches are completed. Then, the textures along structural sketches are synthesized with the optimal sampled patches, which are computed under some known structure components by Belief Propagation algorithm [37]. Figure 1(d) shows the structure completion result. The filled textures along structural sketches can guide the following texture completion process.

After structure completion, this solution uses the confidence term, data term and distance term to decide the texture completion priorities. A texture completion priority computing algorithm is designed to select a point with the maximum priority on the boundary of hole region at each inpainting step and control the order of texture inpainting process. At each step, the point with the maximum priority on the hole boundary is selected as the filling point. To find the best exemplar and reduce the time complexity of exemplar matching, a segmentation algorithm is utilizing for finding adjacent known region, which may have homogenous textures with the hole region, and sampling optimal exemplar for each highest priority pixel in known regions. The similarity of two image patches is measured by the normalized Sum of Squared Differences (SSD). Figure 1(f) shows the texture completion process.

In Fig. 1(g), to achieve more visual pleasing result, the remaining missing textures of hole region are patched up seamlessly using exemplar-based texture synthesis with the sampled optimal image patches. Instead of copying the whole selected patch [23], the proposed solution only pastes an optimal portion of the patch to
the hole region, and the optimal seam is determined by graph cuts algorithm [38]. Figure 1(g) shows the final completion result.

The implementation of the framework needs to complete the missing structure parts, and then synthesis the texture parts of the hole region. Thus, structure completion and texture completion, two essential aspects of the framework, are described in the following sections in detail (Secs. 3 and 4).

3. Structure Completion

3.1. Structure sketch completion

Structure sketch detection. On the basis of primal sketch division, the edges of the input image are detected using gPb algorithm [37]. As the gPb produces proposal edge map only, the structure information are obtained by tracing the map using Canny’s hysteresis thresholding. With the result of edge detection, the solution finds long continuous sketches (curves) around the hole region automatically. 3-degree-junctions of the sketches (curves) and the boundary of the hole region are detected by Wu’s algorithm [34], as illustrated in Fig. 2(a).

Structure sketch matching. As illustrated in Fig. 2(b), 3-degree-junctions are broken into a set of terminators as open bonds. These open bonds need to find a match by testing compatibilities between any two terminators on appearance cues and geometric properties. Given two terminators $a_i$ and $a_j$, the contour to be completed between them, denoted by $\Gamma^*$, is decided by minimizing the Elastica cost function (as shown in Eq. (6)) in a contour space $\Omega^*_\Gamma$ [35]:

$$\Gamma^* = \arg \min_{\Gamma \in \Omega^*_\Gamma} \int_\Gamma [(v_1 + \alpha_1 \kappa_1^2) + (v_2 + \alpha_2 \kappa_2^2)] ds,$$

(6)

where $\kappa_1$ is the curvature of $a_1$, $\kappa_2$ is the curvature of $a_2$, $v_1$ and $v_2$ are constants, $\alpha_1$ and $\alpha_2$ are scalable coefficients.

Fig. 2: (Color online) Structure sketch completion. (a) 3-degree-junctions (such as T-junctions, Y-junctions) of long continuous sketches and the boundary of the hole region. (b) Terminators. (c) Curve connection.
**Tensor voting.** After deciding the terminators matches, this solution predicts the missing structures, such as curves and corners, by tensor voting. Firstly, the tangents (normal directions) of every existing curve points are calculated using linear regression and least square method according to Eq. (7).

\[ Y = kX + b \]
\[ k = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sum(X - \bar{X})^2} = \frac{l_{XY}}{l_{XX}} = \frac{\sum XY - \frac{1}{n} \sum X \sum Y}{\sum X^2 - \frac{1}{n} \sum X^2}, \quad b = \bar{Y} - k\bar{X} \]

where \( k \) is the coefficient of regression, \( b \) is the intercept, and \( n \) is the number of sampled points in the existing curve.

Secondly, these points with the normal directions are represented as 2D stick tensors \[ T = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T = (\lambda_1 - \lambda_2) e_1 e_1^T + \lambda_2 (e_1 e_1^T + e_2 e_2^T), \]

where the tensor \( T \) is a 2-by-2 matrix, \( \lambda_1, \lambda_2 \) are the eigenvalues and \( \lambda_1 > \lambda_2 > 0 \), \( e_1, e_2 \) are the eigenvectors, \( \lambda_1 - \lambda_2 \) defines the saliency or certainty of curve. In this solution: \( \lambda_1 = 1, \lambda_2 = 0, e_2 = \vec{0}, \) and \( e_1 \) is normal vector of the point on the existing curve.

Thirdly, the existing curve points, called voters, can vote a saliency for the receivers which are the other points of the input image. The saliency affects the normal direction of the receiver. The saliency decay function has the following form (Eq. (9)):

\[ DF(s, \kappa, \sigma) = e^{-\frac{s^2+c\kappa^2}{2\sigma^2}}, \]

where \( s \) is the arc length of the osculating circle between the receiver and voter, \( \kappa \) is the curvature, \( c \) controls the degree of decay with curvature, \( \sigma \) is the scale of voting, which determines the effective neighborhood size. Given a voter, the local tensor voting field is calculated with its 2D stick tensors according to Eq. (9), and the neighbor points in the scale of voting receive a direction effect and an energy effect. The direction effect is determined by the line between the voter and receiver. The energy effect defines the saliency or certainty in the direction effect. The local tensor voting fields are as shown in Fig. 3, where \( \sigma = 54 \), the neighborhood size is \( 2 \times 54 + 1 = 109 \), and the colors changing from red to blue illustrate the saliency energy value from high to low.

Finally, according to the local tensor voting fields, each point in the hole region (blue point in Fig. 4) obtains a curve saliency value \( \lambda_1 - \lambda_2 \) after receiving and accumulating the collected votes from the existing curve points (voters) in the known regions of the image. The tensor voting process is illustrated in Fig. 4. The saliency map in the hole region is formed with the accumulation of curve saliency values.

**Non-maxima suppression.** Finding the local maximum in the curve saliency map may allow us to figure out the best curve while using the surrounding context of
Fig. 3: (Color online) The local tensor voting fields of four directions, where $\sigma = 54$, and the neighborhood size is 109. The center point of each figure is the voter, and the other points are receivers. The colors changing from red to blue illustrate the saliency energy value from high to low.

Fig. 4: (Color online) Curve connection by tensor voting. The black lines are existing curves in known regions, and each point plays the role of voters. The rectangle demonstrates the boundary of hole region, and the blue points are receivers. The saliency map of hole region is achieved with the accumulation of curve saliency value of blue points.

Each pixel to determine whether or not the curve passes there. The process of taking the maximum in the normal direction first identifies the normal axis for each tensor point in the image. Once it is done, the likelihoods are checked along the sides of the current pixel to see if it is in fact a maximum. We apply a local maxima algorithm...
Fig. 5: (Color online) The non-maxima suppression during curve connection by tensor voting. With the saliency map, the points with the highest curve saliency are selected as the curve points for the missing curve.

in local area on the salience map to identify edges of the hole region. So, For each position \( x_i \), only the point with the highest curve saliency is selected as the curve point \( P_{x_i} \):

\[
P_{x_i} = \max \{ P_{x_i,y} (\lambda_1 - \lambda_2), 1 \leq j \leq S \},
\]

where \( x_i, y_i \) is the image coordinates, \( S \) is the sample density (in pixels). The result of structural sketch completion with non-maxima suppression is illustrated in Fig. 5 and Fig. 2(c).

3.2. Structure texture completion

To complete the textures along structural sketches, a Belief Propagation algorithm similar to Sun [25, 37] is employed, which finds the optimal labels for each sampled point on the connected curves by minimizing the energy \( E_{\text{structure}} \) in Eq. (4), as illustrated in Fig. 6. The filled textures along structural sketches can guide the following texture completion process of the remaining hole region.

Fig. 6: (Color online) Synthesize textures along structural sketches using Belief Propagation. The white rectangle demonstrates the hole region, and the orange red points are the sampled points on the connected curve. The blue patches are sampled along the existing curves by Belief Propagation.
4. Texture Completion

After structure completion, there are still large hole regions to be filled. The equivalence of the Julesz ensemble and FRAME model [17, 18] states that texture synthesis can be done without necessarily learning MRF model. We adopt this strategy and sample from a subset of the Julesz ensemble by pasting texture patches from the sampled texture (exemplars).

To achieve more visual pleasing result, the textures of the remaining hole region are filled by the graph cuts texture synthesis algorithm [38]. Before applying the graph cuts based texture synthesis to image completion issue, this solution uses the confidence term, data term and distance term to decide the texture completion priorities. Then a texture completion priority computing algorithm is designed to select a point with the maximum priority on the boundary of hole region at each inpainting step and control the order of texture inpainting process. At each step, the point with the maximum priority on the hole boundary is selected as the filling point.

To find the best exemplar and reduce the time complexity of exemplar matching, this solution runs a segmentation algorithm to find adjacent known region which may have homogenous textures around the hole region, and samples optimal exemplar for each highest priority pixel in known regions. The similarity of two image patches is measured by the normalized Sum of Squared Differences (SSD).

With the sampled optimal image patches, the remaining textures of hole region are patched up seamlessly using exemplar-based texture synthesis. Instead of copying the whole selected patch [23], the proposed solution only pastes an optimal portion of a patch to the hole region, and the optimal seam is determined by graph cuts algorithm [38].

4.1. Completion priority

Filling order is crucial to a greedy texture completion method. Thus far, the default favorite was proposed by Criminisi [22], where the completion priority is decided the confidence term and data term. This solution further constrains the completion priority by adding a distance term to ensure that hole region is filled from the outside inward.

After some filling step, some hole regions remain unknown while other hole regions have been inpainted. It is needed to find a filling point with the maximum priority, and the priority is decided by the confidence term, data term and distance term. The distance term is calculated using Felzenszwalb’s distance transform algorithm [40]. The original boundary of hole region has the distance energy of 0, and other pixels have distance energy greater than 0.

This solution performs above task through a best-first filling algorithm that depends entirely on the priority values that are assigned to each patch on the fill front. The priority computation is biased toward those patches which are on the
continuation of strong edges, surrounded by high-confidence pixels and nearer to
the boundary of hole region. For the pixel \( p \) located on the boundary of hole region \( \Omega \), \( \Psi_p \) is the image domain centered at the point \( p \), and some parts of \( \Psi_p \) is known while the others not.

\[
P(p) = C(p)D(p)Dt(p) \quad \text{where} \quad C(p) = \frac{\sum_{q \in \Psi_p \cap (I-\Omega)} C(q)}{|
\Psi_p|}, \quad D(p) = \frac{|
abla I_p \cdot n_p|}{\alpha},
\]

where the priority \( P(p) \) at \( p \) is the product of \( C(p) \), \( D(p) \) and \( Dt(p) \) together. \( C(p) \) and \( D(p) \) are the confidence term and the data term, respectively. \( Dt(p) \) is the distance term. \( I \) defines the input image with hole region \( \Omega \). \( p \) is located on the boundary of hole region \( \Omega \). \( \Psi_p \) is the image domain centered at the point \( p \). \( q \) is a point in the \( \Psi_p \). Confidence term \( C(p) \) may be thought of as a measure of the amount of reliable information surrounding \( p \). \( \nabla I_p \) is a vector of isophotes. \( n_p \) is a unit vector orthogonal to the boundary at point \( p \). Data term \( D(p) \) is a function of the strength of isophotes hitting the boundary at each iteration. Distance term \( Dt(p) \) is calculated using Felzenszwalb’s distance transform algorithm \[40\], which ensure that pixels nearer to the original hole boundary are filled with higher priority.

### 4.2. Graph cuts texture synthesis

To achieve more visual pleasing result, the textures of the remaining hole regions are filled by the graph cuts texture synthesis algorithm \[38\]. First we adopt the graph cuts segmentation algorithm of Felzenszwalb \[40\] with the parameters \( \sigma = 0.5 \), \( k = 500 \), \( \min = 20 \). This can often group homogenous regions of an image together, and produce reasonable oversegmentations with fewer superpixels (typically less than 100 for a 800 \times 600 image). Hole region is always adjacent to some known regions, and these adjacent regions form a region adjacent graph. The proposed solution only needs to search in these adjacent regions for best image patches to synthesis the texture of the remaining hole regions. This search strategy can improve the reliability and efficiency of texture synthesis of the remaining hole regions.

To synthesis the textures of remaining hole regions, this solution samples optimal patch for each highest priority pixel in known regions. The similarity of two image patches is measured by the normalized Sum of Squared Differences (SSD). Unlike Criminisi et al. \[22\], which copy the whole patch to the hole region at a time, we only copy an optimal portion of the patch. The portion of the patch to be copied is determined by graph cuts texture synthesis algorithm of Kwatra \[38\]. It usually produces more visually pleasing results than copying and pasting directly. Let \( s \) and \( t \) be two adjacent pixels in the overlap region between two adjacent image patches \( A \) (or \( x_i \)) and \( B \) (or \( x_j \)). Let \( A(s) \) (or \( x_i(s) \)) and \( B(s) \) (or \( x_j(s) \)) be the colors at
Fig. 7: (Color online) Graph cuts algorithm for computing optimal patch portion. The red indicates the image patch $A$ and the blue is image patch $B$. The orange demonstrates the overlay area between $A$ and $B$. The blue line shows the minimum cut computing by graph cuts algorithm.

the position $s$ in the patch $A$ and $B$. The consistent matching cost $M$ between the two adjacent pixels $s$ and $t$ that are copied from patches $x_i$ and $x_j$, respectively is:

$$E_3(x_i, x_j) = M(s, t, A, B) = \sum_{s, t \in \text{Overlap}} M(x_i, x_j, s, t)$$

$$= \sum_{s, t \in \text{Overlap}} ((A(s) - B(s))^2 + (A(t) - B(t))^2)$$

$$= \sum_{s, t \in \text{Overlap}} ((x_i(s) - x_j(s))^2 + (x_i(t) - x_j(t))^2). \tag{12}$$

We use CIE $Lab$ color space to compute the color difference. As Fig. 7 shows, the nodes of the graph are the overlap pixels $B$ and remaining pixels, and the arcs connecting the adjacent pixel nodes are labeled with their matching cost $M(s, t, A, B)$. The blue line shows the minimum cut, and it means that pixels $1, 4, 5, 7, 8$ must come from patch $A$, and pixels $2, 3, 6, 9$ come from patch $B$.

5. Experimental Results

To demonstrate the performance of our automatic solution, we apply it on a number of natural images where the hole regions are covered by masks. In addition, we
Fig. 8: (Color online) Some typical results of automatic image completion. The first two columns show the input images and marked hole regions. The third column shows the results of automatic structure completion. The last column shows the final results. Better to view on screen.

compare our automatic solution to Criminisi’s method [23], which is a greedy image completion method and similar to ours. We do not compare with [25] since it needs manually added structure information.

Figure 8 shows a few representative results by the proposed solution. The first two columns show the input images and marked hole regions. The third column shows the results of automatic structure completion. The last column shows the final results. As shown in Fig. 8, our automatic greedy solution can produce visually natural results without obvious artifacts. Both structures (such as stairs and the horizons) and textures (such as the sea, roadway and tree) are handled very well in the images with satisfying visual consistency.

Figure 9 gives the comparative results on three example images, where we want to erase some objects for example something in the river, the unpopular people besides the parterre, two girls standing on the bridge. The first column shows the original images. The second column shows the results obtained by the greedy
algorithm in [23]. From these results, we can find some obvious artifacts, and moreover some strong structures (such as the corner of the parterre) cannot be propagated well. The last column is the results of our automatic solution. We successfully propagate the structure and synthesize the texture.

More completion results are provided in Figs. 10 and 11. All these results illustrate the satisfying completion and the characteristics of our solution.
Fig. 10: (Color online) More image completion results where we want to remove the objects, such as the outland players in a baseball match, the front-runner in an athletics game, and the visual inconsistent girl, from the input images. The first column shows the original images. The second column is the results of our automatic solution.
Fig. 11: (Color online) Another image completion results. The laggard on the battlefield, the zebra bursting on the scene and the tourists photographed accidentally would be erased from the three input images. The first column shows the original images. The second column is the results of our automatic solution.
6. Conclusion and Future Works

This paper has presented a novel automatic image completion solution that automatically fills structure and texture components inspired by primal sketch model. Firstly, an image is divided into structure components and texture components, and this solution predicts the missing structures, such as curves and corners, by tensor voting. Secondly, the textures along structural sketches are synthesized with the sampled patches of some known structure components. Then, with the texture completion priorities decided by the confidence term, data term and distance term, the similar image patches of some known texture components are found by selecting a point with the maximum priority on the boundary of hole region. Finally, this solution patches up the remaining textures of hole region seamlessly with these image patches using graph cuts algorithm. Experiment results on the natural images show our characteristics.

However, this solution is still in the preliminary research stage. Nearly 90% time is spending on similar image patches searching, and common image patches searching method such as Sum of Squared Differences (SSD) is computationally-insensitive. It is considerable to choose histogram features as the descriptor of the sampled image patches and apply a nearest neighbor search method, such as kd-tree, for accelerative exemplar matching. On the other hand, although most image completion methods are based on a greedy strategy, we also seek to explore the appealing global solutions.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (90818003&60933006), National High Technology (863) and Key Technology R&D Program of China (2009AA01Z331&2008BAH29B02), and Specialized Research Fund for the Doctoral Program of Higher Education (20091102110019). We would like to thank Dr. Liang Lin for helpful suggestions, and Kai Jiang for data processing.

References


