

# Automatic Objects Removal for Scene Completion

Jianjun Yang  
Department of Computer Science  
University of North Georgia  
Oakwood, GA 30566, USA  
Email: jianjun.yang@ung.edu

Yin Wang  
Department of Mathematics and  
Computer Science  
Lawrence Technological University  
Southfield, Michigan 48075, USA  
Email: ywang12@ltu.edu

Honggang Wang  
Department of Electrical  
and Computer Engineering  
University Massachusetts Dartmouth, USA  
Email: hwang1@umassd.edu

Kun Hua  
Department of Electrical and  
Computer Engineering  
Lawrence Technological University  
Southfield, Michigan 48075, USA  
Email: khua@ltu.edu

Wei Wang  
Department of Electrical Engineering  
and Computer Science  
South Dakota State University, USA  
Email: wei.wang@ieee.org

Ju Shen  
Department of Computer Science  
University of Kentucky  
Lexington, KY 40506, USA  
Email: jushen.tom@uky.edu

**Abstract**—With the explosive growth of web-based cameras and mobile devices, billions of photographs are uploaded to the internet. We can trivially collect a huge number of photo streams for various goals, such as 3D scene reconstruction and other big data applications. However, this is not an easy task due to the fact the retrieved photos are neither aligned nor calibrated. Furthermore, with the occlusion of unexpected foreground objects like people, vehicles, it is even more challenging to find feature correspondences and reconstruct realistic scenes. In this paper, we propose a structure-based image completion algorithm for object removal that produces visually plausible content with consistent structure and scene texture. We use an edge matching technique to infer the potential structure of the unknown region. Driven by the estimated structure, texture synthesis is performed automatically along the estimated curves. We evaluate the proposed method on different types of images: from highly structured indoor environment to the natural scenes. Our experimental results demonstrate satisfactory performance that can be potentially used for subsequent big data processing: 3D scene reconstruction and location recognition.

**Keywords:** Image Completion, Texture Synthesis, Online Photos, Scene Reconstruction, Object Removal

## I. INTRODUCTION

In the past few years, the massive collections of imagery on the Internet have inspired a wave of work on many interesting big data topics: scene reconstruction, location recognition, and online sharing of personal photo streams [2] [3] [4]. For example, one can easily download a huge number of photo streams associated with a particular place. By using features (e.g. SIFT), it is possible to automatically estimate correspondence information and reconstruct 3D geometry for the scene [5] [6]. Imagine building a world-scale location recognition engine from all of the geotagged

images from online photo collections, such as Flickr and street view databases from Google and Microsoft. However, it is a challenging task as the photo streams are neither aligned nor calibrated since they are taken in different temporal, spatial, and personal perspectives. Furthermore, with the occlusion of unexpected foreground objects, it is even more difficult to recover the whole scene or accurately identify overlapping regions between different photos.

To resolve the above issue, image in-painting is an effective solution. In this paper, we propose an automatic object removal algorithm for scene completion, which benefits subsequent large imagery processing. The core of our method is based on the structure and texture consistency. Our proposed approach has two major contributions. First, we develop a curve estimation approach to infer the potential structure of the occluded region on the image. Second, an orientated patch matching algorithm is designed for texture propagation. Our work has a broad range of applications including image localization [7] [8], privacy protection [9] [10] [11], and other network based applications [12] [13] [14] [15] [16].

## II. RELATED WORKS

In the literature, image completion or in-painting has been intensively studied: in [17], Efros and Leung used a one-pass greedy algorithm to render unknown pixels based on the assumption that the probability distribution of the pixel's brightness is independent to the rest of the image when the spatial neighborhood is given. In [18], the authors proposed an example-based approach to fill in the missing regions. It worked well in filling in small gaps but not in large ones. The weakness of such approach is that it fails to preserve the potential structures. Jia *et al.* [19] designed

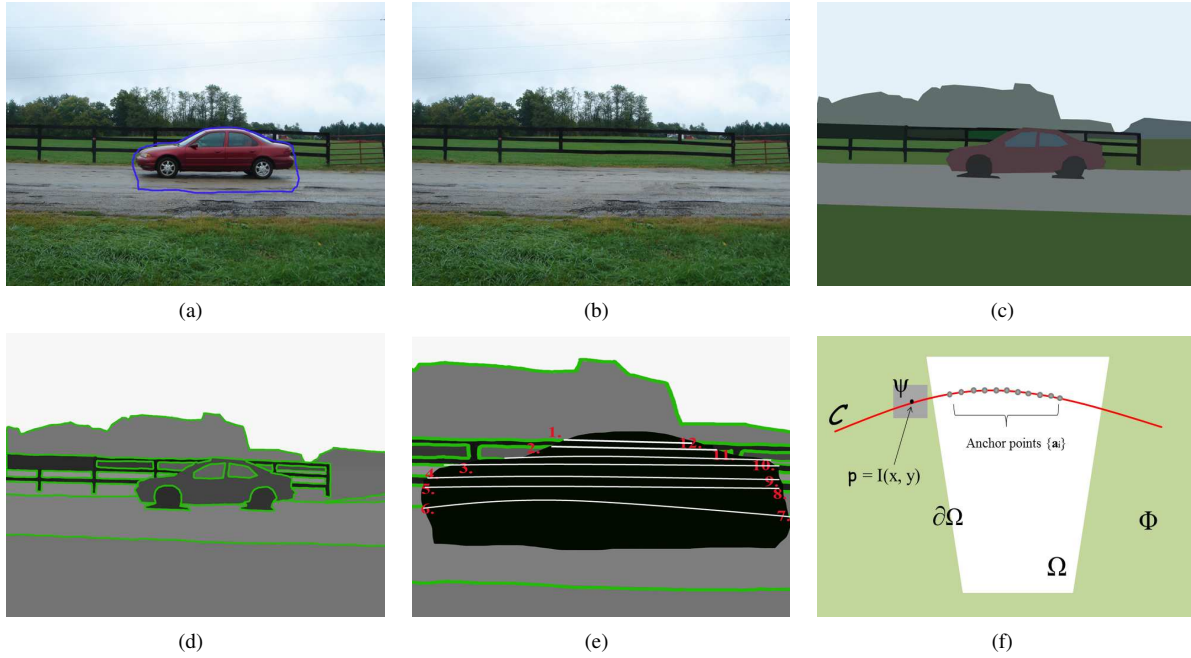


Fig. 1. Scene recovery by removing specified foreground object (a) Original Image (b) Our result (c) Contour detection by using *OWT-UCM* method [1] (d) Edge extraction (e) Structure generation in the occlusion region by identifying corresponding edge pairs (f) Some denotations in our algorithm.

an image in-painting method based on texture-segmentation and tensor-voting that created smooth linking structures in the occluded regions. This method sometimes introduces noticeable artifact due to the texture inconsistency. Criminisi *et al.* [20] made an improvement by assigning in-painting orders based on the edge strength levels. Their algorithm used a confidence map and the image edges to determine the patch completion priority. However, the structures in the resulting images are not well preserved. The method in [21] produced a better result via structure propagation, while this approach requires more interaction. The completion results largely depend on the animator's individual technique. Some other existing work also explored in [22] [23] [24].

### III. OUR APPROACH

The process of our framework is: for a given image, users specify the object for removal by drawing a closed contour around it. The enclosure is considered an unknown region that is inferred and replaced by the remaining region of the image. Figure 1(a) shows an example: the red car is selected as the removing object. In the resulting image Figure 1(b), the occluded region is automatically recovered based on the surrounding environment.

First let us define a set of notations for the rest of our paper. For an image  $\mathbf{I}$ , the target region for in-painting is denoted as  $\Omega$ ; the remaining part of the image is denoted as  $\Phi (= \mathbf{I} - \Omega)$ , which is also known as source region. The boundary contour along  $\Omega$  is denoted as  $\partial\Omega$ . A pixel's value is represented by  $p = \mathbf{I}(x, y)$ , where  $x$  and  $y$  are the 2D

coordinates on the image. The surrounding neighborhood centered at  $(x, y)$  is often called as a patch, denoted as  $\Psi_p$ . The coordinates of pixels inside the patch  $\Psi_p$  should be in the range:  $[x \pm \Delta x, y \pm \Delta y]$ . These concepts are illustrated in Figure 1(f). In our framework, there are three phases involved to achieve the scene recovery: structure estimation, structure propagation, and remaining part filling.

#### A. Structure Estimation

In this phase, we estimate the potential structure in  $\Omega$  by finding all the possible edges. This procedure can be further decomposed into two steps: *Contour Detection in  $\Phi$*  and *Curve Generation in  $\Omega$* .

1) *Contour Detection in  $\Phi$* : We first segment the region  $\Phi$  by using *gPb Contour Detector* [25]. It is based on the idea of computing the oriented gradient signal  $G(x, y, \theta)$  on the four channels of its transformed image: brightness, color  $a$ , color  $b$  and texture channel.  $G(x, y, \theta)$  is the gradient signal, where  $(x, y)$  indicates the center location of the circle mask that is drawn on the image and  $\theta$  indicates the orientation. The *gPb Detector* is composed of two important components: *mPb Edge Detector* and *sPb Spectral Detector* [25]. We apply linear combination on *mPb* and *sPb* (factored by  $\beta$  and  $\gamma$ ) according to the gradient ascent on F-measure:

$$gPb(x, y, \theta) = \beta \cdot mPb(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \quad (1)$$

Thus a set of edges in  $\Phi$  can be retrieved via *gPb*. However, these edges are not in close form and have classification

ambiguities. To solve this problem, we use the *Oriented Watershed Transform* [25] and *Ultrametric Contour Map* [1] (*OWT-UCM*) algorithm to find the potential contours by segmenting the image into different regions. The output of *OWT-UCM* is a set of different contours  $\{C_i\}$  and their corresponding boundary strength levels  $\{L_i\}$  as Figure 1(c) shows.

2) *Curve Generation in  $\Omega$* : After obtaining the contours  $\{C_i\}$  from the above procedure, salient boundaries in  $\Phi$  can be found by traversing  $\{C_i\}$ . Our method for generating the curves in  $\Omega$  is based on the assumption: for the edges on the boundary in  $\Phi$  that intersects with the  $\partial\Omega$ , it either ends inside  $\Omega$  or passes through the missing region  $\Omega$  and exits at another point of  $\partial\Omega$ . Below is our algorithm for identifying the curve segments in  $\Omega$ :

---

**Algorithm III.1 Identifying curve segments in  $\Omega$**

---

**Require:** Construct curve segments in  $\Omega$ .

**Ensure:** The generated curves have smooth transition between known edges.

- 1: Initial  $t = 1.0$
  - 2: For  $t = t - \Delta t$
  - 3:   if  $\exists e \in \{C\} : E \cap \partial\Omega \neq \emptyset$
  - 4:     Insert  $e$  into  $\{E\}$
  - 5: End if  $t < \delta_T$
  - 6: Set  $t = t_0$ , retrieve all the contours in  $\{C_i\}$  with  $L_i > t$
  - 7: Obtain  $\langle \phi_{x1}, \phi_{x2} \rangle$  for each  $E_x$
  - 8: **DP** on  $\langle \phi_{01}, \phi_{02} \rangle, \langle \phi_{11}, \phi_{12} \rangle, \dots$  to find optimal pairs from the list.
  - 9: According to the optimal pairs, retrieve all the corresponding edge-pairs:  $\{(E_{x1}, E_{x2}), (E_{x3}, E_{x4}), \dots\}$ .
  - 10: Compute a transition curve  $C_{st}$  for each  $(E_s, E_t)$ .
- 

In algorithm III.1, it has three main parts: (a) collect all potential edges  $\{E_x\}$  in  $\Phi$  that hits  $\partial\Omega$ ; (b) identify optimal edge pairs  $\{(E_s, E_t)\}$  from  $\{E_x\}$ ; (c) construct a curve  $C_{st}$  for each edge pair  $(E_s, E_t)$ .

**Edges Collection:** The output of *OWT-UCM* are contours sets  $\{C_i\}$  and their corresponding boundary strength levels  $\{L_i\}$ . Given different thresholds  $t$ , one can remove those contours  $C$  with weak  $L$ . Motivated by this, we use the *Region-Split* scheme to gradually demerge the whole  $\Phi$  into multiple sub-regions and extract those salient curves. This process is carried out on lines 1-9: at the beginning the whole region  $\Phi$  is considered as one contour; then iteratively decrease  $t$  to let potential sub-contours  $\{C_i\}$  faint out according the boundary strength; Every time when any edges  $e$  from the newly emerged contours  $\{C\}$  were detected of intersecting with  $\partial\Omega$ , they are put into the set  $\{E\}$ .

**Optimal Edge Pairs:** the reason of identifying edge pairs is based on the assumption if an edge is broken up by  $\Omega$ , there must exist a pair of corresponding contour edges in  $\Phi$  that intersect with  $\partial\Omega$ . To find the potential pairs  $\{(E_s, E_t)\}$  from the edge list  $\{E_x\}$ , we measure the corresponding enclosed regions similarities. The neighboring

regions  $\langle \phi_{x1}, \phi_{x2} \rangle$  which is partitioned by the edge  $E_s$  are used to compare with the corresponding regions of another edge  $E_t$ . This procedure is described on lines 7–9 of the algorithm III.1. Each neighboring region is obtained by lowering down the threshold value  $t$  to faint out more detailed contours as Figure 1(d) shows.

To compute the similarity between regions, we use the *Jensen-Shannon divergence* [26] method that works on the color histograms:

$$d(H_1, H_2) = \sum_{i=1}^n \{H_1^i \cdot \log \frac{2 \cdot H_1^i}{H_1^i + H_2^i} + H_2^i \cdot \log \frac{2 \cdot H_2^i}{H_2^i + H_1^i}\} \quad (2)$$

where  $H_1$  and  $H_2$  are the histograms of the two regions  $\phi_1, \phi_2$ ;  $i$  indicates the index of histogram bin. For any two edge  $(E_s, E_t)$ , the similarity between them can be expressed as:

$$M(E_s, E_t) = \frac{\|L_s - L_t\|}{L_{\max}} \cdot \min\{d(H_{si}, H_{ti}) + d(H_{sj}, H_{tj})\} \quad (3)$$

$i$  and  $j$  are the exclusive numbers in  $\{1, 2\}$ , where 1 and 2 represent the indices of the two neighboring regions in  $\phi$  around a particular edge. The  $L_{\max}$  is the max value of the two comparing edges' strength levels. The first multiplier is a penalty term for big difference between the strength levels of the two edges. To find the optimal pairs among the edge list, dynamic programming is used to minimize the global distance:  $\sum_{s,t} M(E_s, E_t)$ , where  $s \neq t$  and  $s, t \in \{0, 1, \dots, \text{size}(\{E_i\})\}$ . To enhance the accuracy, a maximum constraint is used to limit the regions' difference:  $d(H_1, H_2) < \delta_H$ . If the individual distance is bigger than the pre-specified threshold  $\delta_H$ , the corresponding region matching is not considered. In this way, it ensures if there are no similar edges existed, no matching pairs would be identified.

**Generate Curves for each  $(E_s, E_t)$**  : we adopt the idea of fitting the clothoid segments with polyline stoke data first before generating a curve [27]. Initially, a series of discrete points along the two edges  $E_s$  and  $E_t$  are selected, denoted as  $\{p_{s0}, p_{s1}, \dots, p_{sn}, p_{t0}, p_{t1}, \dots, p_{tm}\}$ . These points have a distance with each other by a pre-specified value  $\Delta_d$ . For any three adjacent points  $\{p_{i-1}, p_i, p_{i+1}\}$ , the corresponding curvature  $k_i$  could be computed according to [28]:

$$k_i = \frac{2 \cdot \det(p_i - p_{i-1}, p_{i+1} - p_i)}{\|p_i - p_{i-1}\| \cdot \|p_{i+1} - p_i\| \cdot \|p_{i+1} - p_{i-1}\|} \quad (4)$$

Combining the above curvature factors, a sequence of polyline are used to fit these points. The polylines are expected to have a possibly small number of line segments while preserving the minimal distance against the original data. Dynamic programming is used to find the

most satisfied polyline sequence by giving a penalty for each additional line segment. A set of clothoid segments can be derived corresponding to each line segment. After a series of rotations and translations over the clothoid, a final curve  $\mathcal{C}$  is obtained by connecting each adjacent pair with  $G^2$  continuity [27]. Figure 1(e) demonstrates the curve generation result.

### B. Structure Propagation:

After the potential curves are generated in  $\Omega$ , a set of texture patches, denoted as  $\{\Psi_0, \Psi_1, \dots\}$ , need to be found from the remaining region  $\Phi$  and placed along the estimated curves by overlapping with each other with a certain proportion. Similar to [21], an energy minimization based method is proposed in a *Belief Propagation* (BP) framework. However, we have different definitions for the energy and message passing functions. The details are in the algorithm III.2.

---

#### Algorithm III.2 BP Propagation Algorithm

---

**Require:** Render the texture for each patch  $\Psi_i$  in  $\Omega$  along the estimated structures.

**Ensure:** Find the best matching patches while ensuring the global coherence and consistency.

- 1: For each curve  $\mathcal{C}$  in  $\Omega$ , define a series of *anchor points* on it,  $\{\mathbf{a}_i, |i = 1 \rightarrow n\}$ .
  - 2: Collect exemplar-texture patches  $\{\hat{\Psi}_{t_i}\}$  in  $\Phi$ , where  $t_i \in [1, m]$
  - 3: Setup a factor graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  based on  $\{\mathcal{C}\}$  and  $\{\mathbf{a}_i\}$
  - 4: Defining the energy function  $\mathbf{E}$  for each  $\mathbf{a}_i$ :  $\mathbf{E}_i(t_i)$ , where  $t_i$  is the index in  $[1, M]$ .
  - 5: Defining the message function  $\mathbf{M}_{ij}$  for each edge  $\mathcal{E}$  in  $\mathcal{G}$ , with initial value  $\mathbf{M}_{ij} \leftarrow 0$
  - 6: Iteratively update all the messages  $\mathbf{M}_{ij}$  passed between  $\{\mathbf{a}_i\}$
  - 7:  $\mathbf{M}_{ij} \leftarrow \min_{\mathbf{a}_i} \{\mathbf{E}_i(t_i) + \mathbf{E}_{ij}(t_i, t_j) + \sum_{k \in \mathcal{N}(i), k \neq j} \mathbf{M}_{ki}\}$
  - 8: end until  $\Delta \mathbf{M}_{ij} < \delta, \forall i, j$  (by Convergence)
  - 9: Assign the best matching texture patch from  $\{\hat{\Psi}_{t_i}\}$  for each  $\mathbf{a}_i$  that  $\arg \min_{[T, R]} \{\sum_{i \in \mathcal{V}} \mathbf{E}_i(t_i) + \sum_{(i, j) \in \mathcal{E}} \mathbf{E}_{ij}(t_i, t_j)\}$ . Here  $T$  is an  $n$  dimensional vector  $[t_1, t_2, \dots, t_n]$ , where  $i \in [1, n]$ ;  $R$  is also an  $n$  dimensional vector  $[r_1, r_2, \dots, r_n]$  with each element representing the orientation of source patch  $\hat{\Psi}_{t_i}$ .
- 

In the algorithm, the *anchor points* are evenly distributed along the curves with an equal distance from each other  $\Delta d$ . These points represent the center where the patches  $\{\Psi_i\}$  ( $l \times l$ ) are synthesized, as shown in Figure 1(f). In practice, we define  $\Delta d = \frac{1}{4} \cdot l$ . The  $\{\hat{\Psi}_{t_i}\}$  is the source texture patches in  $\Phi$ . They are chosen on from the neighborhood around  $\partial\Omega$ . For the factor graph building, we consider each  $\mathbf{a}_i$  as a vertex  $\mathcal{V}_i$  and  $\mathcal{E}_{ij} = \mathbf{a}_i \mathbf{a}_j$ , where  $i, j$  are the two adjacent points.

In previous works [21] [20], each  $\Psi_i$  have the same orientation as  $\hat{\Psi}_{t_i}$ , which limits the varieties in the source texture. Noticing that different patch orientations could produce different results, we introduce a scheme called *Adaptive Patch* by defining a new formulas for  $\mathbf{E}$  and  $\mathbf{M}$  in the structure propagation.

Traditionally, the node energy  $\mathbf{E}_i(t_i)$  is defined as the *Sum of Square Difference*(SSD) by comparing the known pixels in each patch  $\Psi_i$  with the candidate corresponding portion in  $\hat{\Psi}_{t_i}$ . But this method limits the salient structure directions. Instead of using SSD on the two patches, a series of rotations are performed on the candidate patch before computing the similarity. Mathematically,  $\mathbf{E}_i(t_i)$  can be formulated as:

$$\mathbf{E}_i(t_i) = \alpha_\lambda \cdot P \cdot \sum \|\Psi_i - \hat{R}(\theta) \cdot \hat{\Psi}_{t_i}\|_\lambda^2 \quad (5)$$

Where  $\hat{R}$  represents different rotations on the patch  $\hat{\Psi}_{t_i}$ . Since the size of a patch is usually small, the rotation can be specified with an arbitrary number of angles. In our experiment, it is specified as  $\theta \in \{0, \pm\frac{\pi}{4}, \pm\frac{\pi}{2}, \pi\}$ . The parameter  $\lambda$  represents the number of known pixels in  $\Psi_i$  that overlap with the rotated patch  $\hat{\Psi}_{t_i}$ .  $P$  is a penalty term, the more number of overlapping pixels, the higher of similarity is assigned. So we use  $P$  to discourage the patches with smaller number of sharing pixels. Here, the percentage is expressed as  $P = \frac{\lambda}{l^2}$  ( $l$  is the length of  $\Psi$ ).  $\alpha_\lambda$  is the corresponding normalized scalar. Thus the best matching patch  $\hat{\Psi}$  is represented by two factors: index  $t_i$  and rotation  $R_i$ .

In a similar way, the energy  $\mathbf{E}_{ij}(t_i, t_j)$  on each edge  $\mathcal{E}_{ij}$  can be expressed as:

$$\mathbf{E}_{ij}(t_i, t_j) = \alpha_\lambda \cdot P \cdot \sum \|\Psi_i(t_i, \theta_{t_i}) - \Psi_j(t_j, \theta_{t_j})\|_\lambda^2 \quad (6)$$

Here  $i$  and  $j$  are the indices of the two adjacent patches in  $\Omega$ . A penalty scheme is applied to the similarity comparison. The two parameters for  $\Psi_i$  indicate the index and rotation for the source patches in  $\{\hat{\Psi}_{t_i}\}$ . The messages propagation is derived from the results of the above energy functions. We adopt a similar method as [21], where the message  $M_{ij}$  passes by patches  $\Psi_i$  is defined as:

$$M_{ij} = \mathbf{E}_i(t_i) + \mathbf{E}_{ij}(t_i, t_j) \quad (7)$$

Through iterative updating on the *BP* graph, an optimal decision of  $\{t_i\}$  for the patches in  $\{\Psi_i\}$  is made by minimizing the nodes' energy. This principle can be formulated in the definition below:

$$\hat{t}_i = \arg \min_{t_i} \{\mathbf{E}_i(t_i) + \sum_k M_{ki}\} \quad (8)$$

Where  $k$  is one of the neighbors of the patch  $\Psi_i$ :  $k \in \mathcal{N}(i)$ .  $\hat{t}_i$  is the optimal index for the matching patch. To achieve minimum global energy cost, dynamic programming is used. Each assignment for  $\Psi_i$  or  $\mathbf{a}_i$  is considered as a stage. In each stage, the choices of  $\hat{\Psi}_{t_i}$  represent different states. The edge  $\mathcal{E}_{ij}$  represents the transit cost from state  $\hat{\Psi}_{t_i}$  at stage  $i$  to state  $\hat{\Psi}_{t_j}$  at stage  $j$ . Starting from  $i = 0$ , an

optimal solution is achieved by minimizing the total energy  $\xi_i(t_i)$  from last step:

$$\xi_i(t_i) = \mathbf{E}_i(t_i) + \min\{\mathbf{E}_{ij}(t_i, t_j) + \xi_{i-1}(t_{i-1})\} \quad (9)$$

where  $\xi_i(t_i)$  represents a set of different total energy values at current stage  $i$ . In the situation of multiple intersections among curves  $\mathcal{C}$ , we adopted the idea in [21], where readers can refer for further details.

### C. Remaining Part Filling:

After the curves are generated in  $\Omega$ , we fill the remaining regions by using the exemplar-based approach in [20]. The  $\partial\Omega$  is getting smaller and smaller by spreading out the known pixels  $\Phi$  in a certain order. To enhance the accuracy, all the pixels in the above generate patches along the estimated curves are assigned with a pre-computed confidence value based on the confidence updating rule in [20].

## IV. EXPERIMENTS

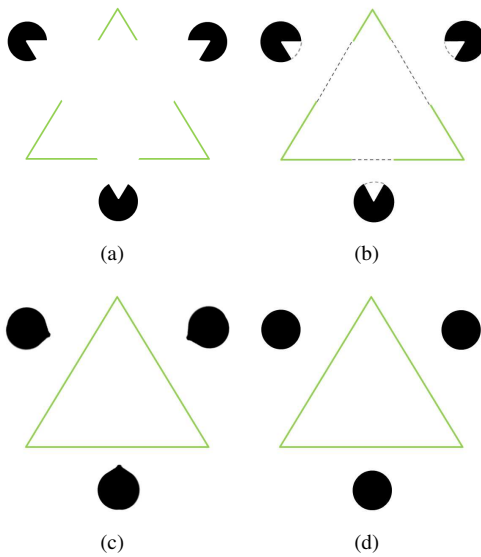


Fig. 2. Kanizsa Triangle Experiment (a) The original Image (b) Curve reconstruction for the missing region  $\Omega$  (c) Result by Criminisi's method (d) Our result.

In our experiments, we first evaluate our proposed approach in terms of structure coherence by comparing our result with the one in [20] that works on the well-known *Kanizsa triangle*. As shown in Figure 2(a), the white triangle in the front is considered as the occluded region  $\Omega$  that needs to be removed. First, a structure propagation is carried out based on the detected edges along  $\partial\Omega$ . The dash lines in Figure 2(b) indicate the estimated potential structures in  $\Omega$ . Texture propagation is applied to the rest of the image based on the confidence and isophote terms. One can notice both

the triangle and the circles are well completed in our result Figure 2(d) comparing with Criminisi's method in Figure 2(c).

To further demonstrate the performance, a set of images are used for scene recovery: ranging from indoor environment to natural scenes. Figure 3(e) shows an indoor case where highly structured patterns often present, such as the furniture, windows, walls. The green bottle on the office partition is successfully removed while preserving the remaining structure. In this example, three pairs of edges are identified and connected by the corresponding curves that are generated in the occluded region  $\Omega$ . Figure 3(g) and 3(f) show the results of removing trees in the nature scenes. Several curves are inferred by matching the broken edges along  $\partial\Omega$  and maximizing the continuity. We can notice the three layers of the scene (sky, background trees, and grass land) are well completed. In Figure 3(h), it shows a case that a perching bird is removed from the tree. Our structure estimation successfully completes the tree branch with smooth geometric and texture transitions.

## V. CONCLUSION

In this paper, we present a novel approach for foreground objects removal while ensuring structure coherence and texture consistency. The core of our approach is using structure as a guidance to complete the remaining scene. This work would benefit a wide range of applications especially for the online massive collections of imagery, such as photo localization and scene reconstructions. Moreover, this work is applied to privacy protection by removing people from the scene.

## REFERENCES

- [1] Pablo Arbeláez, "Boundary extraction in natural images using ultrametric contour maps," *Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06*, pp. 182 – 190, Jun 2006.
- [2] Yu Su, Yi Wang, Gagan Agrawal, and Rajkumar Kettimuthu, "Sd-query dsi: integrating data management support with a wide area data transfer protocol," in *Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 47.
- [3] Yi Wang, Yu Su, and Gagan Agrawal, "Supporting a light-weight data management layer over hdf5," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013, pp. 335–342.
- [4] Yi Wang, Wei Jiang, and Gagan Agrawal, "Scimate: A novel mapreduce-like framework for multiple scientific data formats," in *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*. IEEE, 2012, pp. 443–450.
- [5] Qingquan Sun, Peng Wu, Yeqing Wu, Mengcheng Guo, and Jiang Lu, "Unsupervised multi-level non-negative matrix factorization model: Binary data case," *Journal of Information Security*, vol. 3, no. 4, 2012.
- [6] Fei Hu, Qi Hao, Marcin Lukowiak, Qingquan Sun, Kyle Wilhelm, Stanislaw Radziszowski, and Yao Wu, "Trustworthy data collection from implantable medical devices via high-speed security implementation based on ieee 1363," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 6, pp. 1397–1404, 2010.

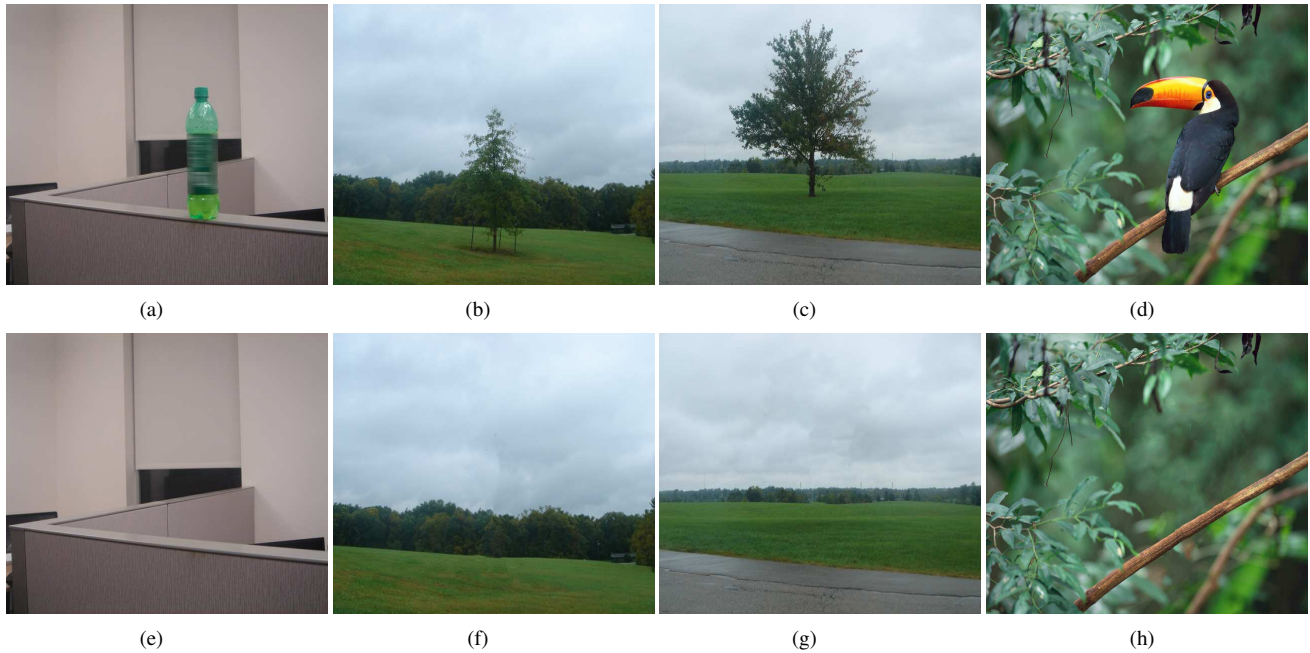


Fig. 3. Scene reconstruction results on different settings: the first row shows the original images; the second row shows the corresponding result images.

- [7] Jianjun Yang and Zongming Fei, "HDAR: Hole detection and adaptive geographic routing for ad hoc networks," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, August 2010.
- [8] Jianjun Yang and Zongming Fei, "Broadcasting with prediction and selective forwarding in vehicular networks," in *International Journal of Distributed Sensor Networks*, 2013.
- [9] Ju Shen, Anusha Raghunathan, S.S. Cheung, and Rita Patel, "Automatic content generation for video self modeling," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 2011, pp. 1–6.
- [10] Ju Shen, Changpeng Ti, S.-C.S. Cheung, and R.R. Patel, "Automatic lip-synchronized video-self-modeling intervention for voice disorders," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, 2012, pp. 244–249.
- [11] Miao Pan, Xiaoyan Zhu, and Yuguang Fang, "Using homomorphic encryption to secure the combinatorial spectrum auction without the trustworthy auctioneer," *Wireless Networks*, vol. 18, no. 2, pp. 113–128, 2012.
- [12] Jianjun Yang and Zongming Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*. IEEE, 2008, pp. 96–101.
- [13] Jianjun Yang and Zongming Fei, "Statistical filtering based broadcast protocol for vehicular networks," in *Proceedings of IEEE ICCCN*, 2011, vol. 11, pp. 1–6.
- [14] Kunjie Xu, David Tipper, Prashant Krishnamurthy, and Yi Qian, "An efficient hybrid model and dynamic performance analysis for multihop wireless networks," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 1090–1096.
- [15] Kunjie Xu and Mu Zhou, "Energy balanced chain in IEEE 802.15.4 low rate WPAN," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 1010–1015.
- [16] Kunjie Xu, Siriluck Tipmongkongsilp, David Tipper, Yi Qian, and Prashant Krishnamurthy, "A time dependent performance model for multi-hop wireless networks with CBR traffic," in *29th IEEE International Performance Computing and Communications Conference*, 2010.
- [17] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," *Proc. Int. Conf. Computer Vision*, pp. 1033–1038, September 1999.
- [18] A. Efros and W.T. Freeman, "Image quilting for texture synthesis and transfer," *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*, pp. 341–346, August 2001.
- [19] J. Jia and C.-K. Tang, "Image repairing: Robust image synthesis by adaptive nd tensor voting," *Proc. Conf. Comp. Vision Pattern Rec.*, 2003.
- [20] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep 2004.
- [21] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum, "Image completion with structure propagation," *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 2005.
- [22] J. Shen, P.C. Su, S.C. Cheung, and J. Zhao, "Virtual mirror rendering with stationary rgb-d cameras and stored 3d background," *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 2013.
- [23] Ju Shen and S.-C.S. Cheung, "Layer depth denoising and completion for structured-light rgb-d cameras," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 1187–1194.
- [24] Ju Shen and Wai Tian Tan, "Image-based indoor place-finder using image to plane matching," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 2013, pp. 1–6.
- [25] Pablo Arbeláez, Michael Maire, Charles Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 898–916, May 2011.
- [26] P. W. Lamberti, A. P. Majtey, A. Borrás, M. Casas, and A. Plastino, "Metric character of the quantum jensen-shannon divergence," *Phys. Rev. A*, vol. 77, pp. 052311, May 2008.
- [27] James McCrae and Karan Singh, "Sketching piecewise clothoid curves," *Computers and Graphics*, vol. 33, pp. 452–461, Aug 2009.
- [28] Mullinex G and Robinson ST, "Fairing point sets using curvature," *Computer Aided Design*, vol. 39, pp. 27–34, 2007.