Volume xx (200y), Number z, pp. 1–15

Recognition-Difficulty-Aware Hidden Images Based on Clue-map

Yandan Zhao, Hui Du, Xiaogang Jin

State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China

Abstract

Hidden images contain one or several concealed foregrounds which can be recognized with the assistance of clues preserved by artists. Experienced artists are trained for years to be skilled enough to find appropriate hidden positions for a given image. However, it is not an easy task for amateurs to quickly find these positions when they try to create satisfactory hidden images. In this paper, we present an interactive framework to suggest the hidden positions and corresponding results. The suggested results generated by our approach are sequenced according to the levels of their recognition difficulties. To this end, we propose a novel approach for assessing the levels of recognition difficulty of the hidden images and a new hidden image synthesis method that takes spatial influence into account to make the foreground harmonious with the local surroundings. During the synthesis stage, we extract the characteristics of the foreground as the clues based on the visual attention model. We validate the effectiveness of our approach by performing two user studies, including the quality of the hidden images and the suggestion accuracy.

Categories and Subject Descriptors (according to ACM CCS): I.4.9 [Image Processing and Computer Vision]: Applications—

1. Introduction

Hidden images, also referred to as camouflage images, are a form of visual illusion art, in which artists embed one or more unapparent figures or foregrounds. At first glance, viewers can only see the apparent background, while they can recognize the foreground through clues after carefully watching over a period of time. This can be explained by the feature integration theory [TG80, Wol94]. To conceal the foreground, artists only retain a portion of features which could be integrated as clues for viewers' recognition [HP07, HTP07].

Generating successful and interesting hidden images is not an easy task, even for skilled artists. To provide a convenient tool for artists, previous work [YLK08, CHM*10, TZHM11, DJM12] has tried to create pleasing results with natural photos. They utilize rough luminance distribution or the edge information of the foreground as clues, which might lead to missing some characteristics of the foreground. Skilled artists may find appropriate hidden positions for a given background according to their prior knowledge. However, due to a lack of this specialized expertise, it is frequently difficult for amateurs to quickly find these positions in the background. Tong et al. [TZHM11] find only one best hidden position by matching the edges extracted from the background and the foreground. The foreground would be hidden perfectly if its edges could be replaced by the edges of the background. However, the saliency of the edges has not been taken into account, and it is inevitable that some trivial edges would be extracted. In most cases, there is little prospect of finding a matching shape for every edge. If the salient edges on which viewers' recognition relies are not matched, the foreground cannot be concealed very well in the best position. Moreover, shape-matching is time-consuming, and blurring artifacts may occur because of the transformation of the foreground during shape-matching.

Fig. 1a, an example of hidden images made by an artist, shows how artists select embedded positions and clues. Instead of a best position, the artist selects several different ones to conceal several eagles, and most of the positions contain a rich texture. Because the local texture of these embedded positions is different, the appearances of these eagles are designed to be dissimilar in order to be harmonious with the local surroundings. Nevertheless, these quite different eagles still can be recognized. This is due to the carefully designed

(a) Hidden images

(b) Answer

Figure 1: *Hidden images created by artist: 9 eagles.*

clues. The artist selects the region representing the salient characteristics of the foreground as the clues. The characteristics of eagles are the eyes and the beaks, according to which we distinguish eagles from other animals. The hidden eagles circled by the green curves in Fig. 1b can be recognized through these characteristics. Unfortunately, automatically selecting the embedded positions and the characteristics of the foreground remains as a challenge.

In this paper, we present a fast hidden-image generation system, which is composed of a recognition difficulty assessment method and a hidden image synthesis method. The former evaluates the recognition difficulty of each embedded position by measuring the rich degree of the texture in the background. Our system suggests several appropriate hidden positions and provides corresponding results based on the evaluation. Users also can select positions personally according to their requirements. Our method generates natural-looking hidden images based on texture synthesis techniques and preserves the salient characteristics as clues. To select the clues as artists do, we propose a method that automatically extracts the clues based on the focus attention model [IKN98]. This model selects the characteristic information which probably attracts relatively more attention. The appearance of the hidden foreground varies when it is embedded in different positions, in order to be harmonious with the surroundings. To this end, we add a space factor to the hidden image synthesis, to change the appearance with the shift of the hidden position.

Our major contributions can be summarized as follows:

- A novel computer-suggested hidden-image generation system which automatically provides several appropriate hidden positions and corresponding results.
- The first recognition difficulty assessing method which estimates the recognition difficulty of each foreground embedded position.
- A new hidden-image synthesis method that utilizes a new clues-extraction scheme, which achieves a balance between the preservation of the foreground's characteristics and as much variation as possible of the foreground with the changing of the embedded position. The space factor is also taken into account to make the foreground harmonious with the local surroundings.
- The performance of hidden-image synthesis is significantly improved. Both our hidden-image synthesis method and the difficulty assessing method can provide real-time feedback.

2. Related Work

In this section, we review related work from three aspects: hidden image synthesis, embedded position analysis, and the application of visual attention models in recreational art.

Hidden image synthesis A collage [GSP*07, HZZ11] is

one kind of hidden image, which is constructed from different elements. In this approach, both the collage images and the elements are immediately recognizable. Moreover, the collage images are not characterized by changing the appearance of the elements. Yoon et al. [YLK08] use stylized line drawings to render the background and the foreground, and then employ the edges of the foreground as clues to find suitable hidden positions by shape-matching. Instead of hiding line art foreground in a line art background, we aim at hiding textured foregrounds into a natural image. Tong et al. [TZHM11] also utilize the edges as clues to aid users' recognition, and try to find the best hidden position by shape-matching. However, their hidden results include blurring artifacts because of the transformation. Du et al. [DJM12] employ the edges of the foreground as clues and formulate the hidden image synthesis as a blending optimization problem as well. When the contrast of the background is quite low, too many details which are distinct from the surroundings are preserved. When the contrast of the background is quite high, the characteristics of the foreground may not be preserved. Chu et al. [CHM*10] synthesize the hidden images on the basis of texture synthesis techniques and use luminance distribution as the clues, which often makes the foreground change slightly in different positions and sometimes be easy to find. To increase recognition difficulty, they have to add some distracting segments randomly. However, this probably disturbs important foreground regions, such as the eyes and mouth of animals. Our method also synthesizes the hidden images based on texture synthesis techniques. The difference is that we utilize the focus attention model [IKN98] to extract the clues, which always contain the characteristics of the foreground. All of the above works are 2D camouflage. Owens et al. [OBF*14] camouflage a 3D object into the background photographs which are captured from many viewpoints. The problem that they face is different from ours, which tries to match the texture of the cube with the background from every possible perspective.

Embedded position analysis Chu et al. [CHM*10] present users with a list of candidate positions and orientations that can minimize their energies. Tong et al. [TZHM11] find the best embedded position by matching the edges of the background and the foreground. Different from them, our method solves this problem on the basis of the observation that the foreground can be concealed better in the regions whose texture is relatively more complicated and colorful. Tong et al. [TZHM11] use the Sobel operator to extract edges which probably include some edges useless for recognition. If the matched edges are mostly useless ones, the foreground cannot be concealed well in the best position. Moreover, both of them leave the recognition difficulty to users and cannot find the positions in real time. Our method not only provides several relevant suggestions but also estimates the recognition difficulty of each position.

Applications of visual attention models in recreational art The visual attention model has been an active research direction with many recreational applications. Change blindness is a psychological phenomenon in which viewers often do not notice some visual changes between images. Ma et al. [MXW*13] employ the visual attention model to quantify the degree of blindness between an image pair. They use a region-based saliency model which investigates how the saliency of one region is influenced by other regions in its long-range context. However, this region-based model is not suitable for our application because what we need is a pixel-based saliency model. Image abstraction (stylization) is a widely acknowledged form of recreational art. To control the level of details of the abstraction results, DeCarlo et al. [DS02] apply the human perception model by using additional data tracking devices to collect the eye movements data. When transferring a photograph into the style of an oil-painting, Collomosse et al. [CH02] use the visual attention model to detect the edges of photographs, and only draw edges with high attention. Their visual attention model determines the importance by the gradient information only. The color information which will be used in our approach is not taken into account in their model. To simulate the emphasis effects of real recreational art and to predict viewers' attention, Zhao et al. [ZMJ*09] and Hate et al. [HTM12] use the attention model [IKN98] to control the degree of abstraction and stylization. Itti et al.'s basic model [IKN98] utilizes three feature channels (color, intensity, and orientation) and defines image saliency using central surrounded differences across multi-scale image features. This model simulates the neuronal architecture of the visual system and has been shown to correlate with human eye movements. Therefore, we employ this visual attention model to calculate the foreground's *clue-map*. The clue-map describes which regions are likely to be selected as clues to assist the viewers' recognition.

3. Our method

Given a background image B and a foreground image F, our algorithm generates a hidden image which hides F in B and retains some clues for viewers to recognize. When generating a hidden image, the user needs to specify the hidden position. A satisfactory hidden image depends highly on the selection of hidden positions. To this end, our hidden image system provides two components: hidden position analysis (the red shaded part) and hidden image synthesis (the green shaded part), as shown in Fig. 2.

For hidden position analysis, we analyse the recognition difficulty for each position and compute a heat-map of the recognition difficulty. Fig. 2 shows an example of a heat-map R of the recognition difficulty. In this map, green regions are of low recognition difficulty, and red regions are of high recognition difficulty. The heat map offers important references for users to select appropriate hidden positions. Moreover, our system will automatically recommend the hidden positions with the highest recognition difficulties

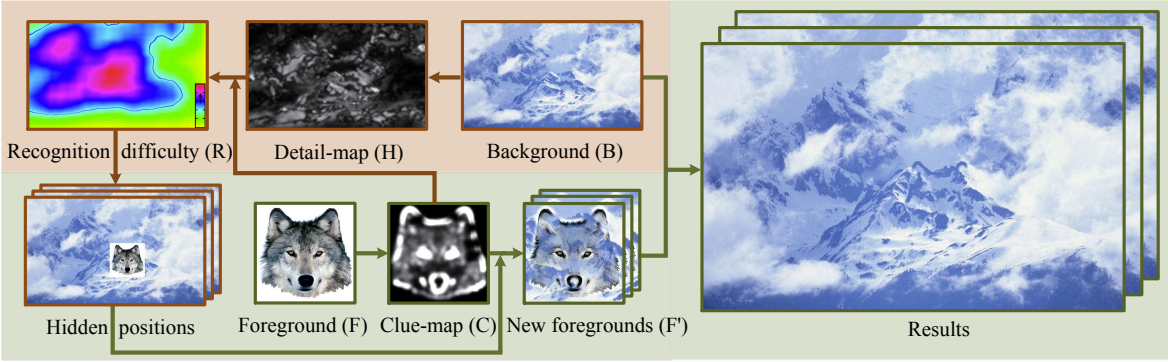


Figure 2: *The pipeline of our approach.*

and will generate hidden results in real time. Visual perception study of hidden images [TBL*09] shows that the foreground is more difficult to detect when there is more high-contrast detail in the surrounding. This result implies that the recognition difficulty R can be modulated by the rich degree of the high-contrast detail. Considering the fact that a viewer can only focus on a small area at one time, we compute the recognition difficulty of a specific position based on the rich degree of the detail of a small surrounding area. Here, we use a *detail-map* H to represent the quantization of the rich degree of detail; and a *clue-map* C to choose surrounding regions, as illustrated in Fig. 2.

In hidden images synthesis, there is a balance between two conflicting factors: *immersion*, which is responsible for the harmony between the foreground and the surrounding background, and *standout*, which is responsible for the viewer recognizing the hidden image. The main task of hidden image synthesis is to find a solution which achieves a satisfactory balance.

Similar to Chu et al. [CHM*10], we use texture synthesis to replace the texture of the foreground with that of the background. Chu et al. [CHM*10] segment the foreground and optimize the luminance distribution of the segments to achieve the balance. They retain the luminance distribution as the recognition clue because the luminance channel can offer sufficient cues for recognition [Pal99]. However, when the luminance contrast is fairly low, the balance cannot be obtained. Moreover, their weights of the balance, which include the length of the boundary and the area of the segments, are not directly related to the contribution of each segment for recognition. As a result, their approach cannot guarantee that the characteristics of the foreground are always preserved. Different from them, we achieve the balance by blending the background B and the foreground F to generate the new foreground F' which contains the characteristics of F and rich details of B. The blend parameter is determined by the *clue-map* which illustrates the contribution of the foreground regions for recognition. The regions with larger contributions are responsible for *standout* and assist the viewers' recognition. The luminance of these regions remains the same as the original luminance of the foreground. The other regions are responsible for *immersion*, which aims for harmonization with the background. Therefore, F' preserves the clues and makes the foreground harmonious with the local surrounding, when hiding in different positions. Similar to hidden image synthesis algorithms utilizing Poisson blending [TZHM11, DJM12], our approach automatically adds the distraction in each location.

After generating the new foreground F', we synthesize it using the texture of B. Using the background texture near the hidden position in the texture synthesis is more likely to make the foreground harmonious with the background. To this end, similar to the method of Chu et al. [CHM*10], we take into account the distance between the location of the background and the location of the foreground in texture synthesis.

Based on the above analysis, we summarize the key steps of our algorithm as follows (Fig. 2):

i) Extract the *clue-map* C of the foreground image F and the *detail-map* H of the background image B.
ii) Compute recognition difficulty R using C and H.
iii) Specify hidden positions according to R. Users can directly adopt the system's suggestion or select the hidden position manually by referring to R.
iv) Blend B and F utilizing C as the interpolation parameter and generating the new foreground F'.
v) Synthesize the result using the texture from the background B near the embedded position.

In the following sections, we will describe *clue-map* generation, recognition difficulty analysis, and hidden image synthesis, respectively.

4. Clue-map

The *clue-map* C is used to select the recognition clue of the foreground. When artists draw a hidden image, the regions that are selected as recognition clues are drawn more like the foreground and the other parts are designed to be harmonious with the background. Salient regions are more likely to be selected as the recognition clues, such as the outline of an object, the facial features of an animal or a human being. The *clue-map* records the probabilities of the foreground positions which are selected as recognition clues. We calculate the *clue-map* of the foreground by utilizing the visual attention model of Itti et al. [IKN98], since the probability of each position in the foreground is related to its saliency.

The saliency map is constructed by considering how conspicuous color U, intensity I and orientation O of each position are, compared to their surroundings. We first build a map of each, normalise them on [0,1], and then calculate a linear combination as follows:

$$M = \frac{1}{3}(N(U) + N(I) + N(O)), \tag{1}$$

where $N(:)$ denotes the normalization.

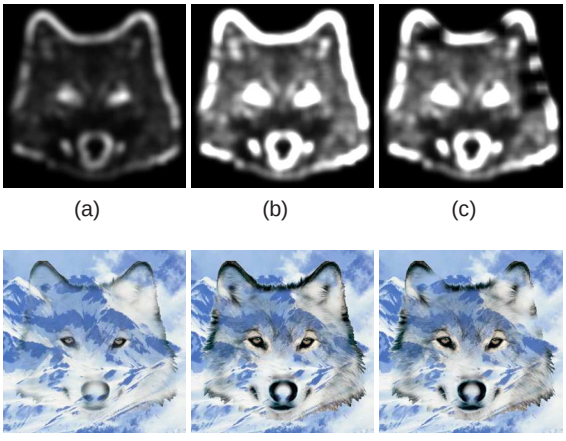


Figure 3: *Intermediate results during the clue-map generation. From left to right: (a), (b) and (c) are the intermediate results of the clue-map from three steps, respectively. (d), (e) and (f) are F' generated using (a), (b) and (c), respectively.*

We then increase the probabilities of the *clue-map* by 3 - 5 times (clamped to [0,1]) (Fig. 3b), in order to sufficiently preserve the characteristics of F when we blend F and B (Fig. 3d).

After that, we detect the long-strip regions of the *clue-map* and break them up (Fig. 3c). Observing Fig. 3b, we find that some regions near the outline of the wolf head present as long strips where the probabilities are higher. The long-strip regions are caused by the dark regions near the wolf's ears and the long dark regions will be preserved in F' completely (Fig. 3e). Perception studies [BM03, EG01] show that long coherent regions attract the viewers' attention, and so viewers will find the foreground too quickly if they are left in place. Accordingly, we reduce the probabilities of the long-strip regions in parts, to break these long regions into short regions and so obtain better hidden results (Fig. 3f). Although the long regions are broken, viewers can fill in the gaps using their imagination and prior experience according to the Gestalt principle of closure.

To cut down the long regions of F', we multiply the probabilities by weights W:

$$W(\mathbf{p}) = e^{-d_p/2}, \tag{2}$$

where $\mathbf{p}$ is a pixel of the *clue-map*. We use Φ to denote the regions whose probabilities will be reduced. d_p is the nearest distance from $\mathbf{p}$ to the pixels outside of Φ. If $\mathbf{p}$ is not in Φ, the weight is 1.0 and the probabilities will not be changed. We compute d_p using the Euclidean distance transform after Φ is obtained.

When determining the region Φ that will be cut down, two problems need to be addressed:

i) Retaining the shape of the long-strip region's skeleton.
ii) The probabilities of regions, except the long-strip regions, should be retained.

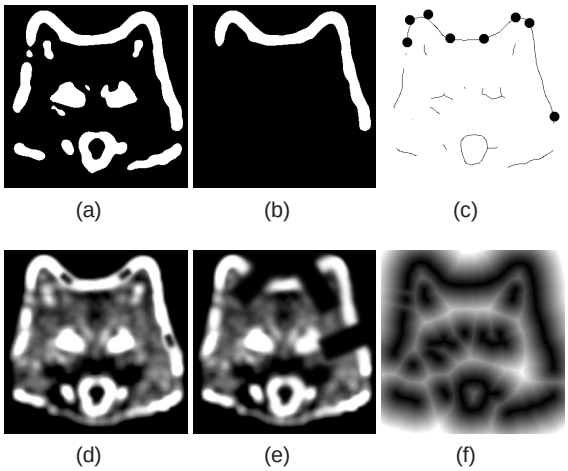


Figure 4: *Intermediate results of cutting down long-strip regions in the clue-map generation. From left to right: (a) Segmentation of the* clue-map*; (b) The long-strip regions; (c) The skeleton of the segmentation; (d) The cutting result using a small fixed width; (e) The cutting result using a large fixed width; (f) The distance transform of skeletons.*

For the first problem, we need to extract the skeletons of the long-strip regions and analyze their shape. To this end, we first select high-probability regions by segmentation, and then choose long-strip regions according to their lengths. After that, we extract their skeletons and maintain their shapes

by preserving their dominant points. We segment the *clue-map* by k-means segmentation ($k = 3$) and choose the region with the highest mean of probability as the candidate region (Fig. 4a). A region is considered as a long-strip region if it is longer than half of the foreground's inscribed radius (Fig. 4b). We extract the skeleton (Fig. 4c) of the candidate region using an image thinning technique [STRA10]. The dominant points (Fig. 4c) of the skeleton are detected by Teh et al.'s method [TC89]. In order to retain the shape of the skeleton of the long-strip regions, the probabilities of the pixels whose distances to the dominant points are less than r are held constant ($r = 10$).

To solve the second problem, we need to determine the regions whose probabilities will be changed. A naive solution for determining the regions is to cut the long-strip regions with a fixed width. However, the width must be appropriate to guarantee the cut is complete (Fig. 4d), otherwise, the characteristics of other regions will be changed (Fig. 4e). To determine the regions automatically, we employ the Euclidean distance transform (Fig. 4f) of the skeleton. For each pixel, we record its nearest pixel on the skeleton. The probability of a pixel in Φ will be changed if its nearest pixel on the skeleton belongs to Φ. Therefore, we first determine the pixels along the skeleton whose probability will be changed, and then the other pixels.

When determining the region Φ to be cut off, we first randomly select n pixels Φ_n on candidate regions ($n = 5$ by default). For a given pixel $\mathbf{p}$ in Φ_n, we calculate the l_T (10 for default) nearest pixels to $\mathbf{p}$ along the skeleton, and these pixels form a pixel set Φ_T. Then, we find the pixels whose nearest pixels belong to Φ_T, and they form the pixel set Φ_N. Then the region Φ is defined as:

$$\Phi = \Phi_n \cup \Phi_T \cup \Phi_N. \tag{3}$$

5. Recognition difficulty analysis

Visual perception study of hidden images [TBL*09] finds that the foreground is more difficult to find where there are more high-contrast internal details. This implies that the recognition difficulty R can be modulated by the rich degree of the high-contrast detail in the detail-map H of the local surrounding. As described in step iv) in Section 3, the details are added from the background by blending, based on the *clue-map* C. Therefore, the recognition difficulty R of a fixed position $\mathbf{Q}$ can be computed according to the rich degree of the detail of the local surrounding Ω_F:

$$R(\mathbf{Q}) = \frac{1}{wh} \sum_{\mathbf{p} \in \Omega_F} (1 - C(\mathbf{p}))H(\mathbf{p} + \mathbf{Q}), \tag{4}$$

where $\mathbf{Q}$ denotes the hidden position of the foreground which is also that of the foreground's top left corner; Ω_F is the region of the background covered by the foreground; w and h are the width and height of the foreground. The *detail-map* H is estimated by the subtraction of the low frequency information from the original information:

$$H_l(\mathbf{p}) = \max_{i=r,g,b} \frac{|B_{i,l}(\mathbf{p}) - f(B_{i,l}(\mathbf{p}))|}{m_{i,l}}, \tag{5}$$

$$m_{i,l} = \max_{\mathbf{p} \in B} |B_{i,l}(\mathbf{p}) - f(B_{i,l}(\mathbf{p}))|, \tag{6}$$

where r, g, and b denote the red, green and blue channels respectively; l represents the level of the Gaussian pyramid (Here, a two-level pyramid is used); H averages the outcomes of all levels; $m_{i,l}$ denotes the maximum of all pixels for corresponding level and channel; and f represents the box filter.

In our experiments, we modified the box filter by considering segmentation. Without segmentation, errors will arise in the regions which cross the boundaries of the different regions, such as sky and mountains. In other words, the values of H for these regions are extremely high, whereas the foreground will be easy to find in these regions. To decrease the H of these regions, we segment the background in advance automatically. We compute SLIC superpixels [ASS*12], and the segmentation is implemented by grouping the superpixels into k clusters by their average values. In our experiments, $k = 5$. After segmentation, the filter only involves the pixels which are in the same segment as the center pixel of the filter kernel. The size of the filter kernel is 25% of the size of the foreground.

6. Hidden image synthesis

Hidden images are synthesized using the texture of the background. Firstly, we convert the foreground F and the background B from the RGB color space to the YIQ color space and use the Y channel as luminance for the following similarity analysis in the texture synthesis. Any color spaces which separate luminance from chrominance will work. Secondly, we reduce the distinct color numbers of F and B by luminance quantization [WOG06]. Thirdly, we formulate the texture synthesis as an energy minimization problem and use PatchMatch [BSFG09] to improve performance. The energy function E is the sum of distances $D(\mathbf{p}, \mathbf{q})$ between each pixel $\mathbf{p}$ of F' and its corresponding pixel $\mathbf{q}$ from B.

$$E = \sum_{\mathbf{p} \in F'} D(\mathbf{p}, \mathbf{q}) \tag{7}$$

The similarity distance D between the foreground pixel $\mathbf{p}$ and the background pixel $\mathbf{q}$ comprises two factors: luminance factor L and space factor S:

$$D(\mathbf{p}, \mathbf{q}) = S(\mathbf{p}, \mathbf{q})L(\mathbf{p}, \mathbf{q}), \tag{8}$$

where L measures the luminance difference between $\mathbf{p}$ and $\mathbf{q}$; S is the weight determined by the Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$. S is employed to encourage hidden image to be synthesised using the background texture near the hidden position.

Luminance factor Intuitively, we can replace pixels in the foreground with ones in the background if the luminance

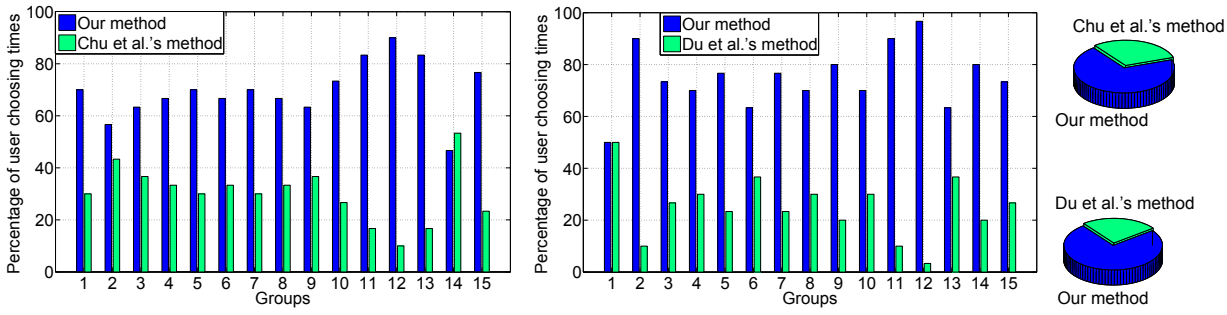


Figure 5: *(a) presents the percentages of times users choose in each group created by Chu et al.'s method [CHM*10] and our method, respectively. Except for group 14 (47%), users choose our method over 50% of the time. On average, our method gets 69.80% of the total times, compared to Chu et al.'s method (upper row of [c]). The same statistical comparison between the Du et al.'s method [DJM12] and our method is presented in (b). Group 12 exhibited the maximum choosing times (97%), while group 1 was the minimum (50%). On average, our method gets 74.87% compared to Du et al.'s method (lower row of [c]).*

difference between their corresponding pixel pair is small. In order to simplify computation, we only use average values of blocks to obtain the value of the luminance difference:

$$L(\mathbf{p},\mathbf{q}) = \left|\overline{N(\mathbf{p})} - \overline{N(\mathbf{q})}\right| \tag{9}$$

where $\overline{N(*)}$ means the average of the pixel's 5×5 neighborhood. **Space factor** Because the texture of the different positions in the natural image probably comes from different objects, we synthesize the foreground using the texture from local surroundings. When concealed in the different positions, the hidden foreground could have different appearances. To this end, we add a space factor in the similarity distance. This factor is measured by the distance from the embedded position. In other words, given a fixed position, the size of the foreground is $w\times h$, and we define the potential region for selecting texture to be $\gamma w\times \gamma h$. We penalize the regions which are not in the potential region with a larger weight. The space factor is defined as follows:

$$S(\mathbf{p},\mathbf{q}) = \begin{cases} 1, & d(\mathbf{p},\mathbf{q}) < \gamma\sqrt{w^2+h^h} \\ \alpha d(\mathbf{p},\mathbf{q}), & otherwise \end{cases} \tag{10}$$

where $d(.)$ denotes the Euclidean distance between two pixels; and we set $\gamma = 1.5$ and $\alpha = 100$ in our experiments.

7. User Study

The ultimate judges of hidden images are humans. Thus, we have devised a user study to objectively verify the effectiveness of our hidden image synthesis method and recognition difficulty assessing method. The user study consisted of two cases. The first case compared the synthetic results of our method with those of state-of-the-art methods. We selected two representative methods [CHM*10, DJM12] which are based on the texture synthesis technique and solving Poisson equation, respectively. The second case verified the recognition difficulty assessing method by comparing the estimated recognition difficulties with the actual recognition times of participants. We invited 30 participants who had different artistic background and areas of knowledge for the study. They completed the task independently, and were unaware of the study's purpose.

Verification of hidden image synthesis In this case, the task consisted of 15 groups of hidden images. Each group contained three images produced by our method, Chu et al.'s method [CHM*10], and Du et al.'s method [DJM12], respectively. The images in each group were created by embedding the same foreground into the same position of the same natural image.

During the task, test images were presented to participants pair-by-pair randomly. Each pair consisted of two images from the same group. The image placement within each pair (left or right side) was also randomized. In order to avoid the case in which users recognize our result when it appears again along with the results of different algorithms, we controlled the interval between every two appearances of pairs from the same sample group. During the task, participants were asked to perform two-alternative forced choices (2AFCs), picking out the better one from the two candidates, i.e., the image looks more harmonious and does not lose the characteristics of the foreground. The time of decision-making was recorded, but not limited.

We analyzed the data of this case (see Fig. 5). When asked to choose the better one from the hidden images created by our method and Chu et al.'s method [CHM*10], our method obtained 69.80% of the total times (95% confidence interval, 63.85 to 75.75%). It also achieved 74.87% against Du et al.'s method [DJM12] (95% confidence interval, 68.27 to 81.47%). By performing one-sample, one-tailed t-tests for the representative state-of-the-art methods, we found that participants preferred our method (p-values $\ll 0.001$).

Verification of recognition difficulty assessing For this case, we prepared five groups, each of which contained 10 results embedding the same foreground in the different positions. At the start of this case, participants were instructed using a trial example. Then, the test images were shown to the participants one-by-one, and a blank frame was shown in between. The participants were asked to find the foreground concealed in each image by clicking the potential position. The test of one image was finished when participants identified the foreground and the recognition time was recorded. To reduce fatigue, the participants were given a short break every five images.

For each test image, we used an average recognition time of all participants as its recognition time. Then, we performed the statistical analysis by using the Pearson's R correlation test. As Fig. 6 shows, we found that the recognition difficulty was highly correlated positively with recognition time (Pearson correlation coefficient $r_p = 0.72, p \ll 0.001$). Through this user study, we can conclude that our approach can estimate recognition difficulty with a high level of reliability.

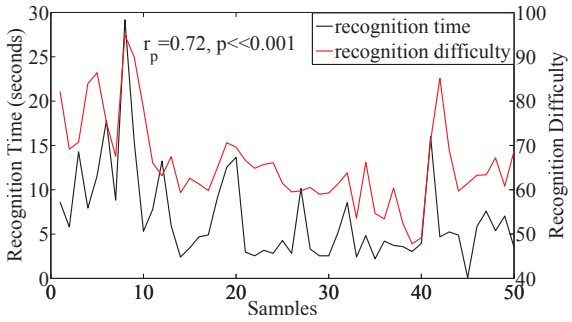


Figure 6: *Statistical analysis of the correlation between participants' recognition time and the recognition difficulty estimated by our method. The recognition time is highly correlated with the recognition difficulty (Pearson correlation coefficient $r_p = 0.72, p \ll 0.001$).*

8. Results

In this section, we show the results generated by our algorithm and compare our approach with previous ones. The experiments demonstrate that our method is fast and our results are visually pleasing.

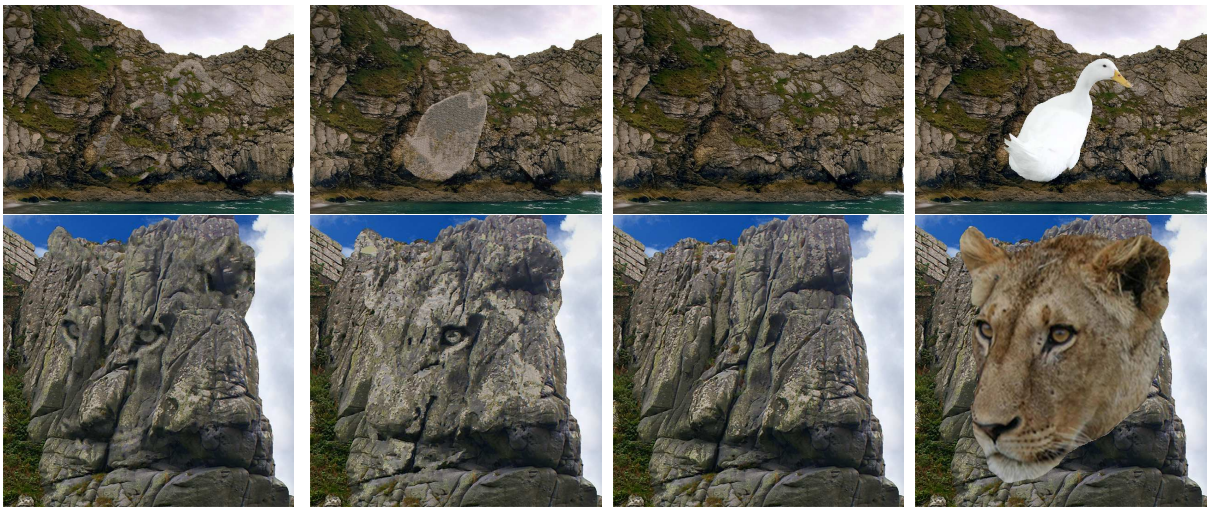
Chu et al. [CHM*10] synthesize the hidden images based on texture synthesis techniques as we do. The difference is that they use rough luminance distribution to present clues, which may make the foreground easy to find, especially when the luminance contrast of the foreground is fairly low. Their failure example (top row of Fig. 7b) is such an extreme example where most of the regions in the foreground are of low contrast and the surrounding of the foreground includes rich details. Even for this example, our approach hides the foreground better. In our result (top row of Fig. 7a), only the salient regions, i.e, the eyes, mouth and outlines of the duck are preserved as the clues, since the low contrast regions are not contained in the *clue-map*. Moreover, the outlines have been cut off to discourage viewers' attention. To increase recognition difficulty, Chu et al. [CHM*10] have to add some distracting segments randomly; however, this probably disturbs critical foreground regions, such as the eyes. For instance, in the bottom row of Fig. 7b, one eye of the lion is not preserved. As a result, Chu et al. [CHM*10] must correct the disturbance interactively. On the contrary, our method can well preserve the eyes automatically (Fig. 7a).

In Fig. 8 and Fig. 9, we compare our synthesized results with those of Du et al. [DJM12]. Unlike us, they formulate the hidden image synthesis as a blending problem and modify the large-scale layer of the background image by nonlinear blending. When the luminance contrast of the background is quite low, too many details of the foreground are preserved, making the foreground easy to find. For example, in the top row of Fig. 8a, the eyes and the fur of the wolf resembles a realistic wolf and is distinct from the surrounding snow. When the luminance contrast of the background is quite high, this may fail to preserve the salient characteristics of the foreground. In Fig. 9a, the rhino horn is blended with the dark background and cannot be found in the result. However, our method can synthesize natural-looking hidden images and preserve the salient characteristics in both situations (Fig. 8b and Fig. 9b).

We only compare our synthesized results in quality with those of Chu et al. [CHM*10] and Du et al. [DJM12], since both of them have not discussed the position issue of hidden image synthesis. Tong et al. [TZHM11] find a best embedded position by shape-matching. Therefore, besides synthesis quality, the difference in solving the position issue between our method and Tong et al.'s method [TZHM11] is also discussed.

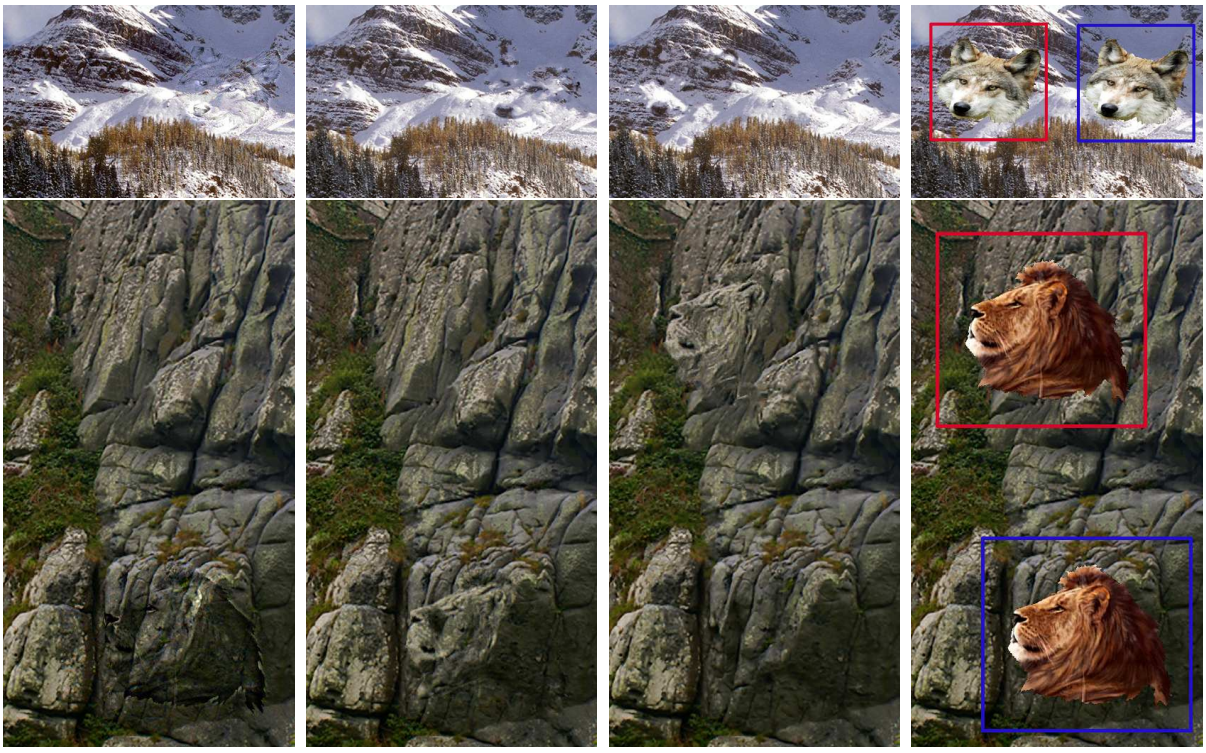
Tong et al. [TZHM11] embed the foreground using a Poisson blending approach. Therefore, their texture in the hidden region is distinct from the surroundings, as is the case in Du et al. [DJM12]. After applying their approach, the luminance in the hidden region is altered and the texture of the hidden region becomes fuzzy, which may lead to an unnatural luminance difference between the hidden region and the background. These may be due to the transformation of the foreground during shape-matching. On the contrary, our approach always guarantees that the resultant texture is natural and coherent (Fig. 10a).

Tong et al. [TZHM11] provide only one best embedded position where the shapes of edges extracted from the background and the foreground are expected to be best matched. However, the saliency of the foreground edges is not taken into consideration. If the edges of the salient characteristic are not matched, their hidden foreground will be easy to recognize. The top row of Fig. 10b is such a case, in which the



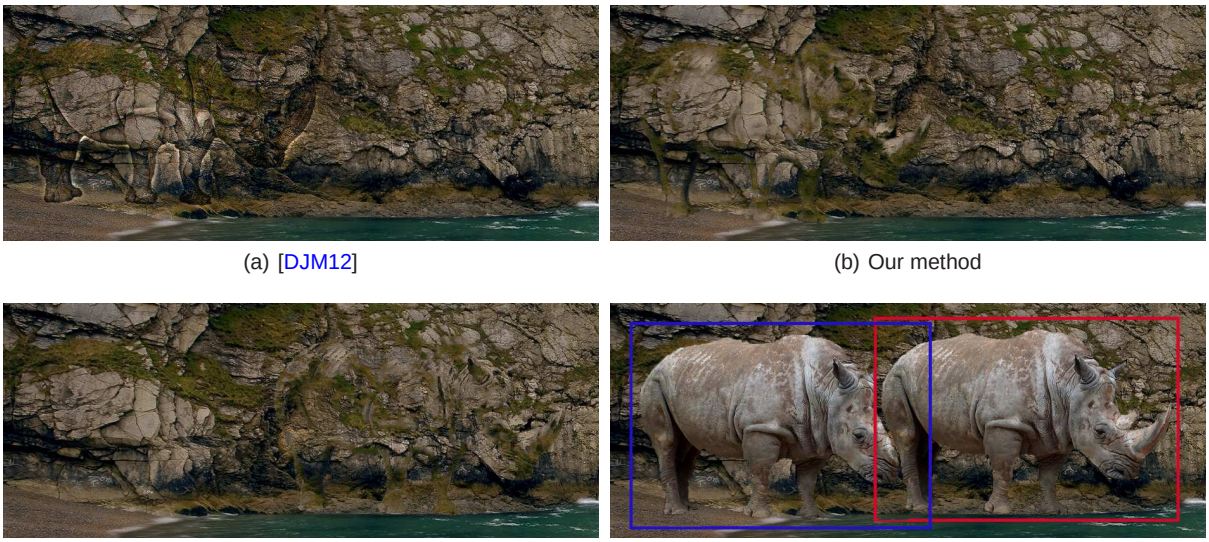
(a) Our method (b) [CHM*10] (c) Background (d) Foreground

Figure 7: *We compare our results with those of Chu et al. [CHM*10] by embedding the foregrounds in the same position in (a) and (b). The original backgrounds, foregrounds, and the embedded positions are shown in (c) and (d).*



(a) [DJM12] (b) Our method (c) Suggested results (d) Answer

Figure 8: *Comparison with Du et al.'s method [DJM12]. We embed the foregrounds in the same position in (a) and (b) and suggest the results in (c) according to recognition difficulty. The embedded positions are shown in (d). The blue and red rectangles denote the embedded positions in the comparison and suggested positions, respectively. Please zoom into the hidden images to better recognize the hidden foregrounds.*



(a) [DJM12] (b) Our method

(c) Suggested results (d) Answer

Figure 9: *Comparison with Du et al.'s method [DJM12]. We embed the foregrounds in the same position in (a) and (b) and suggest the results in (c) according to recognition difficulty. The embedded positions are shown in (d). The blue and red rectangles denote the embedded positions in the comparison and suggested positions, respectively. Please zoom into the hidden images to better recognize the hidden foregrounds.*

eyes and the mouth are not matched while the contour of the face is matched. The unmatched eyes and mouth make the face easy to find. We suggest the positions by assessing the recognition difficulty of each position. Our position analysis finds that the recognition difficulties of large near areas are often approximate. For instance, two different positions represented by rectangles are shown in the second row of Fig. 10d, whose recognition difficulties are approximate. If the best position does not satisfy all the user's requirements, other positions with approximate difficulties could still be chosen. The suggested results are illustrated in Fig. 10c and the hidden positions are shown in red rectangles in Fig. 10d, correspondingly. In these results, the suggested results are more difficult to find. More suggested results are illustrated in Fig. 8c and Fig. 9c.

More hidden images are shown in Fig. 12. The embedded foregrounds are six faces, ten faces, three eagles and four lions, respectively. The answers are shown in Fig. 15. In the second figure of Fig. 12, we embed the same face in two different positions, and two hidden faces present different appearances due to the space factor. All the foregrounds are synthesized using their original color except the left eagle in the third figure. We reverse its color before the synthesis.

We also show the distribution of recognition difficulty with different sizes of the foreground in Figs. 13a-c. The difficult regions (red regions) are scattered when the size becomes small. If the size becomes large, these regions will be concentrated. We also select three results with different recognition difficulties (Figs. 13d-f), whose embedded positions are identified in Fig. 13a.

Our system is implemented using an NVIDIA CUDA programming environment. We ran the program on a 3.40GHz Intel Core i7-2600 CPU and an NVIDIA GeForce GTX 590 GPU. The computation time of our implementation is shown in Table 1. The time of *clue-map* generation does not include that of computing the saliency map. We use Jonathan Harel's Matlab code (http://www.klab.caltech.edu/~harel/share/gbvs.php) for computing the saliency map. The parallel accelerations of image synthesis and recognition difficulty assessment have been implemented, and both of them can obtain real-time feedback (see supplementary video). In Table 1, we also compare with Du et al.'s method [DJM12] on the same machine for performance comparison, quantitatively. The consumed time of our method in comparison contains that of the *clue-map* generation and the image synthesis. Our method is faster than theirs by about 10 times. The other representative methods [CHM*10, TZHM11] take a few to tens of seconds. In addition, Chu et al. [CHM*10] need interaction to add distracting segments and specify the critical foreground regions. In contrast, our method is entirely automatic.

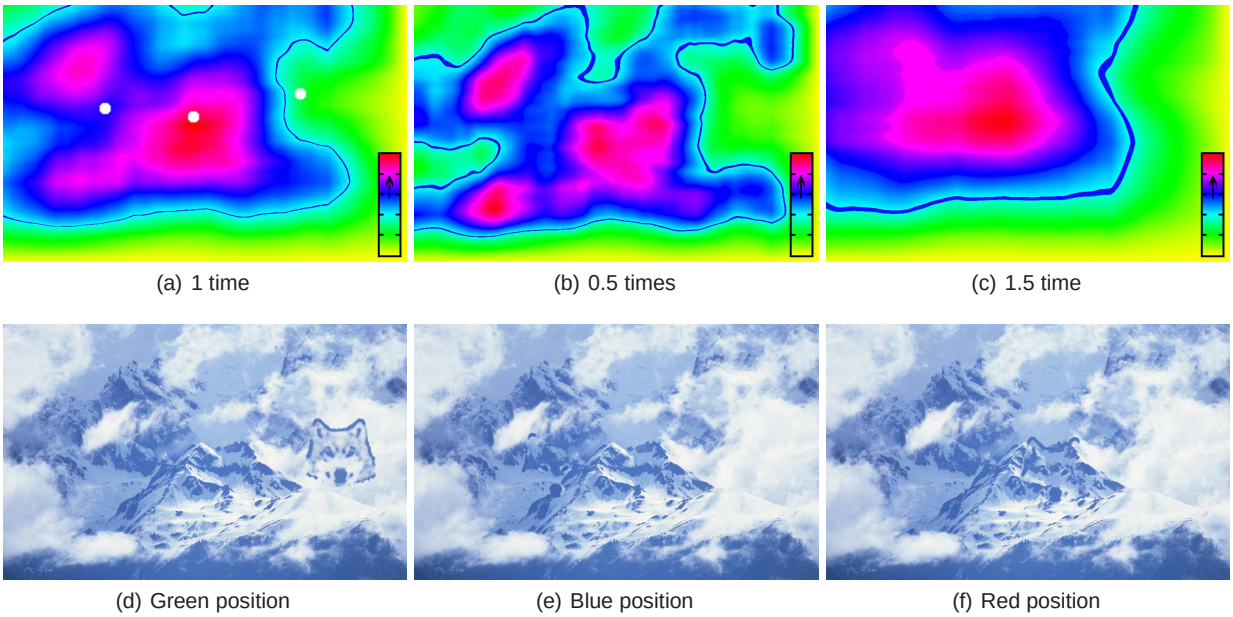
Our method creates hidden images based on texture synthesis techniques. Therefore, it may produce unsatisfactory results (Fig. 14a) when the hidden regions contain some objects which cannot be replaced by surrounding textures. This

(a) Our method (b) [TZHM11] (c) Suggested results (d) Answers

Figure 10: *We compare our results with those of Tong et al. [TZHM11] by embedding the foregrounds in the same position in (a) and (b). The embedded positions are shown in (d). The blue and red rectangles denote the embedded positions in the comparison and suggested positions, respectively. Please zoom into the hidden images to better recognize the hidden foregrounds.*

Figure 11: *Hidden images created by our method. Answers are given in Fig. 15.*

Figure 12: *Hidden images created by our method. Answers are given in Fig. 15.*



(a) 1 time (b) 0.5 times (c) 1.5 time

(d) Green position (e) Blue position (f) Red position

Figure 13: *Heat maps of different sizes of foregrounds and hidden images of different levels of recognition difficulty. From (a) to (c), the sizes of the foregrounds are 1, 0.5 and 1.5 time(s) of the original size, respectively. The heat maps change with the size of the foreground. From (d) to (f), the levels of recognition difficulty increase and the hidden positions can be found in (a). The hidden positions are represented by the upper-left corner of the foreground.*

Table 1: *Performance of our method and comparison to other methods (in seconds).*

Example	Size of the B	Size of the F	Clue-map generation	Image synthesis	Difficulty assessment	Ours CUDA/C++	Du et al. MATLAB
Fig. 8 (row 1)	832×598	192×192	0.047	0.070	0.355	0.117	0.840
Fig. 8 (row 2)	1000×1100	312×267	0.054	0.125	0.962	0.179	2.110
Fig. 9	1340×800	516×382	0.166	0.479	1.334	0.645	4.190
Fig. 10 (row 1)	996×724	316×436	0.102	0.390	0.934	0.492	4.090
Fig. 10 (row 2)	660×786	288×344	0.081	0.176	0.881	0.254	2.350
Fig. 12	1024×648	200×200	0.061	0.082	0.566	0.143	1.070

is a common restriction of hidden image synthesis methods based on texture synthesis.

9. Conclusion

We have developed an interactive hidden image system which can automatically suggest appropriate results by analyzing the recognition difficulties of all positions in the background. To preserve the characteristics of the foreground and increase the recognition difficulty by adding disturbance, we introduced a *clue-map* to guide texture synthesis. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed method.

In the future, we plan to extend our method to hidden videos, which could provide some applications of visual effects. Currently, our method uses only the recognition difficulty of the results as the metric to make suggestions. However, other metrics for selecting appropriate hidden positions might exist. Thus, part of our future work is to explore other metrics to improve the accuracy of our suggestions. In our current implementation, we only suggest results with the highest recognition difficulties. However, we cannot guarantee the highest would always satisfy the users. This is because our recommendation is based on the ranking of recognition difficulty, rather than the consideration of users' design such as the aesthetic effect or the target audiences. How to select an optimum recognition difficulty, which is a highly subjective problem, is another future work.

Acknowledgments We thank Prof. Philip Willis from University of Bath for proof-reading of the paper. Xiagang Jin was supported by the National Natural Science Foundation of China (Grant nos. 61472351, 61272298).

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2274–2282. 6

[BM03] BEAUDOT W. H., MULLEN K. T.: How long range is contour integration in human color vision? *Visual Neuroscience 20*, 1 (2003), 51–64. 5

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics 28*, 3 (2009), 24:1–24:11. 6

[CH02] COLLOMOSSE J. P., HALL P. M.: Painterly rendering using image salience. In *Proceedings of the 20th UK Conference on Eurographics* (2002), IEEE Computer Society, pp. 122–128. 3

[CHM*10] CHU H.-K., HSU W.-H., MITRA N. J., COHEN-OR D., WONG T.-T., LEE T.-Y.: Camouflage images. *ACM Transactions on Graphics 29*, 4 (2010), 51:1–51:8. 1, 3, 4, 7, 8, 9, 10

[DJM12] DU H., JIN X., MAO X.: Digital camouflage images using two-scale decomposition. *Computer Graphics Forum 31*, 7 (2012), 2203–2212. 1, 3, 4, 7, 8, 9, 10, 15

[DS02] DECARLO D., SANTELLA A.: Stylization and abstraction of photographs. *ACM Transactions on Graphics 21*, 3 (2002), 769–776. 3

[EG01] ELDER J. H., GOLDBERG R. M.: Image editing in the contour domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 3 (2001), 291–296. 5

[GSP*07] GAL R., SORKINE O., POPA T., SHEFFER A., COHEN-OR D.: 3d collage: Expressive non-realistic modeling. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering* (2007), NPAR '07, pp. 7–14. 2

[HP07] HUANG L., PASHLER H.: A boolean map theory of visual attention. *Psychological Review 114*, 3 (2007), 599–631. 1

[HTM12] HATA M., TOYOURA M., MAO X.: Automatic generation of accentuated pencil drawing with saliency map and lic. *the Visual Computer 28*, 6-8 (2012), 657–668. 3

[HTP07] HUANG L., TREISMAN A. M., PASHLER H.: Characterizing the limits of human visual awareness. *Science 317*, 5839 (2007), 823–825. 1

[HZZ11] HUANG H., ZHANG L., ZHANG H.-C.: Arcimboldo-like collage using internet images. *ACM Transactions on Graphics 30*, 6 (2011), 155:1–155:8. 2

[IKN98] ITTI L., KOCH C., NIEBUR E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259. 2, 3, 5

[MXW*13] MA L.-Q., XU K., WONG T.-T., JIANG B.-Y., HU S.-M.: Change blindness images. *IEEE Transactions on Visualization and Computer Graphics 19*, 11 (2013), 1808–1819. 3

[OBF*14] OWENS A., BARNES C., FLINT A., SINGH H., FREEMAN W.: Camouflaging an object from many viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition(Oral presentation)* (2014), pp. 1–8. 3

[Pal99] PALMER S. E.: *Vision science: Photons to phenomenology*, vol. 1. MIT press Cambridge, MA, 1999. 4

[STRA10] SAEED K., TABĘDZKI M., RYBNIK M., ADAMSKI M.: K3m: A universal algorithm for image skeletonization and a review of thinning techniques. *International Journal of Applied Mathematics and Computer Science 20*, 2 (2010), 317–335. 6

[TBL*09] TROSCIANKO T., BENTON C. P., LOVELL P. G., TOLHURST D. J., PIZLO Z.: Camouflage and visual perception. *Philosophical Transations of The Royal Society 364*, 1516 (2009), 449–461. 4, 6

[TC89] TEH C. H., CHIN R. T.: On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 8 (1989), 859–872. 6

[TG80] TREISMAN A. M., GELADE G.: A feature-integration theory of attention. *Cognitive Psychology 12*, 1 (1980), 97–136. 1

[TZHM11] TONG Q., ZHANG S.-H., HU S.-M., MARTIN R. R.: Hidden images. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering* (2011), NPAR '11, pp. 27–34. 1, 3, 4, 8, 10, 11

[WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. *ACM Transactions on Graphics 25*, 3 (2006), 1221–1226. 6

[Wol94] WOLFE J. M.: Guided search 2.0: A revised model of visual search. *Psychonomic bulletin and review 1*, 2 (1994), 202–238. 1

[YLK08] YOON J.-C., LEE I.-K., KANG H.: A hidden-picture puzzles generator. *Computer Graphics Forum 27*, 7 (2008), 1869–1877. 1, 3

[ZMJ*09] ZHAO H., MAO X., JIN X., SHEN J., WEI F., FENG J.: Real-time saliency-aware video abstraction. *The Visual Computer 25*, 11 (2009), 973–984. 3

(a) Our method (b) [DJM12] (c) Answer

Figure 14: *Failure case.*

Figure 15: *The answers of the hidden images in Fig 12*