Camouflage Images

Hung-Kuo Chu[†] Wei-Hsin Hsu[†] Niloy J. Mitra[⋆] Daniel Cohen-Or[§] Tien-Tsin Wong[‡] Tong-Yee Lee[†]

[†]National Cheng Kung Univ., Taiwan [⋆]IIT Delhi [§]Tel Aviv Univ. [‡]The Chinese Univ. of Hong Kong

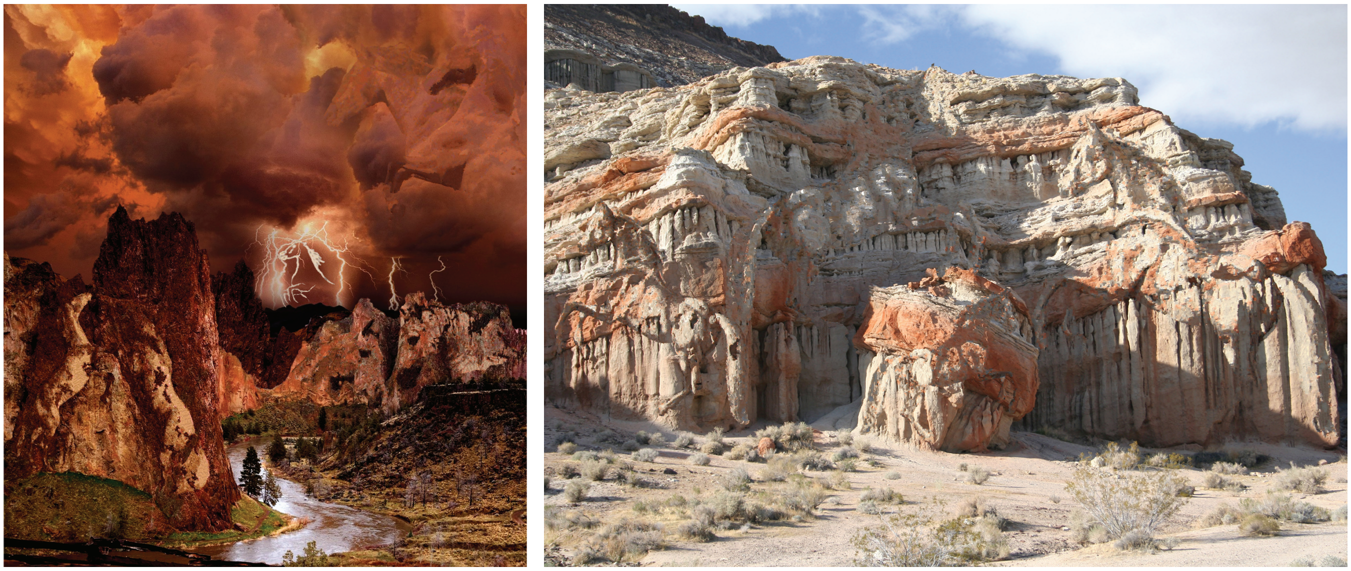


Figure 1: *Two camouflage images produced by our technique. The left and right images have seven and four camouflaged objects, respectively, at various levels of difficulty. By removing distinguishable elements from the camouflaged objects we make feature search difficult, forcing the viewers to use conjunction search, a serial and delayed procedure. (Please zoom in for a better effect. Answer keys are on the last page.)*

Abstract

Camouflage images contain one or more hidden figures that remain imperceptible or unnoticed for a while. In one possible explanation, the ability to delay the perception of the hidden figures is attributed to the theory that human perception works in two main phases: feature search and conjunction search. Effective camouflage images make feature based recognition difficult, and thus force the recognition process to employ conjunction search, which takes considerable effort and time. In this paper, we present a technique for creating camouflage images. To foil the feature search, we remove the original subtle texture details of the hidden figures and replace them by that of the surrounding apparent image. To leave an appropriate degree of clues for the conjunction search, we compute and assign new tones to regions in the embedded figures by performing an optimization between two conflicting terms, which we call *immersion* and *standout*, corresponding to hiding and leaving clues, respectively. We show a large number of camouflage images generated by our technique, with or without user guidance. We have tested the quality of the images in an extensive user study, showing a good control of the difficulty levels.

1 Introduction

Camouflage image, also referred to as hidden image, is an instance of recreational art. Such an image contains one or more hidden figures, or foregrounds, that are typically embedded into a busy apparent image, or background, and remain imperceptible for a while. It takes targeted effort and conscious focus from the viewer to detect the hidden figures. Figure 1 shows two camouflage images generated using our algorithm.

The ability of hiding is closely related to how human perception works. A possible explanation is the feature integration theory, originally proposed by Anne Treisman [1980]. It suggests that human vision and perception work in two main phases: *feature search* and *conjunction search* [Treisman 1988; Wolfe 1994]. Feature search, a largely parallel phase, looks out for characteristic entities like color, edge, texture for quick and instantaneous characterization of figures. In contrast, conjunction search is a serial and slow process, and is responsible for recognition and classification by integrating (scattered) clues from multiple features.

In other words, camouflage images intelligently foil our feature

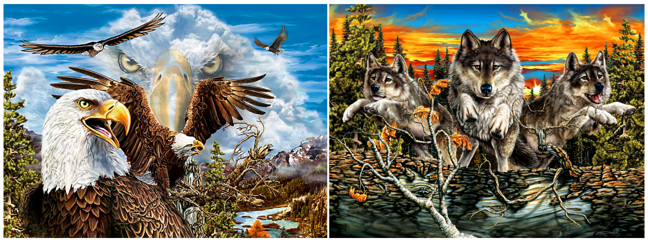


Figure 2: *Artist created camouflage images: (left) 15 eagles, and (right) 14 wolves (© 2010 Steven Michael Gardner. All rights reserved, http://gardnergallery.com/).*

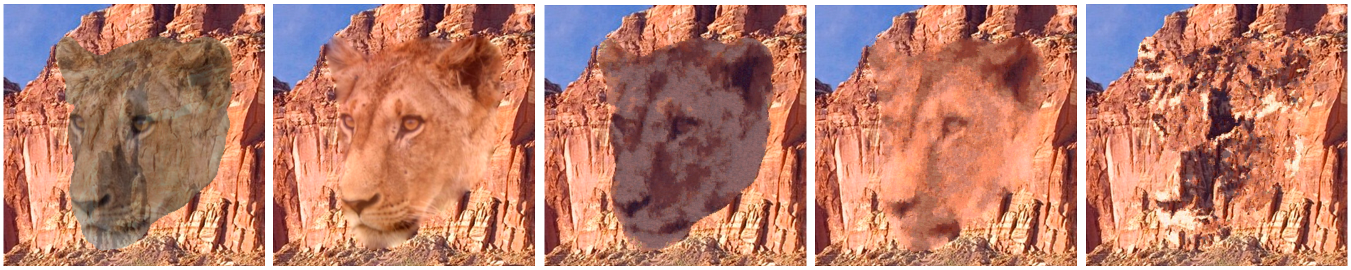


Figure 3: *Results of camouflaging a lion onto a mountain backdrop using various methods: (left to right) alpha blending, Poisson cloning [Pérez et al. 2003], texture transfer [Efros and Freeman 2001], Poisson cloning followed by texture transfer, and our method.*

search mechanism by dressing embedded figures with subtle texture of surrounding background, while leaving us clues, which require integration, for the conjunction search. This suggests a mechanism to create camouflage images by masking out any dominant information, while leaving enough cues in the form of secondary details to aid recognition under focused attention. Effective camouflage images make feature based recognition difficult or impossible, and thus force the recognition process to employ conjunction search. The resultant images are typically busy with the richness of color and texture attracting most of the initial attention of the viewer.

As an art form, camouflage images have been mastered by only a few like Steven Michael Gardner, Bev Doolittle, Jim Warren, and Donald Rust (see Figure 2). When designed at a suitable difficulty level, such images are fascinating to look at. The artist should strike a delicate balance between removing easily identifiable features while simultaneously leaving enough subtle cues for the objects to reveal themselves under a closer scrutiny. Manually creating such images is challenging for most users. We design computational tools to facilitate the process.

Contribution. We present an algorithm for creating camouflage images at controllable levels of difficulty. It assists users, both amateur and advanced, to easily compose and create camouflage images. The proposed method mimics how skilled artists foil our feature search and leave clues for our conjunction search. To deceive our feature search, the original subtle texture details of the foreground figures are removed and replaced by that of the surrounding background. To leave an appropriate degree of clues for our conjunction search, we compute and assign new tones to regions in the embedded figures by performing an optimization consisting of two conflicting terms, which we call *immersion* and *standout*, corresponding to hiding and leaving clues, respectively. Figure 1 shows two hidden images generated with our tool. Using our technique we generated a wide range of examples, with or without user guidance, and widely tested the results with an extensive user study.

2 Related Work

Our work relates to a wide range of topics in computer graphics including visual perception, non-photorealistic rendering (NPR) and texture synthesis. Existing literature targeted towards recreational art is diverse and vast [Kaplan and Salesin 2000; Oliva et al. 2006; Xu and Kaplan 2007; Yen et al. 2008; Chi et al. 2008; Mitra and Pauly 2009]. We primarily focus on works that closely relate to our problem of creating camouflage images.

Visual Perception. Our camouflage image synthesis algorithm is inspired by visual perception research in cognitive psychology [Treisman and Gelade 1980; Treisman 1988; Wolfe 1994; Huang and Pashler 2007; Huang et al. 2007]. According to these studies in psychophysics, *feature search*, a parallel and fast process for classifying objects is believed to be the first stage of visual perception. In this stage cues like color, edge orientation, texture and brightness are used for classification. Objects that can be grouped based on any single feature are partitioned in this stage. With this phenomenon in mind, we selectively remove such feature cues in camouflage images to ensure that a useful saliency map cannot be formed, thus forcing the viewer to perform *conjunction search*. This serial and slower stage requires focused attention to integrate features of several dimensions into a unitary object, and holds the key to creation of successful and engaging camouflage images. Therefore, we have to retain sufficient hints or cues in camouflage images to ensure that the embedded objects are perceived under a closer scrutiny.

Another possible explanation of perceiving camouflage image comes from Boolean map theory [Huang and Pashler 2007; Huang et al. 2007], which separates the perception process into *selection* and *access* phases. A likely perceptual explanation for the connection with our algorithm is a series of Boolean operations leaves only the scaffold of the embedded object. Then the observer has to recognize the object by perceiving the shape of the scaffold as a whole.

However, the above two hypotheses remain to be validated through carefully designed experiments. Importantly, all the existing theories relating to visual perception believe that under most cases, only one feature can be consciously accessed at any one time [Huang et al. 2007].

NPR Rendering. Gal et al. [2007] design a 3D collage system to mimic Arcimboldo's paintings by creating 3D compound shapes of natural elements. Although such paintings illustrate another style of camouflage, they are not characterized by masking out dominant information while leaving enough cues in the form of secondary details to aid recognition of embedded objects. To generate hidden-picture puzzles, Yoon et al. [2008] use NPR stylized line drawing to render the image of the background and objects, and then aim at finding suitable places to hide small objects. Instead of hiding small line art objects, we aim at hiding large objects into the color and highly textured background. Mitra et al. [2009] present a synthesis technique to generate emergence images of 3D objects that are detectable by humans, but difficult for an automatic algorithm to recognize. Our camouflage image can be regarded as its counterpart, but we explore a different dimension of human recognition.

Texture Synthesis. Over the past few years, researchers have made significant advances in seamless texture mapping. Pérez et al. [2003] introduce Poisson image editing to seamlessly insert objects with complex outlines into a new background, regardless of the drastic differences between the source and the destination. Fang and Hart [2004] propose Textureshop to create an embossed result by applying the recovered surface normals from one image to another. Sun et al. [2005] propose an image completion scheme to fill in missing pixels in a large unknown region of an image in a visually plausible manner. Recently, Ritter et al. [2006] present

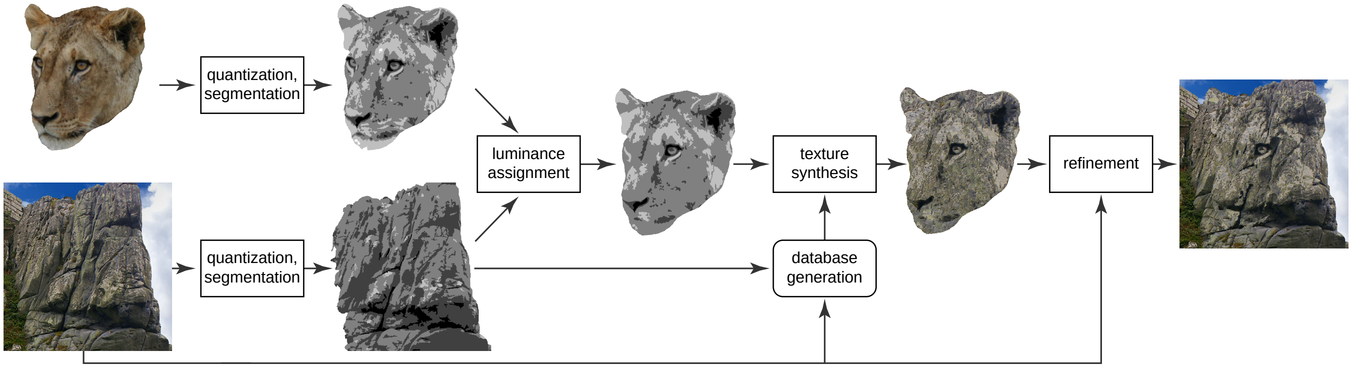


Figure 4: *System overview.*

a texture painting system to combine strokes of texture and produce natural-looking boundary effects. Although texture synthesis techniques provide simple means to immerse and hide figures, they cannot easily control the recognition difficulty. Naïve application of texture synthesis results in obvious-to-recognize camouflage images as demonstrated in Figure 3.

3 Overview

The main task in generating a camouflage image is hiding a foreground figure, while maintaining some degree of its recognition. Our approach is to remove features that allow fast recognition (feature search) and retain other features for slow recognition (conjunction search). Figure 4 overviews the workflow and main components of our camouflage image synthesis. It involves two major goals, leaving clues for conjunction search while foiling the feature search (from left to right).

Any texture distinction between the foreground and background supports fast recognition. Thus, to foil the feature search, we take away the original texture details from the embedded foreground and replace them with the texture details from the background image. To this end, texture synthesis is adopted, however, care must be taken to preserve the features that are essential to conjunction search. We propose a tailor-made two-stage texture synthesis scheme for subtle modifications of edge features. First, to help in preserving the essential edge features, we build a scaffold along the major edge structure of the embedded figure, by copying texture samples with similar edge statistics from the background. Then, we fill in the interior regions using standard texture synthesis.

Palmer [1999] reported that (partial) information carried by only the luminance channel provides sufficient visual cues for a range of recognition tasks. This is also evidenced by our ability to identify objects in grayscale cartoons. The rough luminance distribution is obtained by quantizing and segmenting the grayscale foreground into multiple segments. To introduce control over the recognition difficulty, we assign new tones to these segments by optimizing an objective function of two terms, *immersion* and *standout*. The former immerses the foreground into the background by choosing luminance values that are close to the luminance distribution of background. The latter distinguishes the foreground by preserving the original luminance distribution of foreground. We formulate this tone assignment as a multi-label graph cut on a graph constructed from the images.

4 Algorithm

Given a background image I_B and a foreground object image I_F with its desired position and orientation in I_B, our algorithm embeds and camouflages I_F into I_B at a controllable level of recognition difficulty. Here, we describe the key steps of our algorithm in detail (see Figure 4).

Luminance assignment. We retain the rough luminance distribution of foreground as the clue for conjunction search. The luminance distribution is obtained by grayscale-conversion, quantization, and segmentation of the foreground. This gives us a set of segments of different luminance values. Similarly, we obtain the luminance distribution of the background image. With these segments as containers and luminance values as the intensity guide, we can already dress up the foreground segments by the texture details of background segments using standard texture synthesis. However, the foreground and the background may have substantially different luminance distributions, that lead to trivial recognition. To remove this distinction, we can *re-color* the foreground segment map with the luminance distribution from the background (*immersion*). A naïve replacement of luminance distribution, on the other hand, can completely hide the foreground, and make it unrecognizable. In other words, we need to make the foreground standout by preserving its original luminance distribution (*standout*). Hence we aim for a balance between two conflicting factors: *immersion*, which measures the similarity of luminance distribution between I_F and I_B, and *standout*, which measures the deviation from the original luminance distribution of foreground I_F. We optimize an objective function of these two terms on a graph constructed from the foreground and background segment maps.

Quantization and segmentation: The images are quantized based on their luminance values, and segmented into connected segments using a simple flood-fill algorithm. The background segments are cropped by foreground in overlapped regions. We denote the resulting segmented foreground and background images as $Q_F = \{q_i^F | i = 1, \dots n\}$ and $Q_B = \{q_i^B | i = 1, \dots m\}$, where q_i^F and q_i^B represent connected segments in I_F and I_B, respectively. Notation q_i is used to denote a segment, in general. Let $L_F = \{l_i^F | i = 1, \dots n\}$ and $L_B = \{l_i^B | i = 1, \dots m\}$ be the luminance values of the corresponding segments.

Graph construction: Next, we construct a graph $G = (V, E)$ where the nodes V correspond to the segments in $Q_F \cup Q_B$. Two nodes q_i and q_j are connected by an edge $e_{ij} \in E$, if either of the following is true: (1) both $q_i, q_j \in Q_F$ and they share a partial boundary, or (2) $q_i \in Q_F$, $q_j \in Q_B$ and $q_j \in N(q_i)$ where $N(q_i)$ represents k-nearest neighbors* of q_i (in all our experiments, we set $k = 6$). The edges e_{ij} are respectively called *standout edges* and *immersion edges*. The luminance values l_i are kept in the corresponding nodes. Values l_i^B of nodes corresponding to segments Q_B are known. To camouflage the foreground into background, our goal is to assign

* The k-nearest neighbors are computed based on the closest Euclidian distance between two segments in a camouflage image.

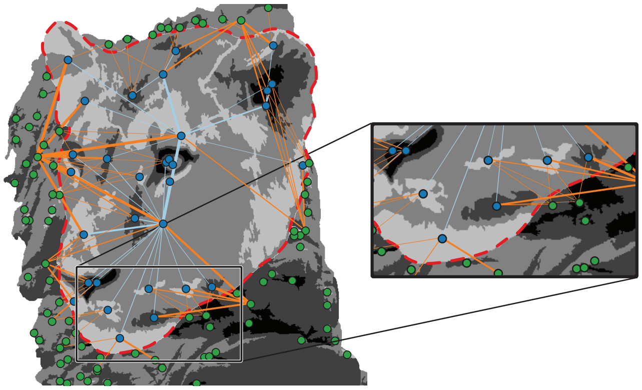


Figure 5: *Construction of luminance graph. For visualization, the background and the foreground are separated by a dotted red curve. Foreground and background segments are denoted by blue and green nodes, respectively. Immersion edges are shown in orange and standout edges in light blue, with their thickness being proportional to their relative weights, i.e., w_I (Equation 1) and w_S (Equation 2), respectively. We use $k = 3$ for this example.*

a set of new luminance values $L'_F = \{l'_i | i = 1, \ldots, n\}$, picked from L_B, to nodes in Q_F to balance the immersion and standout energies. Figure 5 shows such a graph.

The *immersion energy* measures how well the foreground *immerses* into the background, and gives a quality indication of the camouflage effect. As segments may have different sizes and shapes, we account for such differences by weights as:

$$E_I(L'_F) = \sum_{q_i^F \in Q_F} \sum_{q_j^B \in N(q_i^F)} (l'_i - l_j^B)^2 \, w_I(q_i^F, q_j^B), \qquad (1)$$

where,

$$w_I(q_i^F, q_j^B) = \begin{cases} b(q_i^F, q_j^B) a(q_j^B) & \text{if } q_i^F \text{ and } q_j^B \text{ share boundary} \\ a(q_j^B) \exp(-d(q_i^F, q_j^B)/\sigma) & \text{otherwise,} \end{cases}$$

with $b(q_i^F, q_j^B)$ being the length of the shared boundary between q_i^F and q_j^B normalized with respect to the boundary length of q_i^F, while $a(q_i^B)$ returns the area of component q_i normalized with respect to the image area. The closest Euclidean distance between the components q_i^F and q_j^B is denoted by $d(q_i^F, q_j^B)$ with σ being 0.1 of the bounding box length of the entire foreground area.

The *standout energy* term measures how well the new luminance values of the foreground components retain its original differences of luminance. We define it as:

$$E_S(L'_F) = \sum_{\substack{q_i, q_j \in Q_F \\ e_{ij} \in E}} ((l'_i - l'_j) - (l_i^F - l_j^F))^2 \, w_S(q_i, q_j) \qquad (2)$$

where $w_S(q_i, q_j) = (b(q_i, q_j)a(q_i) + b(q_j, q_i)a(q_j))/2$.

The luminance assignment problem becomes a minimization of the combined energy as

$$L'_F \equiv \{l'_i\} = \arg \min_{l'_i \in Q_B} [\lambda E_I(L'_F) + (1 - \lambda) E_S(L'_F)], \qquad (3)$$

with λ controlling the relative importance between the immersion and standout energies. It also serves as a control parameter of the level of recognition difficulty.

Optimization: The above energy can be minimized using a sparse linear system, if we remove the additional requirement that the values have to be chosen from those in the background segment map. This side constraint results in a NP-hard label assignment problem. If $|V|$ is small, we can obtain an exact solution using a multi-label graph construction [Carr and Hartley 2009]. However, for any realistic inputs, $|V|$ can be easily in the scale of hundreds. Therefore, we seek an approximate solution to this problem. We note that our assignment problem is similar to the labeling problem in computer vision that optimizes an energy function. In this problem, commonly referred to as Bayesian labeling of first-order markov random field (MRF), each pixel is assigned a label from a known set. Therefore, to efficiently solve our labeling problem, we employ a multi-label graph cut algorithm, which uses an $\alpha - \beta$ swap algorithm to simultaneously change the labels of a large set of nodes to achieve good performance [Boykov et al. 2001; Kolmogorov and Zabih 2002; Boykov and Kolmogorov 2004].

Camouflaged foreground synthesis. After leaving desired amount of clues in the luminance-optimized segment map, we then dress up the foreground segments with the texture details of the background, in order to foil the feature search. To avoid the boundary information of the foreground segment map being accidentally destroyed by conventional texture synthesis, we tailor-made a two-stage texture synthesis. It first synthesizes a scaffold along the segment boundaries in foreground in order to preserve the boundary, and then fills up the interior. Both synthesis stages give higher preference to samples from the background region that is going to be covered by the foreground than from the rest of the background image. Note that we also take into account the texture details of cropped background to increase the richness of texture synthesis.

Synthesizing segment boundary: As both the foreground and background have the segment maps, we may copy a small local patch of texture from the background to the foreground to build the scaffold, wherever two local regions, in foreground and in background, have the similar segment maps. This is, in fact, an analogy approach [Hertzmann et al. 2001]. Figure 6 illustrates this scaffold building process.

We build a database of *patch pairs* to facilitate the search of patches from the background. Each pair contains the local texture patch t_i and the corresponding luminance patch s_i in the segment map. The segment boundaries of background segment map are densely sampled and each sample forms a pair (t_i^B, s_i^B) in the database (Figure 6). Next, we sample along segment boundaries of the optimized foreground segment map. For each sample, we form a query patch pair (t_q^F, s_q^F), and search the best matched patch pair in the database using the following expression:

$$(t_m^B, s_m^B) := \arg \min_{(t_i^B, s_i^B)} [\alpha D(s_i^B, s_q^F) + (1 - \alpha) D(t_i^B, t_q^F)],$$

with α controlling the relative importance between the segment map and texture distances. Initially, the t_q^F's are mostly empty, except those patches that partially overlap with background. As we

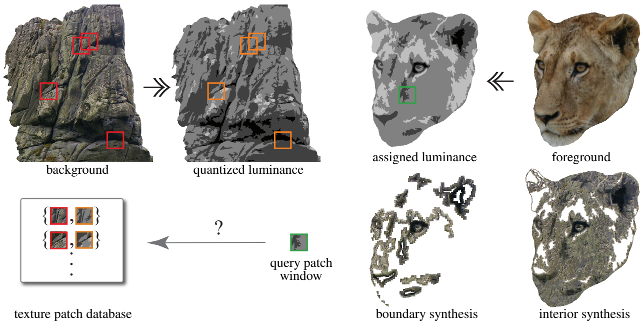


Figure 6: *Analogy approach to construct scaffold. To dress up the foreground along its segment map boundary, texture patches are copied from the background image to form the foreground scaffold, wherever both places have similar patterns in segment maps.*

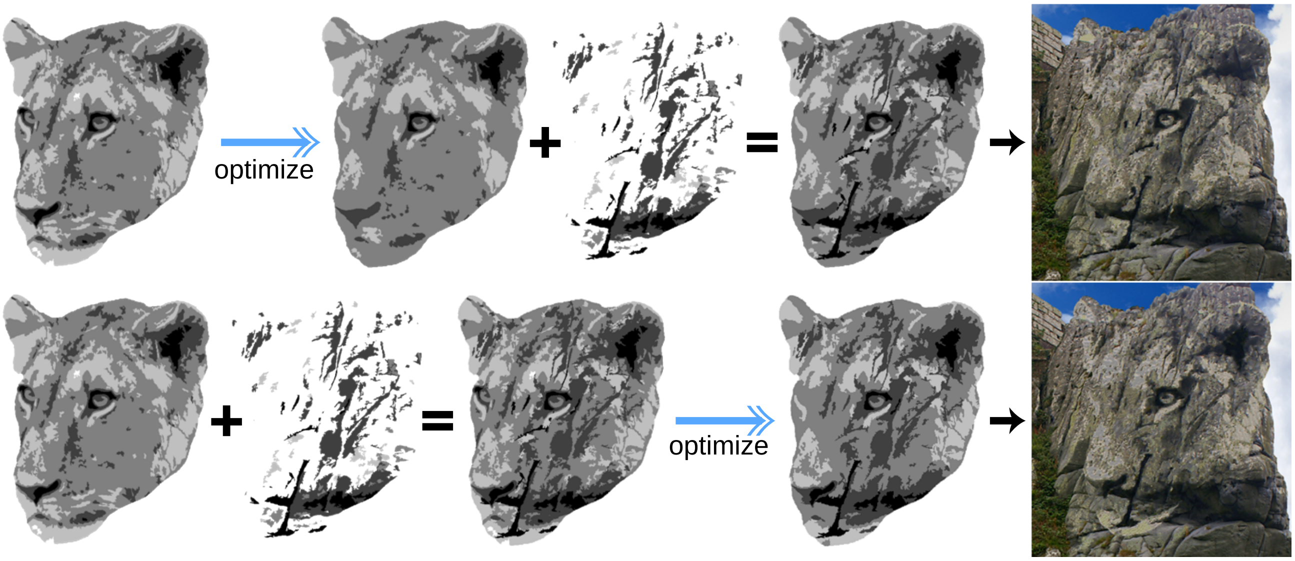


Figure 7: *Effect of adding distracting background segments posterior to luminance optimization. (Top row, Left to right) Segment map L_F, optimized segment map L'_F, distracting background segments D_B, superimposed image S using L'_F and D_B, and camouflage synthesis result using S. (Bottom row, Left to right) Segment map, distracting background segments, superimposed image, luminance optimized segment and resultant camouflage synthesis result.*

progressively fill in texture patches along the boundary, subsequent foreground query patches have a higher chance of non-emptiness. To handle empty query patches, we set $\alpha = 1.0$ when t_q^F is empty; otherwise $\alpha = 0.7$, i.e., when the query lacks any texture information, only the segment map distance is considered. In our implementation, we realize the function D using the sum of normalized squared distances (SSD) between two patches (full region for s_i and partial region for t_i). The texture distance is evaluated in the CIELab color space.

Synthesizing segment interiors: After building the scaffold along segment boundaries, we progressively fill up the interior. Using the luminance information from the optimized foreground segment map, we fill the foreground interior by growing texture from the textured boundary. We copy texture samples from the background using texture-by-numbers techniques [Hertzmann et al. 2001]. Overlapping textures are seamlessly combined by finding an optimized cut path [Kwatra et al. 2003].

Both stages of synthesis require working with large collections of patches in the database, typically 30K-300K. In our experiments, we use patches of size 7×7 leading to large set of vectors of 49 dimensions. A naïve search is slow for interactive applications. To speed up, we reduce the search vector dimensionality by a principal component analysis (PCA) based subspace learning, retain the eigen coefficients with top 80% energy, and use them to k-mean cluster the database into around 1,000 clusters. This leads to about 5-10 times of speedup without any noticeable visual artifact in the synthesized results. Alternatively, it is possible to speed up with GPU-accelerated Gaussian KD-trees [Adams et al. 2009].

5 Enhancements and User Control

We now present additional controls to improve the results using subtle modifications, and also help the user to better explore, compose, and create camouflage images.

Adding distracting segments. We increase the difficulty level of the generated hidden figures by adding additional segments to the luminance-optimized segment map of foreground. Specifically, before the synthesis phase, we add some distracting segments by randomly copying background segments that are either completely or partially overlapped with the foreground and pasting them to the luminance-optimized segment map. These segments help to break the original boundary of foreground segments, making feature search harder. As an undesirable side effect, this random copy-and-paste approach may also disturb critical features such as the eyes and nose of the foreground figure. To prevent this problem, we provide the user a brush tool to indicate critical foreground regions. Subsequently, the system avoids pasting over the marked critical segments. In Figure 7, we used the brush tool in regions around lion's eye. We added distracting segments to most of our results using the same default setting. Specifically, we added filtered segments in ratios 0.5, 0.8, 1.0 for easy, medium, hard difficulty levels, respectively. For any hidden figure in a scene, distractors for easy (setting) are a subset of those of medium, which in turn are a subset of those of hard. Although the distractors can be embedded prior to luminance optimization, as a design choice we decoupled the two stages making the system to be interactive, allowing fine tuning the amount of distractors. In our experiments we observed that for fixed margin of distractors, prior and posterior optimization produce comparable results since the long and thin segments do not significantly influence the optimization(see Figure 7). However, at difficult setting, noticeable difference starts appearing due to the larger-area segments.

Background warping. When the user places the foreground figure near the boundary of the background object, e.g. the contour of mountain, parts of the foreground may protrude out unnaturally, revealing the hidden object. We automatically detect such contour segments with high curvature variation, and then blend them in by warping the background border to fit the figure contour using subtle modifications as proposed by Schaefer et al. [2006] (see Figure 9).

Embedding location search. In our system, we leave the user the final control of positioning the foreground figure. However, to aid the process, we provide two simple search mechanisms for local and global placement of the object. In the local mode, we refine the initial user-specified position by determining the translation and rotation that minimizes the energy function in Equation 3. In our implementation, we constrain the translation to ±1% of the image diagonal length along x and y directions, while the angle is con-

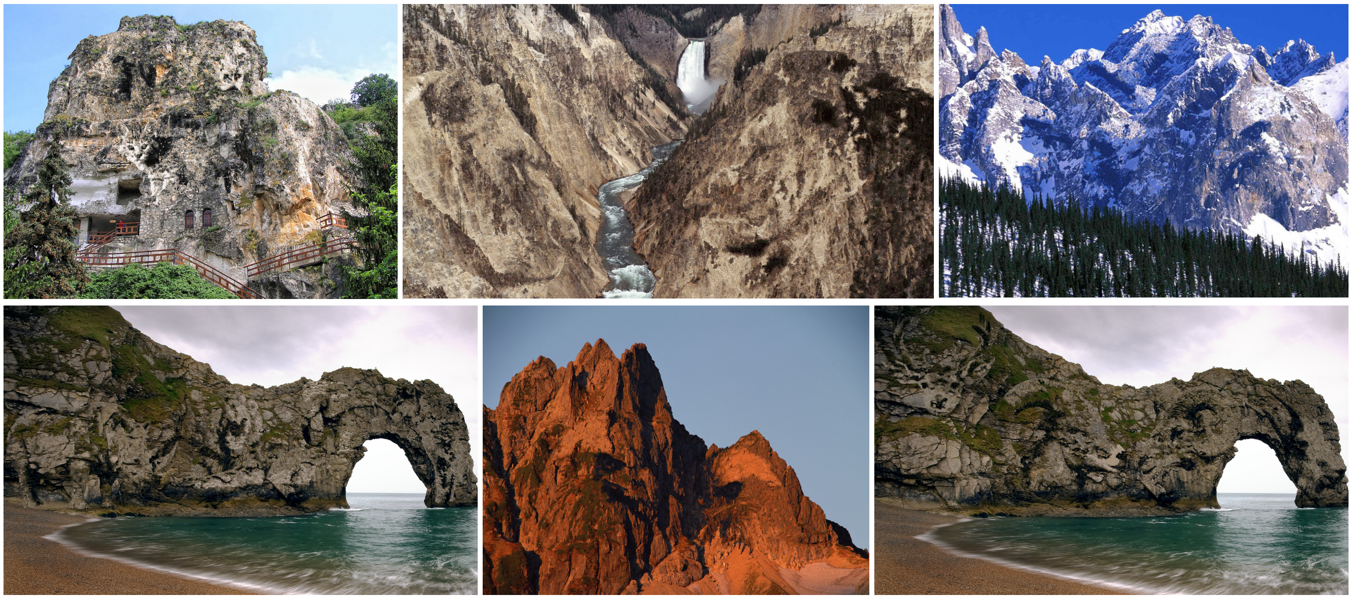


Figure 8: *Six camouflage images created by our algorithm. Additional information about each image is listed in Table 1 (from left to right, top to bottom). Answer keys are on the last page.*

strained to be within ±10 degrees. Gradient-descent is used for solving the optimization. In the global mode, we present the user a list of candidate positions and orientations in the background image, given a foreground object. Our luminance assignment being fast, we sparsely sample background locations and locally optimize the placements around each sample location. The top ten locations in terms of energy are then presented to the user for selection.

6 Results and Discussion

Creating nice camouflage images is not easy, even for skilled artists. In addition to painting, artists are required to achieve a good balance between hiding and leaving clues while crafting such images. In a possible workflow, the artist may provide a theme and initial layout for objects, while our computational framework produces a camouflage result. Our synthesis algorithm is fast and allows efficient exploration of the camouflage space. Computational times for generating images in Figure 8 on an Intel Core 2 Duo E8400 3.0GHz PC with 4 GB RAM are shown in Table 1. We use five quantization levels for foreground and background images in all our examples, typically leading to a few hundred segments. The synthesis time increases in proportion to the image resolution and number of embedded figures. If the user only adjusts the difficulty level without changing the location, we can cache the patch databases to speed up for interactive modification. Our approach not only provides amusement, but also allows users, both amateurs and professionals, to quickly produce camouflage images. For a better impression of our method and its results, we refer the reader to the supplementary video and user study materials.

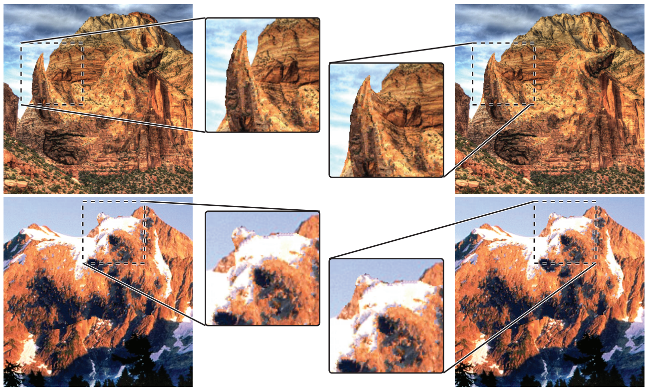


Figure 9: *Background warping to camouflage protrusion of foreground. Note that the rhino's horn (left) and bear's ear (right) protrude out of the background silhouette. (Bottom) Subtle background warping can help hide them naturally.*

The ultimate judge for camouflage images are humans: if they can reliably detect the embedded images, and how captivating they find the images. We conducted an user study with 68 participants over a dataset generated using 13 scenes. Each scene contains one or more hidden figures and is used to produce camouflage images at three difficulty levels using λ as 0.3, 0.6, and 0.8 for difficult, medium, and easy, respectively (see Equation 3).

User study I: recognition time. At the start of the study, participants were instructed using two trial examples. Then 8 scenes, each with one hidden figure, were shown to the participants. Images were produced at three levels of difficulty. Each time, we started from the difficult level and asked users to (roughly) circle the hidden figures and describe what they saw. If user could not recognize any figure and asked for another chance, we proceeded with the same scene, but at a lower difficulty (shown with a blank frame in between). To

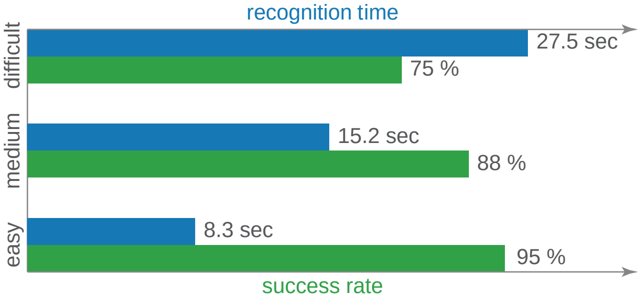


Figure 10: *Recognition time and success rates on three difficulty levels of generated camouflage images as observed in course of our user study (see Section 6).*

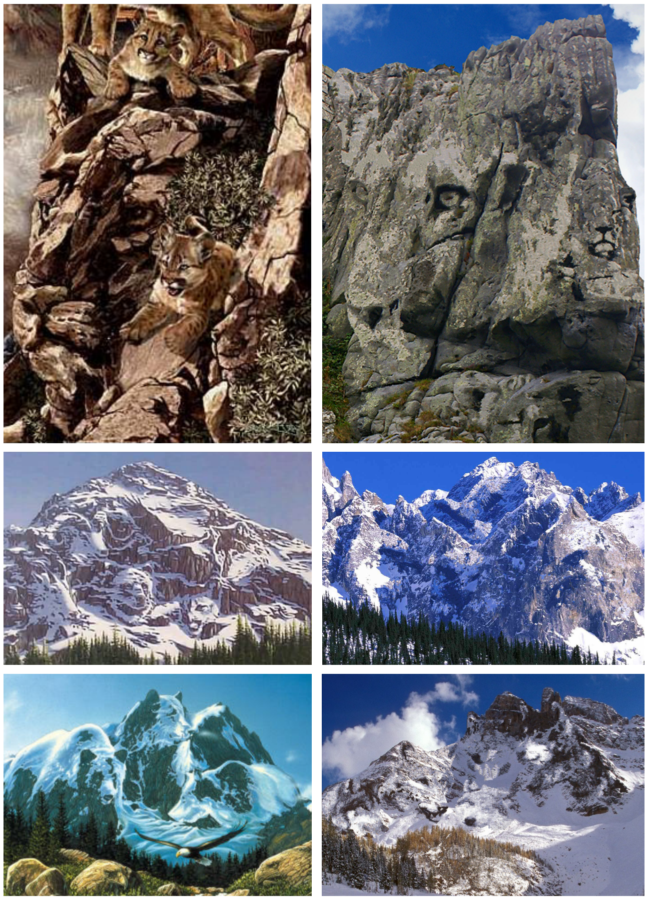


Figure 11: *Comparison of synthesized results (right) with artist generated ones (left) (top: © 2010 Steven Michael Gardner; middle and bottom: © 2010 John Van Straalen. All rights reserved.).*

reduce fatigue after 4 scenes the users were given a short break. The recognition times were recorded counting from when the image is shown until the time when the user started typing what they saw. Figure 10 shows the average recognition times and success rate against difficulty levels. When the participant successfully identified an image, we assume that he/she would have also identified the easier versions. If the user instead of asking for another chance gave a wrong answer, we assume that he/she would have also failed with the easier versions (in absence of a better way to guess, we take a conservative option). We hypothesize that our method can effectively control the difficulty level of produced camouflage images. Using a single factor ANOVA analysis with $\alpha = 0.05$, we obtain p-values as $3e{-}5$ and 0.0028 for recognition time and success rate, respectively, thus verifying our hypothesis.

User study II: comparison with camouflage artwork. In order to get some feedback about the quality of the generated camouflages images, we showed a set of 30 participants 10 camouflage images, 5 of ours and 5 from artists, in random order. Users had no prior knowledge about the source of the images and were asked to rate each image, on a scale of 1 to 5, depending on how natural, seamless, and engaging they found the hidden objects to be. It is very encouraging that our computer-generated camouflage images received an average score of 4.23 comparable to artists' 4.21 among all participants, i.e., comparable within error margins. Figure 11 shows some images used in this user study. This study indicates that our algorithm consistently provides reasonable to good camouflage images starting from user provided foreground-background image pairs. The system is easy to use, and suitable for amateurs.

	image res.	# fig.	optimize	PCA,k-mean	synthesis
2 dogs, 2 cats	383 x 383	4	0.47	5.85	3.05
2 lions	921 x 949	2	1.46	27.3	110
4 eagles	371 x 371	4	0.21	4.36	1.61
Africa wild life	561 x 561	4	0.65	10.07	6.1
2 eagles	531 x 531	2	0.54	13.8	11.3
1 deer, 3 goats	467 x 497	4	0.62	6.06	8.13

Table 1: *Timing of generating camouflage images in Figure 8. The timings, in seconds, are averaged over all hidden figures.*

Limitations. A main component of our algorithm is based on the optimized luminance assignment. It may fail to hide the foreground when the luminance contrast of foreground is fairly low. The foreground may be re-colored with very similar luminance values and hence similar textures from background, resulting in either complete immersion or complete standout (see Figure 12). Interestingly, we found that artists rarely embed low-contrast figures in their camouflage pieces. Another restriction is that embedded figures produce unsatisfactory results across regions with distinctive texture and orientations, e.g. across mountains and sky since the boundary often becomes obvious. In cases when embedded figures are placed slightly across two very different regions, we use background warping to reduce this problem. Finally, our synthesis algorithm cannot guarantee the physical correctness of illumination in the synthetic foreground, which may lead to inconsistency between the foreground and background, specially in presence of strong shadows.

7 Conclusion and Future Work

Camouflage images can be fascinating and entertaining as they challenge viewers to consciously focus to detect the hidden figures. We presented a novel approach to synthesize such images at controllable levels of difficulty. Our user study indicates that the approach can consistently produce results at a level where the viewer can locate embedded figures with focused attention, but not immediately. Although we can create many promising results, our approach is not positioned as a substitute for artists, whose creativity and imagination is invaluable, and unlikely to be replaced computationally. However, we believe that it can be an effective tool for amateurs to create camouflage images as well as for skillful artists for initial design of their artpieces, thereby reducing their production time.

Interesting future work waits to be explored. For example, some artists explore shadows to foil our feature search mechanism. We plan to investigate shape from shading algorithms for obtaining 2.5D information of foreground and background images, leading to shadow based camouflage effects. Recently, Mitra et al. [2009]

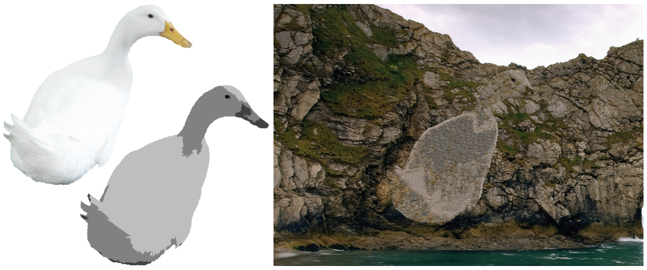


Figure 12: *Limitation: Low contrast foreground image. Left-top: original image. Left-bottom: optimized segment map. Right: the resultant image.*

present a synthesis technique from 3D objects to generate noisy emergence images that is potentially useful for designing an effective Captcha mechanism. Although conceptually our camouflage images can be considered as its counterpart, it is worth exploring the possibility of using our technique to generate photo-realistic pleasing emergence images. Finally, we hope our technique can be used for generating camouflage images at controlled levels of difficulty for studies in cognitive psychology, helping to better understand how humans perceive images.

Acknowledgements

This work is supported in part by the Landmark Program of the NCKU Top University Project (contract B0008), the National Science Council (contracts NSC-97-2628-E-006-125-MY3 and NSC-96-2628-E-006-200-MY3) Taiwan, the Israel Science Foundation and the Research Grants Council of the Hong Kong SAR under General Research Fund (CUHK417107). Niloy was partially supported by a Microsoft Outstanding Young Faculty Fellowship. We thank Chung-Ren Yan for his valuable comments on texture synthesis, Kun-Chuan Feng for helping to design the user study, and Johathan Balzer for the video voice-over. We sincerely thank all the participants of the user study for their time and useful feedback. We thank Steven Michael Gardner, John Van Straalen for granting permission to use their artworks, and Aleta A. Rodriguez, Krum Sergeev, Alexander Stross and Joel Antunes for the photographs used in our examples. Finally, we are grateful to the anonymous reviewers for their comments and suggestions.

References

ADAMS, A., GELFAND, N., DOLSON, J., AND LEVOY, M. 2009. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3, 1–12.

BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 9, 1124–1137.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11, 1222–1239.

CARR, P., AND HARTLEY, R. 2009. Solving multilabel graph cut problems with multilabel swap. *Digital Image Computing: Techniques and Applications 0*, 532–539.

CHI, M.-T., LEE, T.-Y., QU, Y., AND WONG, T.-T. 2008. Self-animating images: Illusory motion using repeated asymmetric patterns. *ACM Trans. on Graphics (Proc. SIGGRAPH) 27*, 3, 62:1–62:8.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proc. of ACM SIGGRAPH*, 341–346.

FANG, H., AND HART, J. 2004. Textureshop: texture synthesis as a photograph editing tool. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3, 354–359.

GAL, R., SORKINE, O., POPA, T., SHEFFER, A., AND COHEN-OR, D. 2007. 3D collage: Expressive non-realistic modeling. In *Proc. of NPAR*, ACM, 14.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proc. of ACM SIGGRAPH*, 327–340.

HUANG, L., AND PASHLER, H. 2007. A boolean map theory of visual attention. *Psychological Review 114*, 599–631.

HUANG, L., TREISMAN, A., AND PASHLER, H. 2007. Characterizing the limits of human visual awareness. *Science 317*, 5839, 823.

KAPLAN, C. S., AND SALESIN, D. H. 2000. Escherization. In *Proc. of ACM SIGGRAPH*, 499–510.

KOLMOGOROV, V., AND ZABIH, R. 2002. What energy functions can be minimized via graph cuts? In *ECCV*, 65–81.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. on Graphics (Proc. SIGGRAPH) 22*, 3, 277–286.

MITRA, N. J., AND PAULY, M. 2009. Shadow art. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 5, 1–7.

MITRA, N. J., CHU, H.-K., LEE, T.-Y., WOLF, L., YESHURUN, H., AND COHEN-OR, D. 2009. Emerging images. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 5, 1–8.

OLIVA, A., TORRALBA, A., AND SCHYNS, P. G. 2006. Hybrid images. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3, 527–532.

PALMER, S. 1999. *Vision Science: Photons to Phenomenology*. MIT Press.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 22*, 3, 313–318.

RITTER, L., LI, W., CURLESS, B., AGRAWALA, M., AND SALESIN, D. 2006. Painting with texture. In *Eurographics Symposium on Rendering*.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3, 533–540.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3, 861–868.

TREISMAN, A., AND GELADE, G. 1980. A feature-integration theory of attention. *Cognitive Psychology 12*, 1, 97–136.

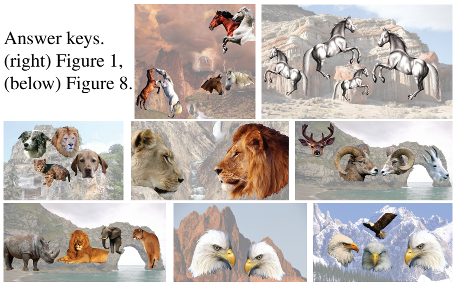
TREISMAN, A. 1988. Features and objects: the fourteenth bartlett memorial lecture. *Quarterly Journal of Experimental Psychology 40A*, 201–236.

WOLFE, J. M. 1994. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review 1*, 2, 202–238.

XU, J., AND KAPLAN, C. S. 2007. Image-guided maze construction. *ACM Trans. on Graphics (Proc. SIGGRAPH) 26*, 3, 29.

YEN, C.-R., CHI, M.-T., LEE, T.-Y., AND LIN, W.-C. 2008. Stylized rendering using samples of a painted image. *IEEE Trans. on Vis. and Comp. Graphics 14*, 468–480.

YOON, J., LEE, I., AND KANG, H. 2008. A Hidden-picture Puzzles Generator. In *Comp. Graphics Forum*, vol. 27, 1869–1877.



Answer keys. (right) Figure 1, (below) Figure 8.