Deep Camouflage Images

Qing Zhang,[1] **Gelin Yin,**[1] **Yongwei Nie,**[2] **Wei-Shi Zheng,**[1,3,4*]

[1]School of Data and Computer Science, Sun Yat-sen University, China,
[2]School of Computer Science and Engineering, South China University of Technology, China
[3]Peng Cheng Laboratory, Shenzhen 518005, China
[4]The Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
zhangqing.whu.cs@gmail.com, gelinyin@gmail.com, nieyongwei@scut.edu.cn, wszheng@ieee.org

Abstract

This paper addresses the problem of creating *camouflage images*. Such images typically contain one or more hidden objects embedded into a background image, so that viewers are required to consciously focus to discover them. Previous methods basically rely on hand-crafted features and texture synthesis to create camouflage images. However, due to lack of reliable understanding of what essentially makes an object recognizable, they typically result in either complete standout or complete invisible hidden objects. Moreover, they may fail to produce seamless and natural images because of the sensitivity to appearance differences. To overcome these limitations, we present a novel neural style transfer approach that adopts the visual perception mechanism to create camouflage images, which allows us to hide objects more effectively while producing natural-looking results. In particular, we design an attention-aware camouflage loss to adaptively mask out information that make the hidden objects visually standout, and also leave subtle yet enough feature clues for viewers to perceive the hidden objects. To remove the appearance discontinuities between the hidden objects and the background, we formulate a naturalness regularization to constrain the hidden objects to maintain the manifold structure of the covered background. Extensive experiments show the advantages of our approach over existing camouflage methods and state-of-the-art neural style transfer algorithms.

Introduction

Camouflage is a concept that stems from the biology. To survive in the wild, animals have developed an ability to hide themselves from predators and preys by making colors and textures on their bodies similar to those of their natural habitats. Artists gain inspiration from this phenomenon and create a new art form referred to as camouflage images. These images usually contain one or more hidden figures or objects that remain imperceptible to viewers for a while, unless under a closer scrutiny (see Fig. 1 for examples).

Camouflage images are widely favored as detecting hidden objects from these images can be challenging and entertaining. However, creating camouflage images is difficult, even for skilled artists, since fooling eyes with hidden objects involves reliable interpretation of how human visual

Figure 1: Two camouflage images produced by our approach. Please zoom in to see details and try to find the hidden objects. Answer keys are shown in Fig. 10.

perception works. Fortunately, the feature integration theory proposed in (Treisman and Gelade 1980) provides a possible explanation. It suggests that the perception mechanism of human vision system can be explained as a two-phase visual search process including *feature search* and *conjunction search* (Treisman 1988; Wolfe 1994). Specifically, feature search is an instant procedure in which we *subconsciously* perceive the scene semantics by leveraging intuitive visual features such as color and texture. While conjunction search is a slow and delayed procedure, since it requires us to integrate scattered multiple features and perform inference to achieve recognition. This theory explains why understanding camouflage images takes effort. Concretely, the reason is that camouflage images foil our feature search by decorating the hidden objects with colors and textures similar to the background, and thus force our perception system to employ the slow conjunction search for recognition.

Currently, there is a paucity of literature on creating camouflage images. Based on the feature integration theory, Chu et al. (Chu et al. 2010) presented the first computational approach for embedding 2D objects into an image, where they designed an optimization consisting of two conflicting terms corresponding to preventing feature search and allowing conjunction search, respectively. Although this approach produces promising results, it is sensitive to appearance differences between the hidden objects and the background image, and may fail due to lack of reliable understanding of what makes an object recognizable. Unlike (Chu et al. 2010), Owens et al. (Owens et al. 2014) introduced a method to completely hide a 3D object from multiple viewpoints. Despite the success of this technique, it is inapplicable in cre-

Figure 2: Overview of our approach. Given the input foreground and background images, we first recommend hiding region for the foreground object, then a camouflage image generation step is implemented to seamlessly camouflage the object into the background image to obtain the output camouflage image.

ating camouflage images that we discuss in this paper, since valid camouflage images should leave enough subtle clues for viewers to discover the hidden objects.

Unlike existing methods which depend on hand-crafted features, we introduce the first deep-learning approach for creating camouflage images. Our approach builds upon the recent work on neural style transfer (Gatys, Ecker, and Bethge 2016), which achieves impressive results by separating style from the content of an image using feature representations learned by a neural network. However, as demonstrated in Fig. 5, by transferring style of the background image to the hidden objects while preserving a certain level of the objects' content using existing neural style transfer algorithms has two limitations in creating camouflage images: (1) The hidden objects will be either complete standout or complete invisible, even with a dedicated balance between the content and style. (2) There often exists obvious appearance discontinuities between the hidden objects and the background, making the results visually unnatural.

Contributions: (1) Distinct from existing work on neural style transfer, we develop a novel loss function to generate camouflage images that avoid the above two limitations. Particularly, we design an attention-aware camouflage loss to more effectively foil our feature search and leave clues for conjunction search. In addition, we formulate a naturalness regularization to remove appearance discontinuities between the hidden objects and the background by maintaining the manifold structure of the covered background within the hidden objects. (2) To aid amateurs create camouflage images, we present a simple yet effective method to automatically recommend viable region to hide an object within the given background image. (3) We conduct extensive experiments to evaluate the proposed approach and compare it with various methods. Results show that camouflage images generated by our algorithm are more preferred by human subjects.

Related Work

Computational Camouflage. There have been some works that address camouflage problems by computational methods. For example, Chu et al. (Chu et al. 2010) presented an optimization for embedding 2D hidden objects into an image based on the feature integration theory (Treisman and

Gelade 1980). While this approach produces impressive results, it may fail due to lack of reliable understanding of objectness and the sensitivity to illumination differences. Owens et al. (Owens et al. 2014) proposed to completely hide a 3D object from multiple viewpoints, which differs our work since we aim to hide 2D objects with remaining clues to detect them. Some other works focus on breaking the camouflage by revealing the hidden patterns or objects (Tankus and Yeshurun 2001; Reynolds 2011).

Texture Synthesis. There are numerous works tackling the texture synthesis problem (Heeger and Bergen 1995; Efros and Freeman 2001; Gatys, Ecker, and Bethge 2015; Ulyanov et al. 2016; Ulyanov, Vedaldi, and Lempitsky 2017; Li et al. 2017b). These techniques provide a simple way to hide objects within an image by dressing them with synthesized background consistent textures. However, they fail to enable control of the recognition difficulty of the hidden objects. As a result, naively applying texture synthesis would yield obvious-to-recognize camouflage images (see Fig. 4).

Neural Style Transfer. Style transfer has been brought to a new era since the pioneer work of Gatys et al. (Gatys, Ecker, and Bethge 2016), which proposed to separately manipulate the style and content of an image by leveraging the feature maps of discriminatively trained deep convolutional neural networks such as VGG-19 (Simonyan and Zisserman 2014). This work has been widely studied, and various subsequent methods were developed (Li and Wand 2016; Johnson, Alahi, and Fei-Fei 2016; Huang and Belongie 2017; Gatys et al. 2017; Luan et al. 2017; Chen et al. 2017; Liao et al. 2017; Li et al. 2017a; 2018; Gu et al. 2018) (see (Jing et al. 2019) for a review). However, a direct application of them usually results in unsatisfactory camouflage images, since they basically treat all features equally and thus may either fail to foil our feature search or fail to leave sufficient clues for conjunction search. Moreover, as claimed in (Luan et al. 2018), these methods typically perform poorly on local style transfer. A recent work on deep painterly harmonization (Luan et al. 2018) allows seamlessly pasting an object into a painting as if it is originally painted. Our work differs it in two aspects. First, we focus on camouflaging rather than compositing an object into an image. Second, our method can handle various image types, not just paintings.

Methodology

To simplify the description, we assume that there is only a single object to be hidden, though our approach can work with multiple hidden objects. Given a background image I_B, a foreground image I_F and its corresponding object mask M_F, our algorithm aims to camouflage the object in I_F into I_B. The workflow of the proposed algorithm is illustrated in Fig. 2. Overall, our approach is comprised of two steps, including: (1) Hiding region recommendation and (2) Camouflage image generation. Specifically, it begins by recommending a hiding region (indicated by mask M_B) for the given hidden object, and then generates the desired camouflage image I_R. In the following subsections, we first describe the details of each of the two components in our approach, and then elaborate the implementation details.

Hiding Region Recommendation

We recommend hiding region (in I_B) for the foreground object in I_F, since we found that the quality of hiding region determines whether the object can be reasonably hidden, and it is usually nontrivial for users to manually select the right hiding region. Our hiding region recommendation is built upon the following two observations: (1) Human perception system is more sensitive to image contrast. (2) A busy background often makes it easier for us to misjudge objects in an image. Formally, we formulate the following minimization to obtain the recommended hiding region:

$$\min_{M_B} \left(H(I_F \odot M_F, I_B \odot M_B) - \gamma E(I_B \odot M_B) \right), \quad (1)$$

where M_B denotes a mask that indicates the hiding region. $\odot$ denotes the pixel-wise multiplication. The first term $H(I_F \odot M_F, I_B \odot M_B)$ measures the HOG (Dalal and Triggs 2005) difference between the foreground object and the covered background region. The second term $E(I_B \odot M_B)$ computes the information entropy of the hiding region. γ is a weighting parameter set as 1 by default. Intuitively, the first term enforces the selected hiding region to have similar distribution of gradient orientation as the foreground object, so as to lower viewers' sensitivity to contrasts inside the object. On the other hand, the second term encourages selecting hiding region with high entropy to provide a busy background that is distracting and more suitable for camouflaging. Note, our method may also work well for other hiding regions that Eq. 1 is not satisfied (see the supplementary material for examples).

The problem in Eq. 1 is non-convex and difficult to solve. Hence, we employed a simple brute force search, which is fast enough because a stride length of 8 was used in both horizontal and vertical directions, instead of sliding the mask pixel by pixel. The time cost is affected by the size of the background image and the hidden object. Specifically, for a 700×435 background image and a foreground object with 34933 pixels, it takes about 1 second to find the solution.

Camouflage Image Generation

Background. We first summarize the neural style (NS) algorithm (Gatys, Ecker, and Bethge 2016), since it inspires the proposal of our camouflage image generation. This algorithm transfers the style of a reference image I_S onto an input image I to produce a target image I_T by minimizing the following objective function within a neural network:

$$\mathcal{L} = \mathcal{L}_{style} + \lambda \mathcal{L}_{content}, \quad (2)$$

where $\mathcal{L}_{style}$ and $\mathcal{L}_{content}$ are the style loss and content loss, which are separately defined as

$$\mathcal{L}_{style} = \sum_{\ell=1}^{L} \frac{\beta_\ell}{2N_\ell^2} \sum_{i=1}^{N_\ell} \sum_{j=1}^{N_\ell} \left(G_{ij}^\ell(I_T) - G_{ij}^\ell(I_S) \right)^2$$

$$\mathcal{L}_{content} = \sum_{\ell=1}^{L} \frac{\alpha_\ell}{2N_\ell D_\ell} \sum_{i=1}^{N_\ell} \sum_{j=1}^{D_\ell} \left(F_{ij}^\ell(I_T) - F_{ij}^\ell(I) \right)^2,$$

$$(3)$$

where L denotes the total number of convolutional layers in the network and ℓ indicates the ℓ-th layer. N_ℓ refers to the number of filters, and D_ℓ is the size of vectorized feature map generated by each filter. $F^\ell \in \mathcal{R}^{N_\ell \times D_\ell}$ is a feature matrix that stores the filter responses, and F_{ij}^ℓ is the activation of the i-th filter at position j. $G^\ell \in \mathcal{R}^{N_\ell \times N_\ell}$ is a Gram matrix that describes the feature correlations, where $G_{ij}^\ell = \sum_k F_{ik}^\ell F_{jk}^\ell$ is the inner product between feature maps. α_ℓ and β_ℓ are weights controlling the influence of each layer. λ is a weight that balances the two losses.

Our Approach. Inspired by the NS algorithm, we formulate a loss function to generate the desired camouflage image I_R for the given background image I_B (with mask M_B) and the foreground image I_F (with mask M_F). Formally, the loss function is defined as

$$\mathcal{L} = \mathcal{L}_{style} + \lambda_{cam} \mathcal{L}_{cam} + \lambda_{reg} \mathcal{L}_{reg} + \lambda_{tv} \mathcal{L}_{tv}, \quad (4)$$

where $\mathcal{L}_{style}$, $\mathcal{L}_{cam}$, $\mathcal{L}_{reg}$ and $\mathcal{L}_{tv}$ are the loss components. $\mathcal{L}_{style}$ is the style loss in Eq. 2. λ_{cam}, λ_{reg} and λ_{tv} are weights for the corresponding losses. Below we describe $\mathcal{L}_{cam}$, $\mathcal{L}_{reg}$ and $\mathcal{L}_{tv}$ in detail. Note, unless otherwise specified, we slightly abuse the notations I_F as well as I_B and I_R in the following illustration to indicate the image regions corresponding to the masks M_F and M_B, respectively.

Attention-aware Camouflage Loss. According to the feature integration theory, the key to camouflaging an object is to remove features that allow fast perception (feature search), and also retain some subtle features for our perception system to reveal the hidden object in terms of conjunction search. Hence, we develop an attention-aware camouflage loss defined as

$$\mathcal{L}_{cam} = \sum_{\ell=1}^{L} \mathcal{L}_{leave}^\ell + \mu \sum_{\ell=1}^{L} \mathcal{L}_{remove}^\ell, \quad (5)$$

where $\mathcal{L}_{leave}^\ell$ and $\mathcal{L}_{remove}^\ell$ are losses for leaving and removing features, which are responsible for enabling conjunction search and preventing feature search, respectively. μ is a weight, which is set as 0.5 by default. Specifically, we de-
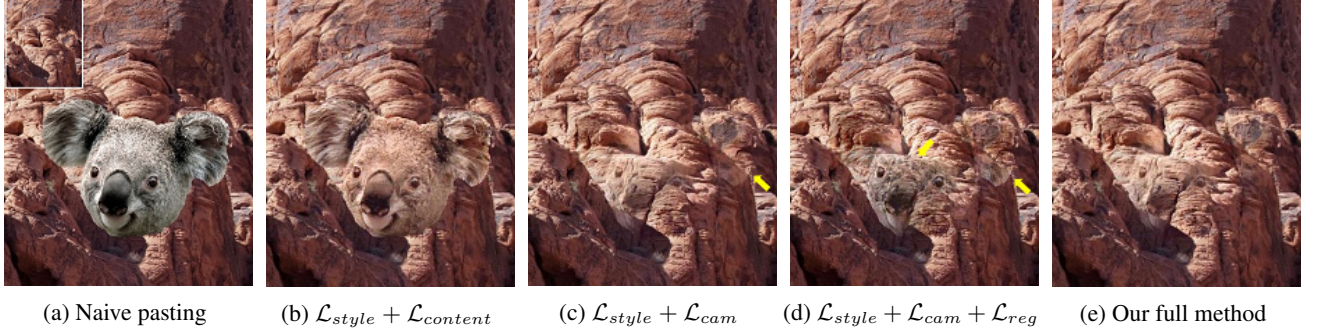
| (a) Naive pasting | (b) $\mathcal{L}_{style} + \mathcal{L}_{content}$ | (c) $\mathcal{L}_{style} + \mathcal{L}_{cam}$ | (d) $\mathcal{L}_{style} + \mathcal{L}_{cam} + \mathcal{L}_{reg}$ | (e) Our full method |

Figure 3: Ablation study that demonstrates the effectiveness of the three loss components $\mathcal{L}_{cam}$, $\mathcal{L}_{reg}$ and $\mathcal{L}_{tv}$ in our loss function. (a) Naive object pasting and the background image (top-left). (b) Camouflage image produced by the NS algorithm (Eq. 2). (c)-(e) are our generated camouflage images with different loss combinations. Zoom in to compare results.

fine $\mathcal{L}_{leave}$ and $\mathcal{L}_{remove}$ as

$$\mathcal{L}_{leave}^{\ell} = \frac{\alpha_{\ell}^1}{2N_{\ell}D_{\ell}} \sum_{i,j} |\mathcal{X}_{\mathcal{A}}(I_R) - \mathcal{X}_{\mathcal{A}}(I_F)|$$

$$\mathcal{L}_{remove}^{\ell} = \frac{\alpha_{\ell}^2}{2N_{\ell}D_{\ell}} \sum_{i,j} \left(\mathcal{A}' \odot (F_{ij}^{\ell}(I_R) - F_{ij}^{\ell}(I_B))\right)^2, \quad (6)$$

where $\mathcal{X}(\cdot)$ computes the normalized cosine distance between feature vectors of an input image, which provides an effective descriptor of the image structures (Kolkin, Salavon, and Shakhnarovich 2019). We adopt this descriptor instead of the content representation utilized in Eq. 2, since we found that image structures are more appropriate features to leave for conjunction search. $\mathcal{A}$ is a normalized attention map of the foreground object in I_F, which indicates the importance of different areas in making the object recognizable. In general, large attention value means high importance. Specifically, $\mathcal{X}_{\mathcal{A}}(I_R)$ is defined as

$$\mathcal{X}_{\mathcal{A}}(I_R) = \frac{\mathcal{C}(\mathcal{A} \odot F_{ij}^{\ell}(I_R))}{\sum_i \mathcal{C}(\mathcal{A} \odot F_{ij}^{\ell}(I_R))}, \quad (7)$$

where $\mathcal{C}(\mathcal{A} \odot F_{ij}^{\ell}(I_R))$ denotes the pairwise cosine distance matrix of all feature vectors in $\mathcal{A} \odot F_{ij}^{\ell}(I_R)$. $\mathcal{X}_{\mathcal{A}}(I_F)$ is defined similarly. Our intention behind the loss $\mathcal{L}_{leave}^{\ell}$ is to only preserve the most foundamental (indicated by the attention map $\mathcal{A}$) structures of the foreground object in the output camouflage image I_R to allow conjunction search of the hidden object. $\mathcal{A}' = 1 - \mathcal{A}$ is used to guide filling the less important foreground object areas with the content of the background image via the loss $\mathcal{L}_{remove}^{\ell}$, which helps prevent our feature search. α_{ℓ}^1 and α_{ℓ}^2 are weighting parameters of each convolutional layer. We describe how we compute the attention map $\mathcal{A}$ in the implementation details.

Naturalness Regularization. Though the style loss is employed to transfer the style of the background to the foreground object, we observed that camouflage images generated based on the style loss and camouflage loss may be visually unnatural due to the remaining appearance discontinuities between the hidden object and the surrounding background (see Fig. 3(c)). There are two reasons for this prob-

lem. First, as analyzed in (Luan et al. 2018), the style loss would be less effective when applied locally. Second, the camouflage loss may incur discontinuities between the foreground features and the newly filled background features. To address this problem, we design a naturalness regularization.

Our regularization is inspired by (Chen et al. 2012), which achieves manifold-preserved edit propagation by enforcing each pixel to be the same linear combination of its neighbors in the result. Specifically, we aim to remove the aforementioned appearance discontinuities by preserving the manifold structure of the covered background region within the hidden object. To this end, we first compute a set of weights that can best reconstruct the feature at each covered background pixel from its spatially neighboring pixels in a $K \times K$ window centered at it using the LLE method (Roweis and Saul 2000). Note, the concatenation of the RGB color channels and the spatial coordinates is used as the feature at each pixel (each dimension is normalized to [0,1]). By this means, for a foreground object with N pixels, we construct a $N \times N$ sparse weight matrix W that has K^2 nonzero entries in each column. Based on the weight matrix W, we define the following regularization to maintain the manifold structure formed by pixels in the covered background in the output camouflage image I_R:

$$\mathcal{L}_{reg} = \sum_{c \in \{r,g,b\}} \mathcal{V}_c(I_R)^T (E - W)^T (E - W) \mathcal{V}_c(I_R), \quad (8)$$

where $\mathcal{V}_c(I_R)$ is the vector representation of the color channel c of the output image I_R. E is the identity matrix. To maintain reliable manifold, we empirically set $K = 7$ in our experiments. It is worth noting that we also tried the matting Laplacian based regularization introduced in (Luan et al. 2017). However, we found that it is less effective, and may incur artifacts such as color distortion and illumination discontinuities within the hidden object (see the supplementary material for details).

Total Variational Loss. To ensure that the output camouflage image has smooth color and illumination transitions within the hidden object, we employ a total variational (TV) loss (Johnson, Alahi, and Fei-Fei 2016) defined as

$$\mathcal{L}_{tv} = \sum_p \sum_{c \in \{r,g,b\}} (\partial_x I_R)_{p,c}^2 + (\partial_y I_R))_{p,c}^2, \quad (9)$$
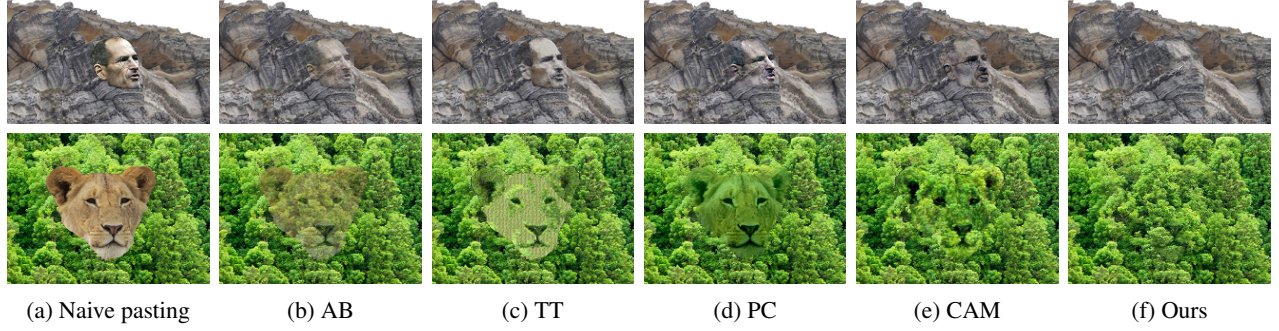
| (a) Naive pasting | (b) AB | (c) TT | (d) PC | (e) CAM | (f) Ours |

Figure 4: Comparison of our method against conventional computational camouflage methods. Zoom in to compare results.

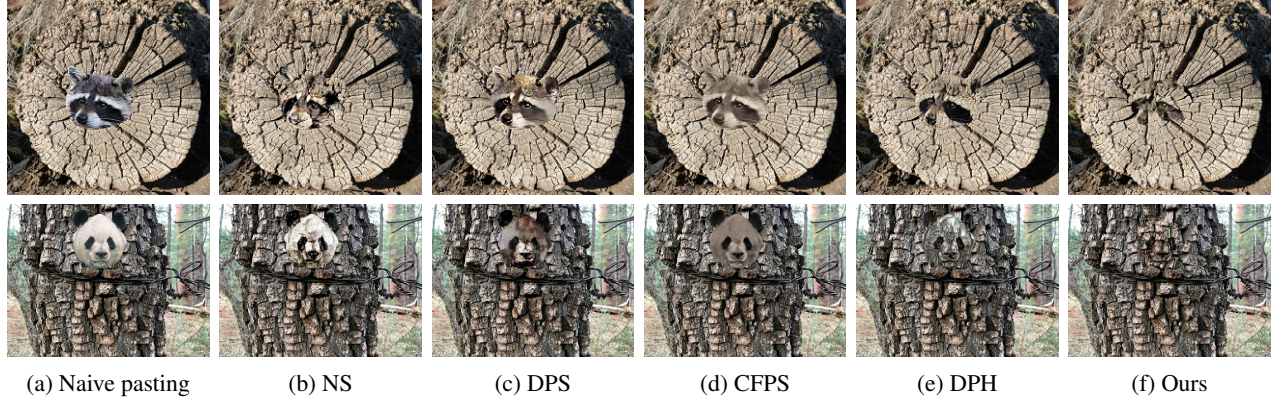| (a) Naive pasting | (b) NS | (c) DPS | (d) CFPS | (e) DPH | (f) Ours |

Figure 5: Comparison of our method against recent neural style transfer algorithms. Zoom in to compare results.

where p indexes pixels. ∂_x and ∂_y are the partial derivatives in the horizontal and vertical directions.

Implementation Details

Similar to (Gatys, Ecker, and Bethge 2016), our results are generated based on the pre-trained VGG-19 (Simonyan and Zisserman 2014). *conv4_1* is used in the camouflage loss, while *conv1_1*, *conv2_1*, *conv3_1* and *conv4_1* are chosen for the style loss. $\alpha_\ell^1 = 1$, $\alpha_\ell^2 = 1$ and $\beta_\ell = 1.5$ are set for selected convolutional layers in the respective losses, and are set as zeros for other unselected layers. The parameters $\lambda_{cam} = 10^{-6}$, $\lambda_{reg} = 10^{-9}$ and $\lambda_{tv} = 10^{-3}$ are used to produce all our results, which works well for most cases.

Attention. Here we introduce how we estimate the static attention map utilized in the camouflage loss. Our main idea is to first predict coarse attention maps for the foreground object in I_F based on (Hou and Zhang 2007) by leveraging feature representations characterized by the VGG-19, and then combine them to produce the final fine-scale attention map. Specifically, for a feature map $\mathcal{F}$ of the foreground object, we compute an attention map by

$$A_\mathcal{F} = \sum_k \Phi^{-1} \left(\log \Phi(\mathcal{F}_k) - \mathcal{G}(\log \Phi(\mathcal{F}_k)) \right), \quad (10)$$

where k indexes the feature channels. Φ and Φ^{-1} denote the Fourier Transform and Inverse Fourier Transform. $\mathcal{G}(\cdot)$ denotes the Gaussian smoothing. Based on Eq. 10, we propose

to estimate a fine-scale attention map for the foreground object by fusing coarse attention maps predicted from different convolutional layers, which is expressed as

$$\mathcal{A} = \frac{1}{5} \left(\frac{1}{4} \sum_{i=1}^{4} A_{\mathcal{F}[conv3\_i]}^{\downarrow} + \sum_{i=1}^{4} A_{\mathcal{F}[conv4\_i]} \right), \quad (11)$$

where $\mathcal{F}[\cdot]$ refers to the feature map of a convolutional layer. $\downarrow$ denotes a downsampling operation, which allows us to sum attention maps with different spatial resolutions. We do not employ shallow convolutional layers (e.g. *conv1_1* and *conv2_1*), since they usually fail to provide reliable feature representation. The 4-th level convolutional layers play more important role, since we found that they typically characterize the desired discriminative features. The average attention map of the 3-th level convolutional layers is also added to avoid missing important middle-level features.

Boundary Concealment. To conceal the boundary of the hidden object, we slightly dilate the foreground object mask and include a small portion of surrounding background as the actual hidden object, and then only obtain the resulting part indicated by the originally non-dilated object mask to generate the camouflage image. In this way, we are able to conceal the boundary, since the total variational loss together with the camouflage loss explicitly enforce smooth transitions between the hidden object and the included background (see the supplementary material for validation).
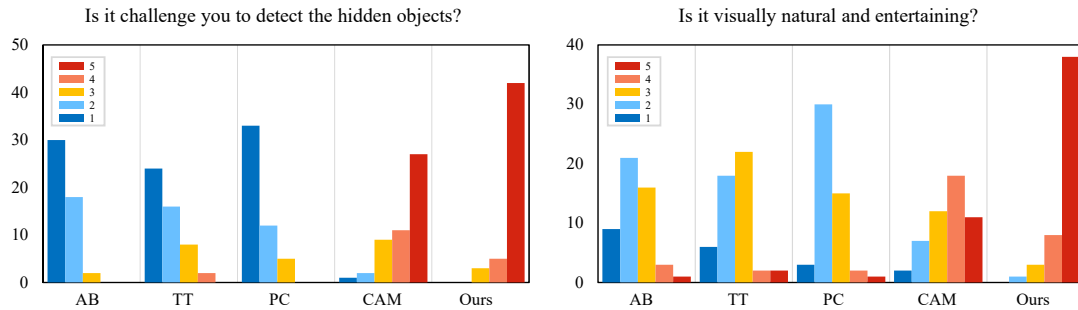
Figure 6: Rating distributions for our method and the compared conventional computational camouflage methods on the two questions in the user study. The ordinate axis shows the rating frequency.
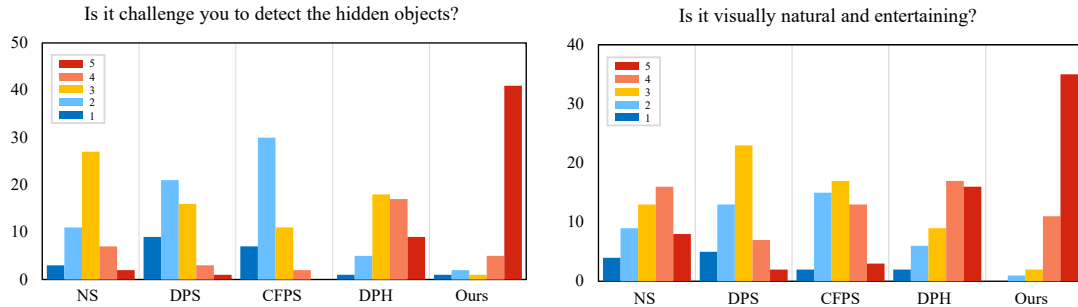

Figure 7: Rating distributions for our method and the compared neural style transfer methods on the questions in the user study.

Our algorithm was implemented in PyTorch (Paszke et al. 2017). All our experiments were conducted on a PC with an NVIDIA 1080Ti GPU. We employ the L-BFGS solver (Liu and Nocedal 1989) for image reconstruction. It takes about 2-4 minutes to generate a 700×700 camouflage image. Our code will be made publicly available at http://zhangqing-home.net/.

Experiments

In this section, we perform various experiments to validate the effectiveness of the proposed approach. We first compare our approach against existing methods. Then we conduct ablation studies to evaluate the effectiveness of the loss components and the attention in our algorithm. Finally, we present more analysis of the proposed approach. Please also see the supplementary material for more results and comparisons.

Comparison with Existing Methods

Our comparison to existing methods is twofold. First, we compare our method with various related conventional computational camouflage techniques, including alpha blending (AB), Poisson cloning (Pérez, Gangnet, and Blake 2003) (PC), texture transfer (TT) (Efros and Freeman 2001), and CAM (Chu et al. 2010). Second, we compare our method with state-of-the-art neural style transfer algorithms, including NS (Gatys, Ecker, and Bethge 2016), DPS (Luan et al. 2017), CFPS (Li et al. 2018), and DPH (Luan et al. 2018). For fair comparison, we produce the results of our competitors using their publicly-available implementations and finetune their parameters to select best visual results. We im-

plemented the algorithm of CAM (Chu et al. 2010) by ourselves, since there is no available implementation.

Visual Comparison. Fig. 4 compares our method with conventional computational camouflage methods. As shown, alpha blending, Poisson cloning and texture transfer fail to camouflage the portrait and the lion. CAM (Chu et al. 2010) successfully hides the two objects. However, the color and texture on the hidden objects are less consistent to the background, making their results visually unnatural. In contrast, our results look more satisfying, since our method better camouflages the objects and avoids appearance discontinuities. In Fig. 5, we further compare our method with recent neural style transfer methods. We can see that the four compared methods fail to hide the foreground objects though they decorate them with the style of the background. Moreover, there also exists obvious color, texture, illumination discontinuities in their results. Our method achieves much better results than the others on the two cases, demonstrating its effectiveness and superiority.

User Study. Inspired by (Chu et al. 2010; Wang et al. 2019), we also conducted a user study to evaluate our method since judging camouflage images is a highly subjective process. To this end, we first collected 28 background images that cover a broad range of scenes, subjects, and lighting conditions, and another 17 foreground objects. Then, three participants were invited to selected one or more foreground objects and their corresponding hiding regions for each background image. Next, we camouflaged the selected foreground objects into each background image using our method and other compared methods, and invited other 50

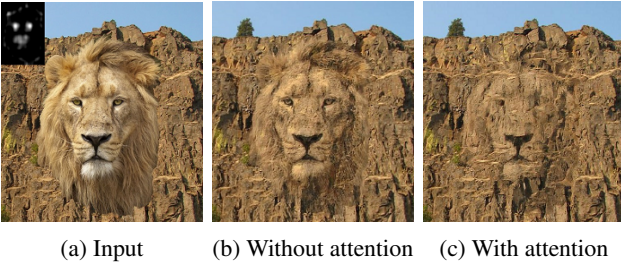(a) Input (b) Without attention (c) With attention

Figure 8: Effect of the attention. (a) Naive object pasting and the attention map for the foreground object (top-left). (b) and (c) are results without and with attention.

(a) Naive pasting (b) $\lambda_{cam} = 1e - 5$ (c) $\lambda_{cam} = 1e - 6$

Figure 9: Effect of varying λ_{cam} on the recognition difficulty of the camouflage images. Zoom in to see details.

participants via Amazon Mechanical Turk to rate each group of results, which are presented in a random order to avoid subjective bias. For each result, the participants were asked to give a rating with a scale from 1 (worst) to 5 (best) on two questions: "(1) Is it challenge you to detect the hidden objects?" and "(2) Is it visually natural and entertaining?". Figs. 6 and 7 summarize the results, where each subfigure shows five rating distributions of the evaluated methods on a particular question. The rating distributions across methods show that our results are more preferred by human subjects, where our method receives more "red" and far less "blue" ratings compared to the others.

Ablation Analysis

Effect of Loss Components. We perform ablation experiments to evaluate the effectiveness of the loss components in our loss function. Comparing Fig. 3(b) and (c), we observe better camouflage effect of the koala by using the proposed camouflage loss instead of the content loss in Eq. 2. In contrast, a naive application of the NS algorithm (i.e. $\mathcal{L}_{style} + \mathcal{L}_{content}$) results in obvious-to-recognize hidden object that most fine details (even the fur and hair) are well-preserved. By further incorporating the naturalness regularization, we successfully remove the disturbing black spots on face of the koala and produce a visually more natural camouflage image with background consistent hidden object in Fig. 3(d). As can be observed by comparing Fig. 3(d) and (e), we obtain better result with smooth boundary transitions by incorporating the total variational loss.

Effect of the Attention. Fig. 8 demonstrates the effectiveness of the attention. As shown, the attention map shows that the eyes and nose are the parts making the lion discriminative and recognizable. Hence, the result with attention in Fig. 8(c) only preserves these essential information and masks out other less important parts, while the result without attention in Fig. 8(b) retains most details (even the fur and hair), and thus makes the lion easy to recognize.

More Analysis

Robustness to Varying Illumination. Different from previous methods that are sensitive to varying illumination, our method is relatively robust to the illumination. We can see that the background image (top-left) in Fig. 3(a) has complex illumination on the hiding region, while our method
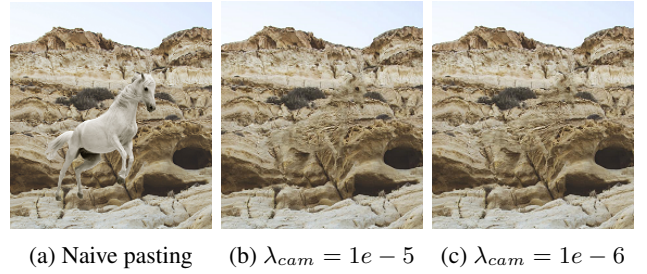
also produces satisfactory camouflage image. This advantage benefits from the designed naturalness regularization which enforces the hidden object to preserve the manifold structure of the covered background region.

Effect of Varying Parameters. Fig. 9 evaluates how λ_{cam} affects the recognition difficulty of the generated camouflage images. As shown, large λ_{cam} makes the horse easier to recognize, since the essential parts indicated by the attention, e.g. the eye, nose, and the front legs, are more standout in this situation. In contrast, small λ_{cam} yields result with high recognition difficulty. This experiment shows that it is easy for users to create camouflage images with our method at controllable levels of difficulty by adjusting λ_{cam}.

Limitations. While our method produces satisfactory camouflage images for most of our testing images, it still has limitations. First, it cannot camouflage objects into smooth background due to lack of exploitable distracting image contents. Second, when the hidden objects are placed across regions with significantly different textures (e.g. across mountain and sky), it may fail.

Conclusion

We have presented an approach for creating camouflage images. Unlike previous methods, we propose to generate camouflage images by minimizing a novel perception-driven loss function within a neural network, which allows us to take full advantage of the high-level image representations learned by the network to achieve better camouflage effect. Extensive experiments have been performed to validate the effectiveness of the propose approach. We believe that our approach can be an effective tool for amateurs to create camouflage images, and has great potential in advancing studies in cognitive psychology on how humans perceive as well as in other applications such as gaming and entertainment.
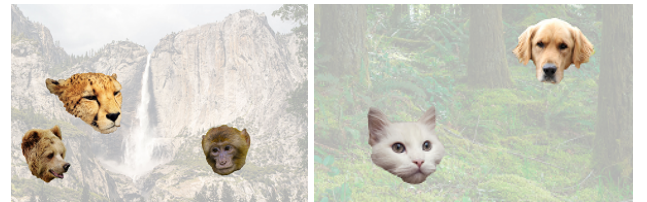

Figure 10: Answers keys for camouflage images in Fig. 1.

References

Chen, X.; Zou, D.; Zhao, Q.; and Tan, P. 2012. Manifold preserving edit propagation. *ACM Transactions on Graphics (TOG)* 31(6):132.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *CVPR*.

Chu, H.-K.; Hsu, W.-H.; Mitra, N. J.; Cohen-Or, D.; Wong, T.-T.; and Lee, T.-Y. 2010. Camouflage images. *ACM Transactions on Graphics (TOG)* 29(4):51–1.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

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

Gatys, L. A.; Ecker, A. S.; Bethge, M.; Hertzmann, A.; and Shechtman, E. 2017. Controlling perceptual factors in neural style transfer. In *CVPR*.

Gatys, L.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. In *NIPS*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *CVPR*.

Gu, S.; Chen, C.; Liao, J.; and Yuan, L. 2018. Arbitrary style transfer with deep feature reshuffle. In *CVPR*.

Heeger, D. J., and Bergen, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proc. of ACM SIGGRAPH*.

Hou, X., and Zhang, L. 2007. Saliency detection: A spectral residual approach. In *CVPR*.

Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.

Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; and Song, M. 2019. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.

Kolkin, N.; Salavon, J.; and Shakhnarovich, G. 2019. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*.

Li, C., and Wand, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017a. Demystifying neural style transfer. In *IJCAI*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017b. Diversified texture synthesis with feed-forward networks. In *CVPR*.

Li, Y.; Liu, M.-Y.; Li, X.; Yang, M.-H.; and Kautz, J. 2018. A closed-form solution to photorealistic image stylization. In *ECCV*, 453–468.

Liao, J.; Yao, Y.; Yuan, L.; Hua, G.; and Kang, S. B. 2017. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)* 36(4):120.

Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming* 45(1-3):503–528.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *CVPR*.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2018. Deep painterly harmonization. *Computer Graphics Forum* 37(4):95–106.

Owens, A.; Barnes, C.; Flint, A.; Singh, H.; and Freeman, W. 2014. Camouflaging an object from many viewpoints. In *CVPR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Pérez, P.; Gangnet, M.; and Blake, A. 2003. Poisson image editing. *ACM Transactions on graphics (TOG)* 22(3):313–318.

Reynolds, C. 2011. Interactive evolution of camouflage. *Artificial life* 17(2):123–136.

Roweis, S. T., and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290(5500):2323–2326.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tankus, A., and Yeshurun, Y. 2001. Convexity-based visual camouflage breaking. *Computer Vision and Image Understanding* 82(3):208–237.

Treisman, A. M., 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. *The Quarterly Journal of Experimental Psychology Section A* 40(2):201–237.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. S. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*.

Wang, R.; Zhang, Q.; Fu, C.-W.; Shen, X.; Zheng, W.-S.; and Jia, J. 2019. Underexposed photo enhancement using deep illumination estimation. In *CVPR*.

Wolfe, J. M. 1994. Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review* 1(2):202–238.