A Statistical Prior for Photo Forensics: Object Removal

Wei Fan and Hany Farid
Dartmouth College
Hanover, NH 03755

***Abstract*—If we consider photo forensics within a Bayesian framework, then the probability that an image has been manipulated given the results of a forensic test can be expressed as a product of a likelihood term (the probability of a forensic test detecting manipulation given that an image was manipulated) and a prior term (the probability that an image was manipulated). Despite the success of many forensic techniques, the incorporation of a statistical prior has not been previously considered. We describe a framework for incorporating statistical priors into any forensic analysis and specifically address the problem of quantifying the probability that a portion of an image is the result of content-aware fill, cloning, or some other form of information removal. We posit that the incorporation of such a prior will improve the overall accuracy of a broad range of forensic techniques.**

***Index Terms*—Image forensics, image manipulation, perceptual similarity**

I. Introduction

To date, many photo forensic techniques have been proposed to detect various forms of photo manipulation [1]. Because most of these techniques require manual intervention, the human analyst remains a critical part of many forensic analyses. In Fig. 1 (left), for example, an analyst might reasonably assume that the addition of a child would have been technically difficult to fake because of the complex interaction between the two people in the scene. In Fig. 1 (right), however, the addition of the child would have been relatively easy because the child is isolated from her surroundings. This type of prior information should be incorporated into any forensic analysis.

Broadly speaking, we can characterize photo manipulation into one of three categories: (1) removal; (2) addition; or (3) modification of one or more people/objects. We seek to quantify the prior likelihood that part of an image has been manipulated in one of these ways.

There are myriad of forensic techniques for detecting photo manipulation. Formally, we would like to ask what is the probability that an image was manipulated given the result of a forensic test, $P(m|f)$. Within a Bayesian framework, we can express this probability as $P(m|f) \propto P(f|m) \cdot P(m)$, where $P(f|m)$ is the (typically) known true positive rate of the forensic test and $P(m)$ is the prior that the image was manipulated. To date, we have no systematic and quantitative way of quantifying the prior $P(m)$. We describe an approach to quantify this prior for object removal – we leave the task of quantifying a prior for object addition and modification to future work.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Fig. 1. It would have been difficult to digitally insert the child into the image on the left due to the complex physical interaction between mother and child. In contrast, it would have been relatively easy to insert the child into the image on the right because the child is isolated from her surroundings. [photo credits: (left) flickr user Paul Wan: flic.kr/p/5nsFFy; (right) flickr user Ian D. Keating: flic.kr/p/eK8aDg]

We draw inspiration for quantifying object removal from Photoshop's content-aware fill feature [2]. Content-aware fill removes an object by copying and synthesizing image content to create a seamless transition with nearby content. This manipulation replaces an object with content that is perceptually similar (but not necessarily identical) to other content in the image. We reason, therefore, that the perceptual similarity of one region to the rest of the image will provide a measure of likelihood that this region is the result of content-aware fill or another similar manipulation. We first describe how to quantify this perceptual similarity and then describe how to convert this measure to a probability of object removal.

II. Perceptual Similarity

Standard techniques for measuring perceptual similarity rely heavily on a pixel-to-pixel comparison [3]. While this is appropriate in many situations, it is not appropriate to the type of perceptual similarity that we seek to quantify. To illustrate this, consider the three textures in Fig. 2. The left texture is perceptually similar to the middle texture and the right texture is perceptually dissimilar to the middle texture (in terms of both color and structure). In contrast to this obvious perceptual similarity, SSIM [4] and root mean square (RMS) each rate the middle texture as more similar to the right than the left texture. Our similarity measure correctly rates the middle and

Fig. 2. Shown are two perceptually similar textures (middle and left) and a perceptually dissimilar texture (right). According to both SSIM and RMS, the middle texture is more similar to the right than the left texture (larger SSIM values correspond to higher similarity). In contrast, our similarity measure, s, correctly identifies the middle and left textures as perceptually similar and the middle and right textures as highly dissimilar (larger values of s correspond to higher similarity).

left textures as highly similar and the middle and right textures as highly dissimilar.

Despite some shortcomings on the part of SSIM in capturing perceptually similarity, our perceptual similarity measure is inspired by the building blocks of SSIM that measures quantities such as luminance, contrast, and structure. In the next few sections, we describe the four building blocks of our measure and then describe how to combine these components into a single measure of perceptual similarity.

A. Notation

We will denote a 3-channel RGB image as $I_c(x,y) = I_c(\vec{x})$ where $c \in \{r,g,b\}$, and $\vec{x} = (x,y)$ corresponds to the horizontal and vertical pixel location. When converted from RGB to HSV space, an image will be denoted as $I_c(\vec{x})$ where $c \in \{h,s,v\}$ corresponds to each of the HSV image channels.

Consider now a $n \times n$ neighborhood centered at pixel location $\vec{x}_i = (x_i, y_i)$. We will denote the n^2 pixels in this neighborhood sorted in descending order as $\hat{I}_c(\vec{x}_i^k)$, where the subscript c denotes the image channel, the subscript i denotes the pixel location in the image, and the superscript k denotes the k^{th} spatial value in sorted order.

B. Color: hue

We consider the similarity of two image regions in terms of the hue channel, $I_h(\cdot)$. The hue takes on a value in the range $[0,1]$ but is circular in nature so a hue value of 0 is the same as a hue value of 1. The difference between two hue values h_1 and h_2 is therefore measured in angular units as $|2\pi h_1 - 2\pi h_2| \ \% \ \pi$, where $\%$ is the modulus operator.

The difference in hue between two $n \times n$ pixel neighborhoods centered at $\vec{x}_i$ and $\vec{x}_j$ is defined to be:

$$s_h(i,j) \quad = \quad \frac{1}{n^2} \sum_{k=1}^{n^2} \frac{\cos(\min(\theta_0, \delta_k)) - \cos(\theta_0)}{1 - \cos(\theta_0)}, \qquad (1)$$

where $\delta_k = \left| 2\pi \hat{I}_h(\vec{x}_i^k) - 2\pi \hat{I}_h(\vec{x}_j^k) \right| \ \% \ \pi$. The angular difference in hue is subjected to two non-linearities: the difference is subjected to a point-wise cosine non-linearity and differences in angular hue larger than a specified threshold of θ_0 are pegged to a value of θ_0. Combined with the additive and divisive normalization, the effect of this second non-linearity is that the overall similarity $s_h(\cdot)$ is equal to 0 for any angular hue difference greater than θ_0.

Note that this similarity is computed over the sorted hue values $\hat{I}_h(\cdot)$ and not the original hue values $I_h(\cdot)$. By doing so, we avoid a pixel-by-pixel comparison that often fails to capture the desired measure of perceptual similarity.

The hue similarity term, $s_h(\cdot)$, is a scalar in the range $[0,1]$ where values close to 0 correspond to a dissimilar hue and values close to 1 correspond to a similar hue.

C. Color: saturation

In this section we will consider the similarity of two image regions in terms of the saturation channel, $I_s(\cdot)$. The difference in saturation between two $n \times n$ pixel neighborhoods centered at $\vec{x}_i$ and $\vec{x}_j$ is defined to be:

$$d_s(i,j) \quad = \quad \frac{1}{n^2} \sum_{k=1}^{n^2} \left(\hat{I}_s(\vec{x}_i^k) - \hat{I}_s(\vec{x}_j^k) \right)^2. \qquad (2)$$

This mean-squared distance is subjected to a point-wise non-linearity (an exponential) of the form:

$$s_s(i,j) \quad = \quad \exp\left(-\frac{d_s(i,j)}{2{\sigma_s}^2} \right). \qquad (3)$$

Note again that this similarity is computed over the sorted saturation values $\hat{I}_s(\cdot)$. The saturation similarity term, $s_s(\cdot)$, is a scalar in the range $[0,1]$ where values close to 0 correspond to a dissimilar saturation and values close to 1 correspond to a similar saturation.

D. Color: value

In this section we will consider the similarity of two image regions in terms of the value channel, $I_v(\cdot)$. This similarity term takes on the same form as the saturation term defined above. The difference in intensity value between two $n \times n$ pixel neighborhoods centered at $\vec{x}_i$ and $\vec{x}_j$ is defined to be:

$$d_v(i,j) \quad = \quad \frac{1}{n^2} \sum_{k=1}^{n^2} \left(\hat{I}_v(\vec{x}_i^k) - \hat{I}_v(\vec{x}_j^k) \right)^2. \qquad (4)$$

This mean-squared distance is again subjected to a point-wise non-linearity of the form:

$$s_v(i,j) \quad = \quad \exp\left(-\frac{d_v(i,j)}{2{\sigma_v}^2} \right). \qquad (5)$$

As in the case of the hue and saturation, this similarity is computed over sorted values $\hat{I}_v(\cdot)$. The similarity term, $s_v(\cdot)$, for intensity value is a scalar in the range $[0,1]$ where values close to 0 correspond to a dissimilar intensity value and values close to 1 correspond to a similar intensity value.

E. Structure

The color-based terms of hue $s_h(\cdot)$, saturation $s_s(\cdot)$, and value $s_v(\cdot)$ measure the overall similarity in color between two neighborhoods. These terms do not, however, measure the structural similarity in terms of local structure and spatial frequency. This fourth, and final, structure-based term is designed to fill this gap.

We begin by computing the first-order horizontal and vertical derivatives of the value channel $I_v(\cdot)$:

$$d_x(x,y) = I_v(x,y) \star d(x) \star p(y) \quad (6)$$

$$d_y(x,y) = I_v(x,y) \star p(x) \star d(y), \quad (7)$$

where $*$ is the convolution operator and $d(\cdot)$ and $p(\cdot)$ are the three-tap filters defined in [5]. From each of these directional derivatives, we compute four auto-correlations of a $n \times n$ pixel neighborhood. In particular, four correlations are computed along the horizontal, vertical, diagonal, and anti-diagonal directions with a delay of one pixel. This yields a total of eight auto-correlation values (four for the horizontal derivative and four for the vertical derivative) which are then packed into a single vector $\vec{c}$.

The structural similarity between two regions centered at $\vec{x}_i$ and $\vec{x}_j$ is then given by:

$$s_c(i,j) = \exp\left(-\frac{\frac{1}{8}\left\|\vec{c}_i - \vec{c}_j\right\|^2}{2{\sigma_c}^2}\right), \quad (8)$$

where $\|\cdot\|$ denotes the ℓ_2-norm.

The structural similarity term, $s_c(\cdot)$, is a scalar in the range $[0,1]$ where values close to 0 correspond to a dissimilar structure and values close to 1 correspond to a similar structure.

F. Overall Similarity

The four similarity measures (three color and one structure terms) are combined with a simple product to yield the similarity measure between two neighborhoods centered at spatial locations $\vec{x}_i$ and $\vec{x}_j$:

$$s(i,j) = s_h(i,j) \cdot s_s(i,j) \cdot s_v(i,j) \cdot s_c(i,j). \quad (9)$$

Because each of the four measures on the right-hand side are in the range $[0,1]$, the final similarity measure is also bound into this range. The rationale for multiplying these values is that if any of the four individual terms is dissimilar (close to 0) then the entire similarity measure is penalized. Or, put another way, a high similarity measure (close to 1) is only obtainable if each individual measure is close to unit-value.

III. From Perceptual Similarity to Statistical Prior

Our perceptual similarity measure is defined between two $n \times n$ neighborhoods. Here we describe how to extend this analysis to measure the perceptual similarity between an arbitrary sized image region – e.g., the output of clone detection [6]–[8] – and the rest of the image.

Define the set of all image pixel locations as $\mathcal{I}$, the set of all pixel locations in a user-specified region as $\mathcal{R}$, and the remaining pixel locations as $\mathcal{D} = \mathcal{I} - \mathcal{R}$. Our goal is to determine the perceptual similarity of the region $\mathcal{R}$ to the rest of the image $\mathcal{D}$. We accomplish this by iteratively matching $n \times n$ neighborhoods using a modified RANSAC algorithm [9].

To begin, we define $\mathcal{M}$ to be the set $\{(\vec{x}_i, \vec{x}_j)\}$ for all $\vec{x}_i \in \mathcal{R}$ where $\vec{x}_j$ is the pixel location that maximizes $s(i,j)$, Equation (9), over all $j \in \mathcal{D}$. That is, based on only a $n \times n$ neighborhood, $\vec{x}_i$ in the region of interest is maximally similar to $\vec{x}_j$ outside of the region of interest.

The iterative matching algorithm then proceeds as follows. Select a random pair of matching neighborhoods in $\mathcal{M}$ and compute the 2-D translation $\vec{t}$ between these two neighborhoods. For every pixel location in $\mathcal{R}$ compute the similarity $s(i,t)$ between the neighborhoods centered at $\vec{x}_i$ and $\vec{x}_t = \vec{x}_i + \vec{t}$ (if $\vec{x}_t$ is not in $\mathcal{D}$, then this location is ignored). If this similarity is within a specified threshold of the maximal similarity for $\vec{x}_i$ specified in $\mathcal{M}$ then add $\vec{x}_i$ to an initially empty set $\mathcal{M}^*$ and set the similarity for $\vec{x}_i$ to $S(i) = s(i,t)$. Starting with the initially randomly selected point, eliminate any pixel locations in $\mathcal{M}^*$ that is not connected to this point within an 8-pixel neighborhood connectivity. For every pixel in this region, set $C(\cdot)$ to be the cardinality of $\mathcal{M}^*$. This connectivity constraint favors matching a smaller number of spatially coherent regions as opposed to a larger number of spatially disparate regions. On each successive iteration, all pixel locations in the trimmed $\mathcal{M}^*$ are removed from $\mathcal{M}$, and these iterations are repeated until $\mathcal{M}$ is empty.

At the end of this iterative matching, each pixel location $\vec{x}_i$ in the region of interest $\mathcal{R}$ has associated with a similarity $S(i)$ and a cardinality $C(i)$ of the size of the connected region to which it belongs. This similarity and cardinality are converted to a probability as follows:

$$P(\vec{x}_i) = \min(\log_{100}(C(i)), 1) \cdot \exp\left(-\frac{\left|(S(i)-1\right|^\beta}{\alpha}\right). \quad (10)$$

The exponential non-linearity penalizes small values of $S(i)$ (recall that a similarity value of 1 corresponds to maximal similarity while a value of 0 corresponds to minimal similarity). The multiplicative term in front of the exponential penalizes small regions – this penalty is clipped at the high end so as not to disproportionately reward regions larger than 100 pixels.

Shown in top two rows of Fig. 3 are three iterations of this process. The first row shows the user-specified region of interest (green) and the matched regions (red). The second row shows the probability $P(\cdot)$, Equation (10). The pixels corresponding to the plane have low probability because there are no similar coherent regions in the image, while the surrounding sky has high probability because of its similarity to the rest of the sky. Shown in the next two rows is the same information but now for a version of the image in which the plane was removed using Photoshop's content-aware fill. The user-specified region has uniformly high probability suggesting a high likelihood that this part of the image is the result of some form of object removal. Any part of the sky would have yielded a similar result so the result of this

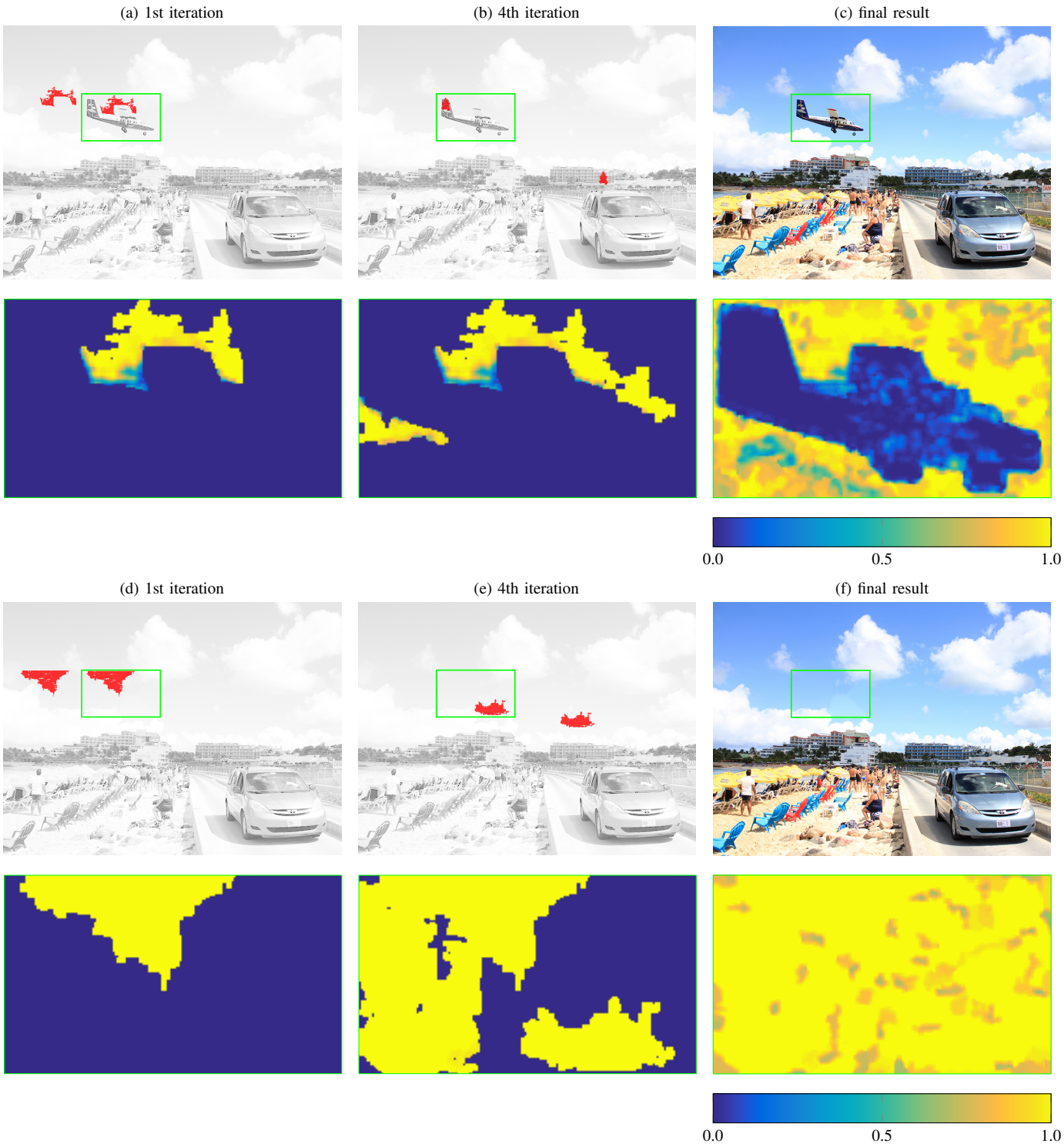

Fig. 3. Shown in panels (a)-(c) and (d)-(f) are several iterations involved in the estimation of the probability of object removal, Equation (10). The green bounding box corresponds to the user-supplied region of interest. The red regions correspond to the matched locations on a single iteration. The probability maps, shown below each image, correspond to the pixel-wise probability of object removal. Because the plane is dissimilar to the rest of the image, it results in a relatively low probability as compared to the sky region which is highly similar to other parts of the image. [photo credit (original): flickr user Zippo S, flic.kr/p/n9TjSf]

analysis does not indicate manipulation, but only a prior on the likelihood of manipulation.

IV. Validation

One of the challenges of this work is devising a systematic mechanism for validating our perceptual measure of similarity. To this end, we generated chimeric textures as shown in Fig. 5. Each texture in this figure is constructed by combining two textures T_1 and T_2 with a statistical feature vector $\vec{f}_1$ and $\vec{f}_2$ [10]. Specifically, a central region in texture T_1 is replaced with a texture synthesized from the following linear combination of statistical feature vectors: $\vec{f}_r = (1-r)\vec{f}_1 + r\vec{f}_2$, where $r \in [0, 1]$. When $r = 0$, the central texture perceptually matches the surrounding texture T_1 and when $r = 1$ the central texture perceptually matches the texture T_2.

Shown in Fig. 5 are four mixtures (with $r \in \{0.0, 0.3, 0.6, 1.0\}$) for eight pairs of source textures of varying similarity. Also shown in each figure is the morphing value r plotted as a function of probability of object removal $P(\cdot)$, Equation (10). This probability is averaged over the central region (error bars correspond to plus/minus one standard deviation). In each case, $P(\cdot)$ and r are well correlated and at a qualitative level, the rate of decay in probability as a function of texture mixture is consistent with the perceptual similarity of the central and surrounding regions.

Shown in Fig. 4 are results from real-world images with a user-specified region (green) and the resulting probability of object removal. Shown are (a) an original image, (b) an image subjected to Photoshop's content-aware fill to remove an object, and (c)-(d) two images with cloning. In each case, our perceptual measure of similarity appropriately characterizes the specified regions of interest.

A. Implementation Details

We describe the various parameters used in our implementation. The neighborhood size is $n = 15$ pixels. The parameters for the perceptual similarity measure are $\theta_0 = \pi/6$, $\sigma_s = 0.2$, $\sigma_v = 0.2$, and $\sigma_c = 0.2$ (see Equations (1), (3), (5), and (8)). The parameters for converting similarity to probability are $\alpha = 0.0625$ and $\beta = 4$ (see Equation (10)).

The computational demands for our analysis depend on a number of factors including the size of the user-specified region and the overall size of the image. For the images shown in Fig. 4, the analysis (implemented in MatLab) took between 13 minutes and 50 minutes running on a 2.7 GHz processor. Considerable effort will be needed to improve the run-time efficiency of this algorithm.

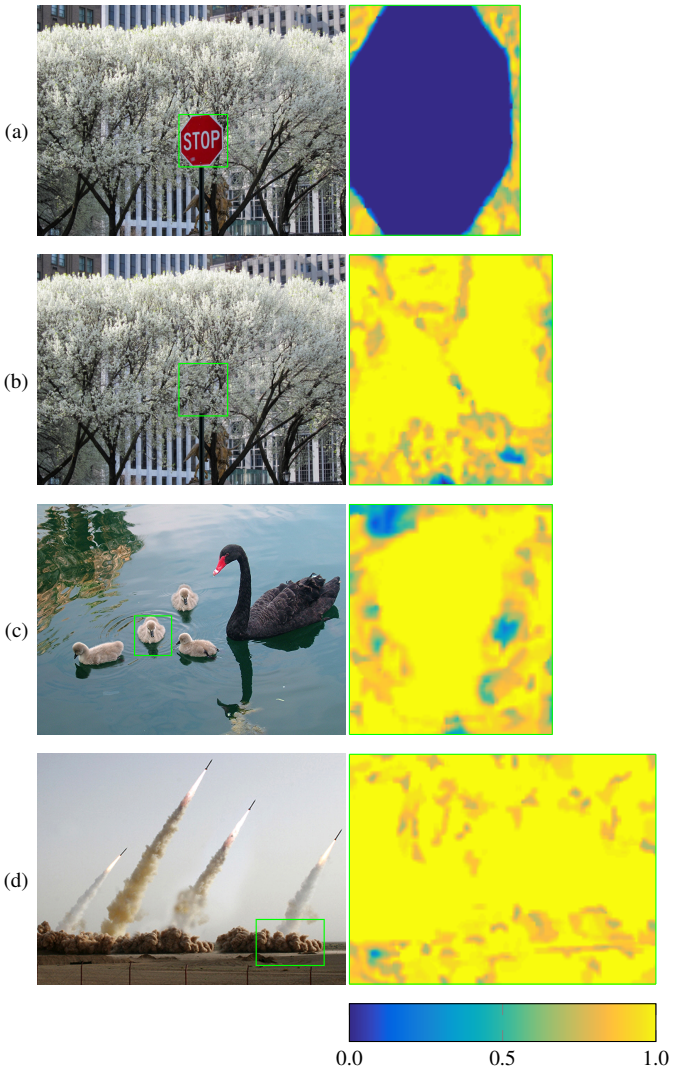

Fig. 4. Shown in each panel is an image with a user-specified region (green) and the resulting probability of object removal: (a) an original image; (b)-(d) manipulated images [photo credits (original): (a) flickr user Bosc d'Anjou, flic.kr/p/9AhSwK; (c) flickr user Pamala Wilson, flic.kr/p/fbgJK].

References

[1] H. Farid, *Photo Forensics*. MIT Press, 2016.

[2] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[3] Z. Zujovic, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2545–2558, 2013.

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[5] H. Farid and E. P. Simoncelli, "Differentiation of discrete multidimensional signals," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 496–508, 2004.

[6] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 857–867, 2010.

[7] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.

[8] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, 2015.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–71, 2000.

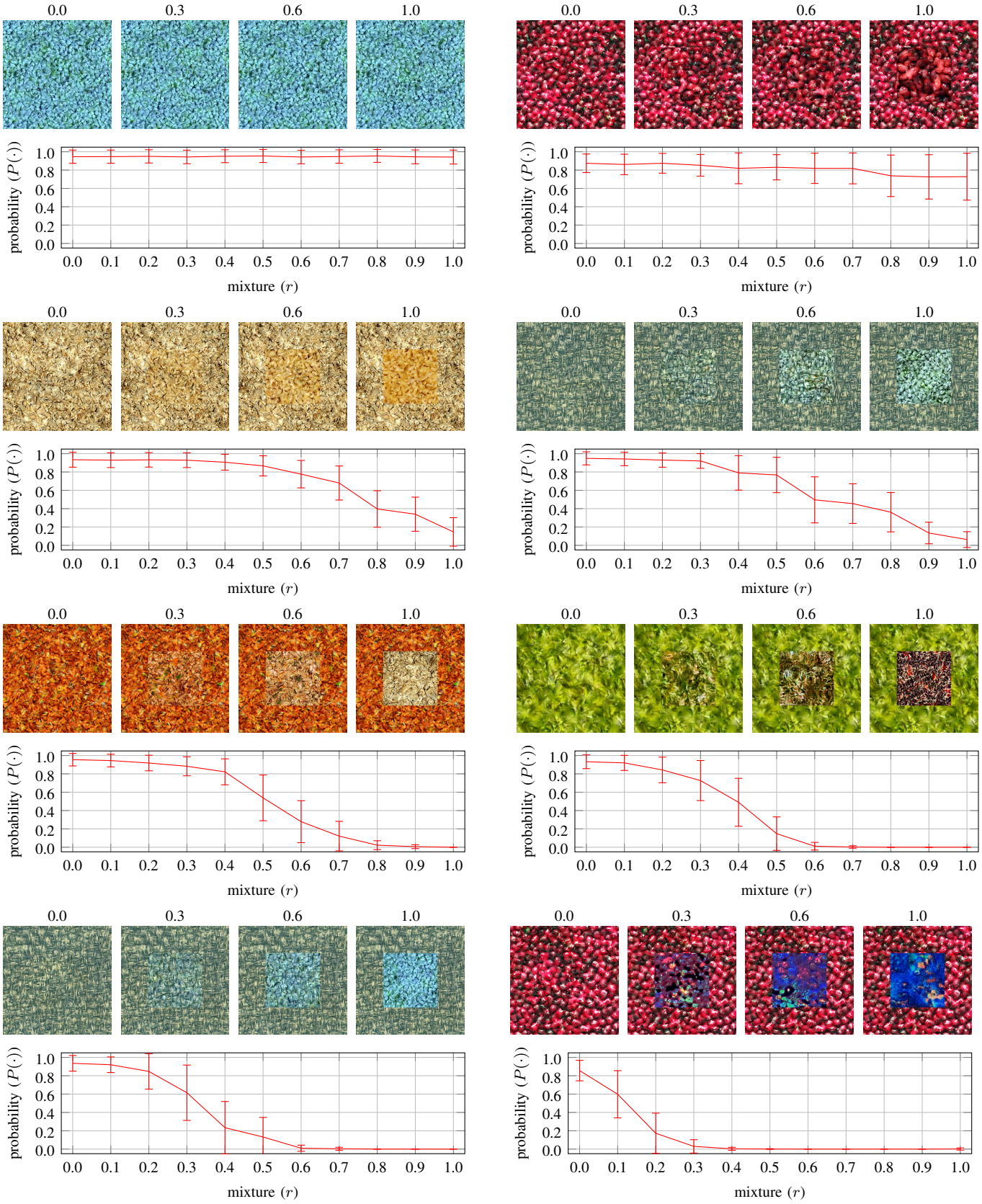

Fig. 5. Shown in each panel are (top) a series of chimeric textures in which the central region has varying degrees of perceptual similarity to the surround (specified by a mixture term r), and (bottom) the relationship between r and our measure of similarity $P(\cdot)$. Each data point corresponds to the average probability in the central region and the error bars correspond to plus/minus one standard deviation.