Detecting and Extracting the Photo Composites Using Planar Homography and Graph Cut

Wei Zhang, Xiaochun Cao, Yanling Qu, Yuexian Hou, Handong Zhao, and Chenyang Zhang

***Abstract*—With the advancement of photo and video editing tools, it has become fairly easy to tamper with photos and videos. One common way is to insert visually plausible composites into target images and videos. In this paper, we propose an automatic fake region detection method based on the planar homography constraint, and an automatic extraction method using graph cut with online feature/parameter selection. Two steps are taken in our method: 1) the targeting step, and 2) the segmentation step. First, the fake region is located roughly by enforcing the planar homography constraint. Second, the fake object is segmented via graph cut with the initialization given by the targeting step. To achieve an automatic segmentation, the optimal features and parameters for graph cut are dynamically selected via the proposed online feature/parameter selection. Performance of this method is evaluated on both semisimulated and real images. Our method works efficiently on images as long as there are regions satisfying the planar homography constraint, including image pairs captured by the approximately cocentered cameras, image pairs photographing planar or distant scenes, and a single image with duplications.**

***Index Terms*—Graph cut, online feature/parameter selection, photo composites, planar homography.**

I. Introduction

THERE is a phenomenon rising in the past decades: people are fond of tampering with photos and videos. With the popularity of the networks and multimedia, there is a growing number of tampered photos and videos flooding televisions, magazines, and networks, which hide the truth. At the same time, with powerful image and video editing tools, it is becoming easy to tamper with images and videos. Evaluating the authentication has turned out to be an important task today.

In the past few years, both active and passive methods have been developed for image forensics. Digital watermarking [1], [2] is known as a popular active technique. However, it is applicable mainly in a controlled environment since it requires cooperation from the image taker to insert a watermark at the recording time. Passive methods are the newly developed techniques and have wider usage since they require nothing from the image taker. Generally, previous passive techniques can be roughly grouped into five categories [3]: 1) pixel-based techniques; 2) format-based techniques; 3) camera-based techniques; 4) physically-based techniques; and 5) geometric-based techniques. To be detailed, various cues in different perspectives have been used as evidence in finding the fake region from images. Copy-move detection [4] and duplication detection [5] are those pixel-based examples. JPEG quantization [6], double JPEG [7], and JPEG ghost [8] are the format-based ones. Camera responses [9] and sensor noises [10], [11] are those camera-based ones. Lighting conditions [12] and shadow matte consistency [13] are examples of physically-based techniques. Estimating the principle point [14] and skew parameter [15] belongs to the geometric methods.

Methods using inconsistency to detect forgery serve a big branch of the above five categories. Many previous works have been done by the pioneering groups, e.g., Wu [16]–[18], Fridrich [10], [11], [19], and Farid [14], [15], [20]. Swaminathan *et al.* detect the forensics using inconsistency in CFA interpolation features [16], [18], and inconsistency in coefficients of the linear components [17]. Chen *et al.* [10] and Fridrich [11] detect the forensics by evaluating the inconsistency in sensor noise. Lin *et al.* [9] evaluate the images with inconsistency on camera response. Johnson and Farid [14] detect composites of people using the inconsistency of the principle points estimated from human eyes. Wang and Farid [15] detect photo/video reprojections by judging the inconsistency in the skew parameter. Johnson and Farid [20] reveal the inconsistency by rectifying regions, which requires known world geometry (polygons or circles) existing in the scene.

Our method is a new inconsistency-based method which uses the inconsistency within the planar homography, and extracts them using the graph cut. As shown in Fig. 1, the real world appears differently through different camera setups, and the inconsistencies between different images are usually un-noticeable for human eyes due to the distortions in shapes and positions. Fortunately, these distortions follow certain laws. That is, appearances and positions of rigid objects in one image are related with another image monitoring the same scene. Theoretically, our method works as long as there are objects satisfying the planar homography. There are cases when duplication appears within a single image, e.g., copying and pasting a region to a different position (Fig. 1, the last row). There are also cases when two or more images are capturing the same scene but with different setups (e.g., the visual surveillance sites, famous places

Manuscript received September 02, 2009; revised April 21, 2010; accepted May 14, 2010. Date of publication June 10, 2010; date of current version August 13, 2010. This work was supported by the National Natural Science Foundation of China (60905019, 50735003), by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, by the State Education Ministry, by the Open Projects Program of National Laboratory of Pattern Recognition, by Tianjin University 985 Research Fund, and by the State Key Laboratory of Precision Measuring Technology and Instruments Open Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Min Wu.

W. Zhang, X. Cao, Y. Qu, Y. Hou, and H. Zhao are with the Department of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: wzhang@tju.edu.cn; xcao@tju.edu.cn; yanlingqu@tju.edu.cn; yxhou@tju.edu.cn; hdzhao@tju.edu.cn).

C. Zhang is with School of Computer Software, Tianjin University, Tianjin 300072, China (e-mail: chenyangzhang@tju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2051666

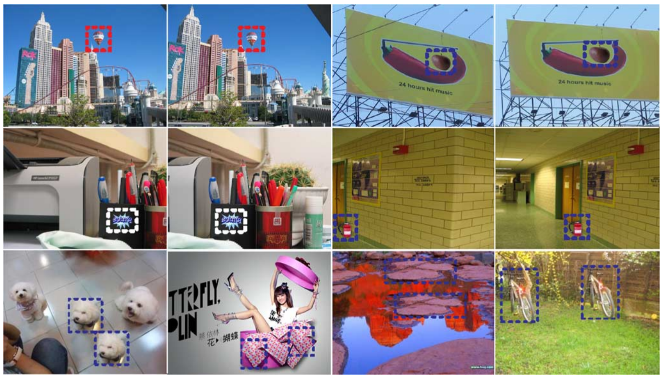


Fig. 1. It is difficult even for humans to judge the authenticity or to locate the photo composites in visually plausible images. The dashed rectangles, which indicate the fake regions, are generated with our proposed method.

photographed by various tourists across the year). In this work, we rectify/align the images using the planar homography and then extract the fake region using the graph cut.

The common weaknesses of the existing methods are the absence of automatization and segmentation. Usually, suspected regions are selected by humans, i.e., the detection must be under human supervision, and thus lacks automatization. Another weakness is the lack of segmentation for the faked object. Farid mentioned in [8] that segmentation is a necessary step for automatically and efficiently analyzing large amounts of images. For most existing techniques, a rough bounding box containing the fake object is the final output. In this work, this weakness is tackled with graph-cut-based segmentation techniques. The traditional graph cut [21] is not directly applicable when picking up fake regions from a large amount of images since it requires human interaction (specifying the source/sink values, adding hard constraints, choosing features/parameters, etc.). Automatic graph cut is challenging since the optimal features and parameters are highly picture-dependent. Peng and Veksler [22] introduce an automatic parameter training method via AdaBoost. However, this method requires a large image database and long training time, and also involves user interaction. In this work, we make use of the online feature/parameter selection, which was originally utilized in object tracking [23], to extract fake regions automatically. Partial results were presented in [24].

The main steps of our method are illustrated in Fig. 2. We use a pair of distant scenes as the example. Later in this paper, we extend the method to the single image case. In general, our method takes two steps. One is the targeting step which locates the location of the fake region roughly using the planar homography constraint. The other step is the segmentation which extracts the fake object more precisely.

In summary, our method has the following advantages compared with previous ones:

1) The planar homography constraint is introduced for the fake region detection as a geometrical method.
2) Rather than the rough location, precise boundaries of the fake object are extracted.
3) The online feature/parameter selection framework is adopted to improve the performance and automatization of the segmentation process.

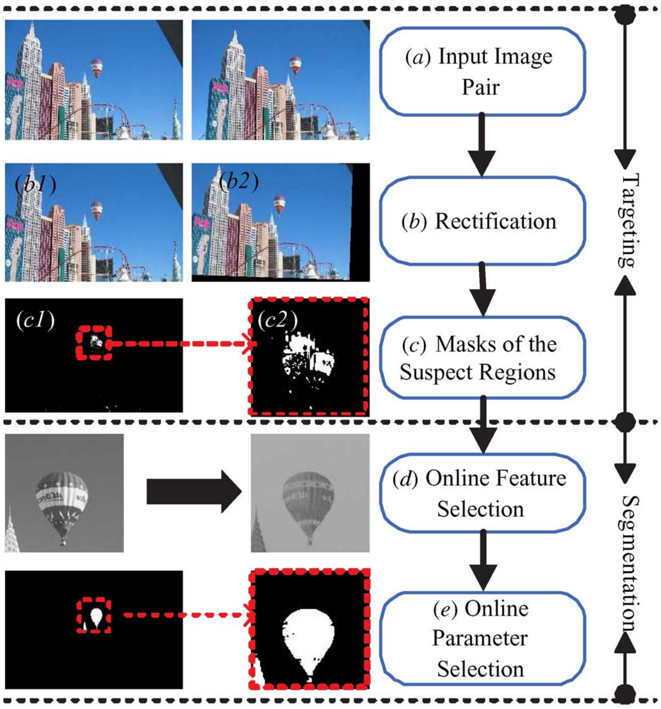

(*a*) Input Image Pair
(*b1*)
(*b2*)
(*b*) Rectification
(*c1*)
(*c2*)
(*c*) Masks of the Suspect Regions
(*d*) Online Feature Selection
(*e*) Online Parameter Selection
Targeting
Segmentation

Fig. 2. Overview of our method. Steps are shown on the right and examples are on the left.

II. Fake Region Targeting

As we know from multiple view geometry [25], there is a planar homography ($\mathbf{H}$) relating the projected images, when these images are captured by cocentered cameras or the scenes are coplanar. We denote this with: $\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the corresponding points on the two images. This $\mathbf{H}$ constraint defines a one-to-one mapping of pixel locations between two photos. In this section, we get a rough estimation of the fake regions by enforcing this constraint.

The $\mathbf{H}$ matrix has the degrees of freedom (DOF) of 8, and each pair of the matching points gives 2 DOF, so at least 4 pairs are required to calculate $\mathbf{H}$ using direct linear transformation (DLT) [25]. However, this linear algorithm is vulnerable to errors. Usually more pairs are used for a robust estimation. In this work, SIFT [26] is used to find initial matching points, and an automatic purification step is introduced before $\mathbf{H}$ calculation since there are outliers. Alternatively, Wang and Farid [15] use the Harris detector to locate interest points and use standard optical flow to track them. This method is not applicable since the optical flow is not always computable in our case due to the large displacements. In addition, the wrong matches are removed manually in [15], which limits its use as an automatic detection. Compared with [15], our purification can be done automatically and effectively.

A. Purification of Matching Pairs

Theoretically, with authentic images and perfect matching points between them, $\mathbf{H}$ can be calculated correctly. However, there are mainly two categories of wrong matches.

1) The first kind of wrong matches are from the false matches by SIFT. SIFT may not work very well with wide baseline or large rotation angle [27]. Estimation with wrong

matches leads to a wrong $\mathbf{H}$ of course. RANSAC is known as a robust estimation method towards the samples with outliers and suits our case well. In this work, we adopt RANSAC to remove wrong matches caused by SIFT.

2) The second category denotes the "wrong" matches within the fake region. Note that these "wrong" matches are not caused by SIFT but the fake region in the images. We can not avoid this error, but we can minimize the impact. When two pictures with complex fake regions are to be examined, SIFT might return dense matching points mainly from the fake region. If all of the matches are used to estimate $\mathbf{H}$, the confusion between the original region and the fake region is likely to occur. That is, we may regard the fake region as authenticated, while the original region as faked, since the majority of the points used to estimate $\mathbf{H}$ are from the fake region. In this work, the Bucketing technique [28] is introduced to avoid the unreasonable confusion.

The Bucketing technique is to make global spatial optimization on selecting matching points. Images are divided into $M \times N$ square buckets; then all the feature points fall into one of the buckets. Buckets with at least one feature point are indexed for later selection. We select correspondences based on the following policies: 1) selection takes place at only the buckets with at least one feature point; 2) one point is allowed from each bucket at most; 3) each bucket votes its point randomly if it has many. After bucketing, the hit ratio for authentic points can be increased sharply.

B. Calculating $\mathbf{H}$

Estimation of the planar homography matrix $\mathbf{H}$ is critical to our detection. After the two-step purification, corresponding points are prepared, and next comes the calculation.

DLT is linear and needs only 4 pairs of corresponding points to calculate the planar homography matrix, but vulnerability is the price it pays. Instead, the Gold Standard Rule [25] is adopted in over-determined cases (more than 4 matching pairs are available to estimate $\mathbf{H}$). The general idea of this method is to minimize the geometric error, and thus finds the best parameter of a certain model. More details on how to calculate $\mathbf{H}$ can be found in [25].

C. Targeting Fake Regions Roughly via $\mathbf{H}$ Constraint

Now that we have the $\mathbf{H}$ matrix, we first recover the rectified image $\mathbf{I}_{1'}$ (Fig. 3(b) upper) from the original image $\mathbf{I}_1$ (Fig. 3(a) upper) using $\mathbf{H}$, since it defines a one-to-one pixel location mapping between the two pictures. Here, inverse mapping and bilinear interpolation are used to get a smooth warped image. After this mapping, common areas of $\mathbf{I}_{1'}$ and $\mathbf{I}_2$ should be the same. Direct subtraction is used to produce the difference map as the metric determining the fake region. Reasons on choosing direct subtraction rather than correlation are based on the following considerations. 1) Correlation magnifies small undesired mismatches between warped images, which are usually borders of objects. Fig. 3 shows good agreement with this argument. 2) Correlation is more computationally expensive than the adopted difference based on pixel subtraction.

Intuitively, before the subtraction operation, common areas of the two rectified photos should be normalized to handle illumination changes. That is, sums of intensities of the two common areas are enforced to be equal. With such normalization, our method is immune to the global illumination changes.

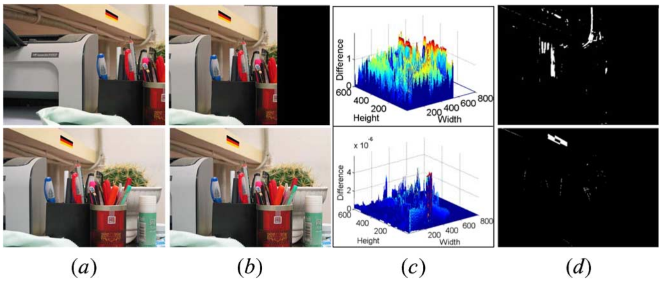


Fig. 3. Comparison between direct subtraction and correlation. (a) The original fake image pair. (b) The rectified image pair. (c) The difference maps using correlation (upper) and direct subtraction (lower). (d) The targeting results using correlation (upper) and direct subtraction (lower).

Practically, there are much more complex illumination cases (Fig. 5), which suffer nonlinear transformations. Removing such transformations are difficult since we have little information on the nonlinear model. However, in most cases, nonlinear transformations can be approximately treated as linear ones for small picture size (in the extreme case, 1 × 1 images can always be treated as linear transformations). Therefore, in this work, we normalize the common areas of the two images piecewisely. The whole image is divided into parts, which are normalized separately. Thus to some extend, our method is robust against complex illumination changes. Fig. 4 shows some results against nonlinear transformations (histogram equalization and gamma correction). With the piecewise normalization [Fig. 4(d)], the pollution is reduced greatly. Fig. 5 shows some real images with unknown transformations. It also works well as expected. Note that the blocking effects in the fake region are the side effects (zoom in Fig. 5(d) for a better view). However, this result has already been good enough for the targeting step. The exact boundaries are extracted later in the segmentation step.

The difference map is further thresholded to a binary map, and the threshold is given by $t = c \times \max(\mathbf{D})$, where $\mathbf{D}$ denotes the difference of frame $\mathbf{I}_2$ and $\mathbf{I}_{1'}$ in common region, and the constant value c usually locates in [0.3, 0.7]. In this work, 0.5 is used through our experiments. Note there are always some tiny and discrete false positives which are wrongly marked as fake regions after filtering, especially along the edges of objects (Fig. 3(d) lower). That is because the corresponding matches cannot be fully purified. Statistically, these mismatches are prone to be heavily clustered in the fake region, while in those authentic regions, mismatches are usually distributed sparsely. So areas with dense high $\mathbf{D}$ are targeted as the fake region.

III. Fake Region Segmentation

In this section, the fake objects are extracted via graph cut [29]. Let us refer to Fig. 2 and assume that the binary map for the rough fake region location has already been obtained in the targeting step. Then a subpicture [Fig. 2(c2)] around the center of the rough fake region is cropped for latter segmentation. Actually in this work, we get the subpicture using the following policy: 1) locate the mean center (x_i, y_i) of the mismatches on the binary map obtained in Section II; 2) starting from (x_i, y_i), expand a square with a minimum side length l, which covers

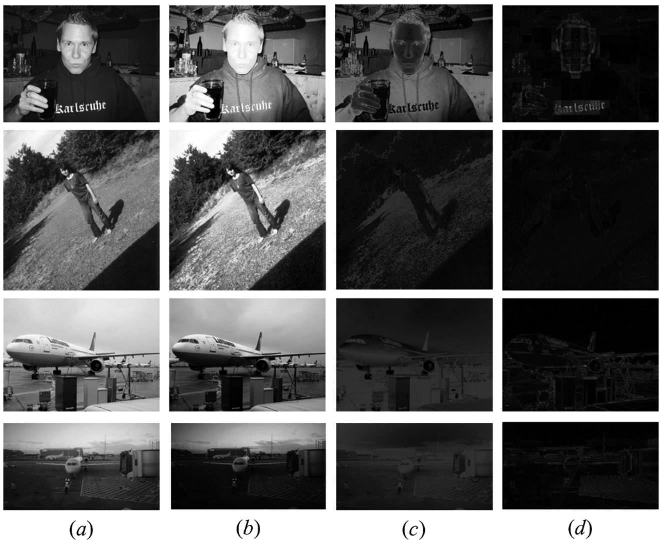


Fig. 4. Performance against nonlinear illumination changes. (a) The original images. (b) The images after nonlinear transformations. The first two rows undergo histogram equalization, while the last two rows suffer gamma correction. (c) The subtraction results without piecewise normalization. (d) The subtraction results with piecewise normalization.

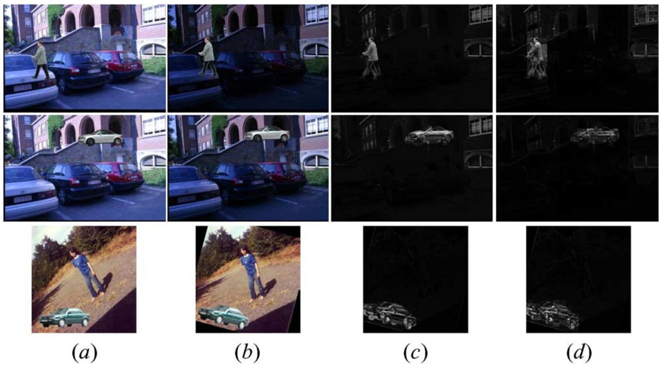


Fig. 5. Performance against complex illumination changes. (a), (b) The original fake image pair. (c) Targeting results using global normalization. (d) Targeting results using piecewise normalization.

80% of all mismatching points; 3) further extend the square with a margin $m = 0.5 \times l$.

To achieve an automatic cutting, we introduce the online selection framework for different features and parameters. As we know, different cues usually hold different powers separating a desired object from its surrounding. Online selection is to choose a set of features/parameters automatically which maximizes the quality function

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}\in\mathbf{\Omega}} \mathbf{Q}(\mathbf{p}) \tag{1}$$

where $\mathbf{p}$ is the parameter vector; $\mathbf{\Omega}$ is the parameter space for $\mathbf{p}$; and $\mathbf{Q}(\mathbf{p})$ is the quality function that gives higher score to better result.

In this section, we will start by reviewing the graph cut briefly, and then focus on the online feature/parameter selection framework which improves the graph cut.

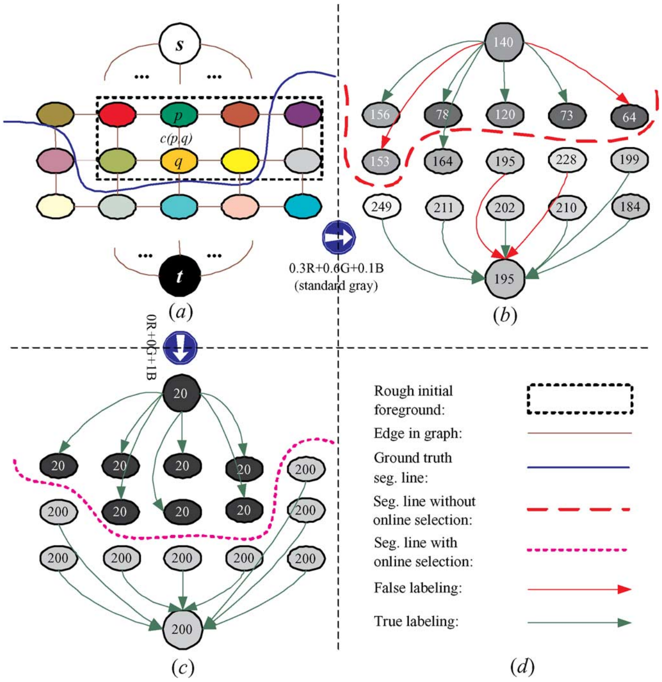


Fig. 6. Online feature selection during graph cut. (a) The graph constructed given an image $\mathbf{I}$. (b) The graph cut result without online feature selection. The standard gray $(0.30\mathrm{R} + 0.59\mathrm{G} + 0.11\mathrm{B})$ is used as the feature for computing the data and smoothness terms in (3) and (4). (c) The graph cut result with online feature selection. The online selected feature $(0\mathrm{R} + 0\mathrm{G} + 1\mathrm{B})$ enhances the differences between the foreground and background. Note that for better illustration, only half of the data links are shown in (b) and (c). (d) The legend for this figure.

A. *Graph Cut*

Graph cut [29] is a well-known effective segmentation method, which formulates the cutting problem into the energy minimization problem, solved by the maximum-flow/minimum-cut theory.

Given an image $\mathbf{I}$ with some fake region inside, a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is constructed as shown in Fig. 6(a). $\mathcal{V}$ is the set of nodes corresponding to the pixels in $\mathbf{I}$, and an edge $(p, q) \in \mathcal{E}$ connects every pair of four-neighboring pixels. Note that $\mathcal{V}$ also contains two extra nodes called the source $(\mathbf{s})$ and the sink $(\mathbf{t})$, which are the labels for the foreground (the fake region) and background, respectively. Every other node in $\mathcal{V}$ has two edges connecting to $\mathbf{s}$ and $\mathbf{t}$. Each edge $(p, q) \in \mathcal{E}$ has a corresponding cost $c(p, q)$, which indicates the dissimilarity of the two nodes. Various functions can be used measuring the cost for a given edge (p, q), such as the ad-hoc function [30] $(p, q) = \exp(-(p - q)^2/2\sigma^2)/dist(p, q)$, where σ is the factor of "camera noise," or alternatively [31] $c(p, q) = \exp(-(p - q)^2/2\sigma^2)$. Thus, the segmentation problem is formulated as a maximum-flow/minimum-cut problem in graph $\mathcal{G}$.

Let L be a labeling of an image; the energy function is formulated as [29]

$$\mathbf{E}(L) = \sum_{p\in\mathcal{P}} D_p(L_p) + \lambda \sum_{(p,q)\in\mathcal{N}} V_{p,q}(L_p, L_q) \tag{2}$$

where $\mathcal{P}$ is the set of all pixels in $\mathbf{I}$, $\mathcal{N}$ is the set of all pairs of four-neighboring pixels, L_p denotes the label for the pixel

p, $D_p(L_p)$ is the data term, and $V_{p,q}(L_p, L_q)$ is the smoothness term. Generally, $D_p(\cdot)$ indicates individual label-preferences of pixels based on observed intensities and prespecified likelihood function, while $V_{p,q}(\cdot)$ encourages spatial coherence by penalizing discontinuities between neighboring pixels [29]. That is, the data term tends to segment an image into small sharp pieces, while the smoothness term tends to segment an image into bigger blocks with gentle boundaries. The relative weight λ between them controls the tradeoff, which is highly picture-dependent for a pleasing segmentation. In this work, the data term and smoothness term are defined as

$$D_p(L_p) = \exp\left(-s_1|C_p - C_t|\right) - \exp\left(-s_1|C_q - C_s|\right) \quad (3)$$

$$V_{p,q}(L_p, L_q) = 0.5 \cdot \exp\left(-s_2|C_p - C_q|\right) \quad (4)$$

where s_1 and s_2 are constants, and C_p, C_s, and C_t are the colors of p, source, and sink, respectively. In our implementation, the constants were set as $s_1 = 0.01$, and $s_2 = 0.02$.

B. Online Feature Selection for Graph Cut

Different from the interactive graph cut, performance of the automatic segmentation depends mainly on the distinction of the foreground and background. Let $\mathbf{F}$ be the $W \times H \times d$ dimensional feature image extracted from original image $\mathbf{I}$ and binary map $\mathbf{D}$

$$\mathbf{F}(x, y) = \mathbf{\Phi}(\mathbf{I}, \mathbf{D}, x, y) \quad (5)$$

where the function $\mathbf{\Phi}$ could be any mapping such as color channels, saliency, texture scores, etc., each of which holds different separating power. In this work, we use the color channels as the seeds to extract the feature image, since the linear combination has a relative low computation cost and holds good separating power

$$\mathbf{F} = \{\omega_1\mathbf{R} + \omega_2\mathbf{G} + \omega_3\mathbf{B} | \omega_i \in \{-2, -1, 0, 1, 2\}\}. \quad (6)$$

Note that there are only 49 distinctive features left, rather than 5^3, after pruning those linearly dependent combinations. This set of candidate features is chosen because [23]: 1) the features are efficient to compute (only integer arithmetic); 2) the features approximately uniformly sample the set of 1-D subspaces of 3-D RGB space; and 3) some common features from the literature are covered in the candidate space, such as raw R, G, and B values, intensity R+G+B, approximate chrominance features such as R-B, and so-called excess color features such as 2G-R-B. In this work, we adopt a coarse-to-fine search procedure to refine the feature selection. First, the best $\omega = (\omega_1, \omega_2, \omega_3)$ with maximal separating power is selected coarsely among the 49 discrete positions (the interval is 1). Next we search a better ω' near ω in the parameter space with a smaller interval. Then, normalization is taken to map $\mathbf{F}$ always onto [0, 255]

$$\mathbf{F} = \left(\mathbf{F} - 255 \times \sum \omega_-\right) \Big/ \sum_{i=1}^{3} |\omega_i| \quad (7)$$

where ω_- denotes the negative entries among $\{\omega_i | i = 1, 2, 3\}$. For example, if $\omega = [-1, -2, 2]$, then $\sum \omega_- = -3$. Intuitively, $(\mathbf{F} - 255 \times \sum \omega_-)$ maps $\mathbf{F}$ onto unsigned integers, and the division by $\sum_{i=1}^{3} |\omega_i|$ maps $\mathbf{F}$ linearly onto [0, 255].

Note our method also requires specifying the initial foreground area to run automatically, but that does not mean our method requires human interaction. The initial foreground is defined by the level set [32], [33] automatically on the difference map obtained in Section II. The level set is an effective contour finding method which addresses the problem (2-D) in a higher dimension (3-D). That is, the contour can be regarded as the intersection of a 3-D shape and a 2-D hyperplane (that is, a closed 2-D curve with the same level). In our implementation, the initial contour is defined as a rectangle, which has the same size as the image, on the difference map generated after the subtraction. The contour shrinks at each iteration, and finally to be a closed curve containing the mismatches after several iterations.

Although the difference map may indicate two contours and each is incomplete (Fig. 3(d), lower), it is good enough for an initial guess for the foreground since the high recall rate is guaranteed. Naturally, the background is defined by an outer rectangle, which extends $0.5 \times \max(w, h)$ pixels from the inner foreground bounding box, with w and h denoting the width and height of the inner bounding box for foreground.

Next we turn to construct the quality function which scores a certain feature. As long as the foreground and background are defined, the discrete probability distributions $p(i)$ for foreground and $q(i)$ for background can be obtained. Further, a log likelihood of $L(i)$ is computed as

$$L(i) = \log \frac{\max\left(p(i), \delta\right)}{\max\left(q(i), \delta\right)} \quad (8)$$

where δ is a small positive number to avoid zeros at both the numerator and denominator. Then the variance ratio (VR) is formulated as

$$\mathrm{VR}(L; p, q) = \frac{\mathrm{var}\left(L; (p+q)/2\right)}{\mathrm{var}(L; p) + \mathrm{var}(L; q)} \quad (9)$$

where $\mathrm{var}(x; r)$ is the variance of x with respect to a probability distribution r

$$\mathrm{var}(x; r) = \sum_i r(i)\left[x^2(i)\right] - \left[\sum_i r(i)\left[x(i)\right]\right]^2. \quad (10)$$

VR indicates the degree of separation, since large VR indicates that colors in the foreground and background are tightly clustered (low intraclass variance), and the differences between foreground and background are enlarged (high interclass variance). In Fig. 7, all 49 combinations are shown in (d) with their VR labeled, which indicates agreement with (9). Fig. 8 shows the segmentation (graph cut) results for the top seven VR images in Fig. 7(d), first row. With online feature selection, the object becomes more separable during the running time. Also in Fig. 9, there are some examples demonstrating the benefits.

Fig. 6 shows how online feature selection improves graph cut. Graph cut without human supervision is challenging since the shapes, colors, and perspectives vary from object to object in the real world. It is unlikely to satisfy such variety with a fixed standard. A foreground may cover a large range of colors (for example: a person with a red jacket and green pants), and so does the background. In such cases, it is unlikely to define a

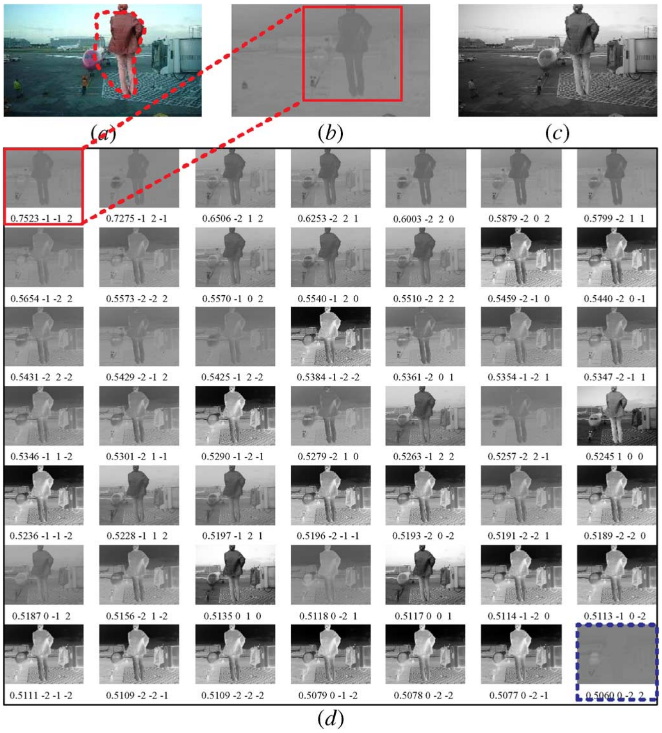


Fig. 7. VR indicates the degree of separation. (a) The original color image with rough initial foreground in red dashed curve. (b) The selected feature image with highest VR. (c) The standard gray image. (d) All feature images in 49 combinations which are listed in descending order in VR (top to bottom, left to right). Labels below each feature image are in format: $(\mathrm{VR}\ \omega_1\ \omega_2\ \omega_3)$. The feature images with highest and lowest VR are marked with red rectangle and blue dashed rectangle, respectively.

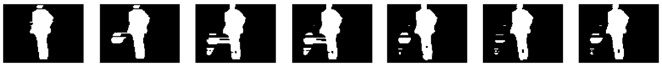


Fig. 8. Segmentation results for the feature images in Fig. 7(d) first row.

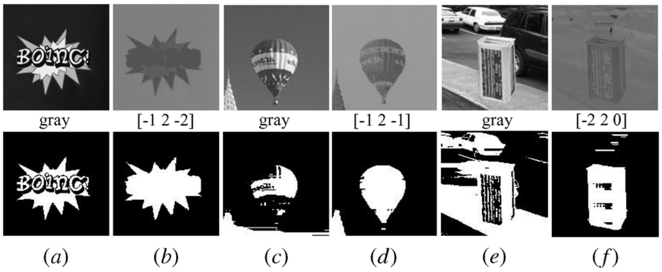


Fig. 9. Benefits from online feature selection. Row 1: Input images for graph cut. Row 2: The graph cut results corresponding to Row 1. (a), (c), (e) Standard gray images. (b), (d), (f) Feature images selected via online feature selection. The weights for [R, G, B] in (6) are labeled under each image.

source/sink value which can separate them apart in the original picture. However, online feature selection can find the best combination, which holds the most power separating the desired object from its surroundings, or at least alleviates the difficulties, thus making the segmentation better. Note all it requires is just a rough initial definition for the foreground [dashed black rectangle in Fig. 6(a), or the red curve in Fig. 7(a)]. This definition does not have to be precise and complete, but needs to contain most true foreground pixels.

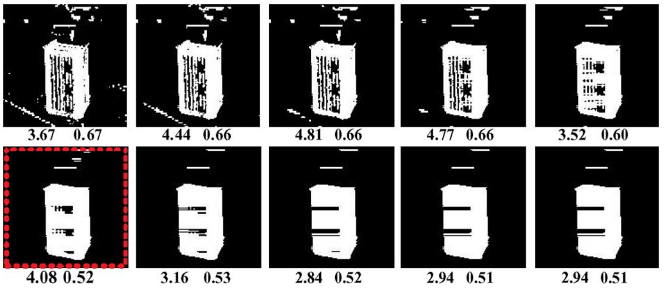


Fig. 10. Cutting results for the image in Fig. 9(e). The scores labeled under each image are in the format: $[F_i\ F_g]$. The best parameter combination in (13) is marked in red dashed rectangle.

C. Online Parameter Selection for Graph Cut

A fixed set of parameters for graph cut is difficult for the uncertainty and variety through different images. Most of the existing automatic methods need several strokes constructing the model of foreground and selecting parameters manually, which, however, are all impractical in fake region detection since we know little about the fake region before we extract it. The initial foreground can be regarded as an alternative to the strokes, and in this section, we introduce a parameter selection framework to achieve automatic graph cut.

There are several parameters to be determined during graph cut, each of which affects the final result. These parameters can be learned during the training process in [22]. However, the global optimal trained parameters are not optimized for each single image; thus, it will not suit every image. In this work, we adopt an online parameter selection framework to optimize the parameters for each image locally since images are known to be highly distinct from each other.

Theoretically, all parameters in graph cut can be selected dynamically using the online parameter selection as long as the quality function is properly defined. Considering the corresponding costs, however, only the most important ones are worth a try. One of them is the weight ratio: λ, between the smooth term and the data term in the energy function. The source and sink can be determined simply using the mean intensity of the foreground and background, respectively. λ is chosen dynamically from a small range, e.g., [0, 0.1], which is sufficient for a desirable result. The quality function which marks scores for different results can be defined considering both the intensity (11) and gradient information (12) [22]

$$F_i(\mathbf{I},\mathbf{S}) = \frac{\sum_{(a,b)\in\mathbf{B}} |\mathbf{I}_a - \mathbf{I}_b|}{|\mathbf{B}|} - \frac{\sum_{(a,b)\in\mathbf{O}} |\mathbf{I}_a - \mathbf{I}_b|}{|\mathbf{O}|} \tag{11}$$

$$F_g(\mathbf{I},\mathbf{S}) = \sum_{(a,b)\in\mathbf{B}} \frac{\left|G(\vec{a}) - G(\vec{b})\right|}{|\mathbf{B}|} \tag{12}$$

where $\mathbf{I}$ is the image after online feature selection; $\mathbf{S}$ is the segmentation result; $\mathbf{B}$ is the set of pairs of neighboring pixels along the object edge: $\mathbf{B} = \{(a,b)|\mathbf{S}_a = 1, \mathbf{S}_b = 0 \text{ or } \mathbf{S}_a = 0, \mathbf{S}_b = 1\}$; $\mathbf{O}$ is the set of pairs of neighboring pixels inside the object region, $\mathbf{O} = \{(a,b)|\mathbf{S}_a = 1, \mathbf{S}_b = 1\}$; $|\mathbf{N}|$ is the length of the set $\mathbf{N}$; and $G(\vec{a})$ is the normalized gradient at point

Fig. 11. Sample images in PASCAL 2008 dataset.

a. For a good segmentation, F_i should be positive and larger to encourage high contrast borders, while a smaller F_g is preferred to get smooth boundaries. An index of the best result is given by

$$\hat{id} = \arg\max_{id} \left(\mathrm{rank}(F_i, id, D) + \mathrm{rank}(F_g, id, A)\right), \quad (13)$$

where $\mathrm{rank}(F, id, M)$ is the rank of the idth image with respect to the feature F, when sorted in mode M: ascending (A) or descending (D). Fig. 10 shows that this method takes advantages from both quality functions in (11) and (12).

Finally, we select the most reasonable result according to (13) as the segmentation output. Then the final result is a fusion of the targeting and segmentation outputs, which takes advantages from both sides. Automatic segmentation is challenging and sensitive among various images, while the targeting output is more stable. Usually the targeting step gives better recall but low precision [Fig. 2(c2)], while the segmentation step is to refine the precision. If segmentation returns a much different result from targeting step, it is more likely that the segmentation fails. So, in our implementation, the fusion is defined as

$$\mathrm{Output} = \begin{cases} \mathrm{Output}_{\mathrm{tar}} \,\&\, \mathrm{Output}_{\mathrm{seg}}, & p \geq T, \quad (14) \\ \mathrm{Output}_{\mathrm{tar}}, & p < T. \quad (14') \end{cases}$$

$$p = \frac{|\mathrm{Output}_{\mathrm{tar}} \,\&\, \mathrm{Output}_{\mathrm{seg}}|}{|\mathrm{Output}_{\mathrm{tar}}|} \quad (15)$$

where $\mathrm{Output}_{\mathrm{tar}}$ and $\mathrm{Output}_{\mathrm{seg}}$ are the outputs after targeting and segmentation steps, respectively, and T is a predefined threshold, which is 0.4 throughout our experiments.

IV. Experimental Results

In this section, performance of our method is evaluated on both semisimulated and real cases.

A. Datasets

PASCAL 2008, OXFORD's affine covariant feature images, IG02 images, and real images with composites are used in our experiments.

PASCAL 2008 images [34] are used to evaluate the performance of our method with respect to different scales, rotations, and offsets. As shown in Fig. 11, this dataset covers a large range of natural scenes, including people/animal, indoor/outdoor, and trees/flowers. In our experiments, 5096 images with an average resolution 500 × 300 (pixels) are used.[1]

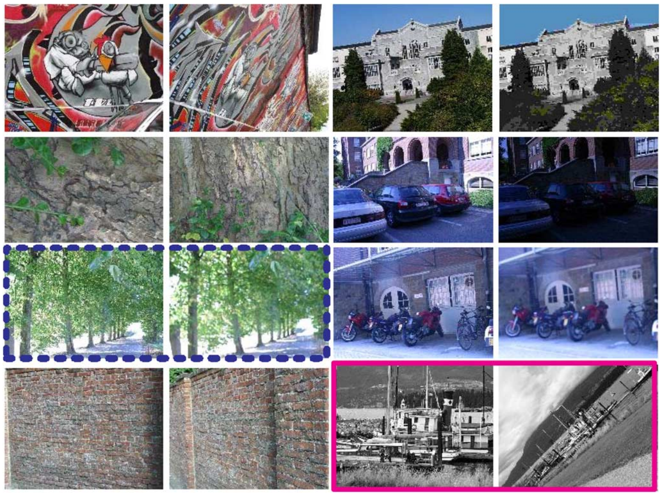


Fig. 12. Sample images in OXFORD dataset (From top to bottom, left to right: GRAF, UBC, BARK, LEUVEN, TREES, BIKES, WALL, and BOAT).

OXFORD's affine covariant feature images[2] contain a set of challenging image pairs with tough affine transformations, which are originally used for the performance evaluation of feature detectors/descriptors. Fig. 12 shows the sample images, which cover blur effects, viewpoint variations, zoom, rotation, lighting changes, and JPEG compression. All of the image pairs satisfy the planar homography constraint approximately. The affine transformations in this dataset are generally tough:[3] 1) the viewpoint changes up to 60° (GRAF, BARK, WALLS, BOAT); 2) the scale changes by about a factor of 4 (BARK, BOAT); 3) the lighting changes are introduced by varying the camera aperture (LEUVEN); 4) the JPEG compression group is generated using a standard xv image browser with the image quality parameter varying from 40% to 2% (UBC); 5) the blur group is acquired by varying the camera focus (TREES). There are eight groups and each has five image pairs, with an average resolution of 800 × 640 pixels. Note that the images are either capturing the planar scenes or acquired by cocentered cameras, so that the images are related by homographies. However, there are also some regions that do not satisfy the planar homography, e.g., the waving leaves in TREES, the driving car in GRAF, and the walking people in BOAT. Without surprise, these moving local regions together with the above-mentioned significant distortions challenge the proposed method. Therefore, we use this dataset to evaluate the performance of our method in the worst case scenarios.

The IG02[4] (INRIA Annotations for Graz-02) [35], [36] dataset is used as the source images in our experiments, as the ground truth segmentation is available. IG02 is a popular natural-scene object category dataset. Some sample source images are shown in Fig. 13. With the help of the provided masks, we extract the objects to be copied as the source objects. Then

[1] Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/.

[2] Available: http://www.robots.ox.ac.uk/~vgg/research/affine/index.html.

[3] Available: http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/DataREADME.

[4] Available: http://lear.inrialpes.fr/people/marszalek/data/ig02/.

Fig. 13. Sample source images. The objects are extracted from the original images in IG02 based on their corresponding masks.

TABLE I
PERFORMANCE ON PASCAL 2008 WITH AN OFFSET OF 10

Rotation/Zoom	Precision	Recall	Impr. P	Impr. R
0/1.0	77.03%/81.68%	90.12%/91.77%	6.04%	2.01%
15/1.2	73.92%/78.72%	86.74%/88.62%	6.49%	2.17%
30/1.4	65.84%/70.67%	81.66%/82.78%	7.33%	1.37%
45/1.6	58.40%/61.69%	75.16%/76.43%	5.64%	1.70%
60/1.8	53.89%/57.04%	69.77%/71.86%	5.84%	3.00%
75/2.0	49.23%/51.69%	65.13%/68.17%	5.00%	4.67%

TABLE II
DISTRIBUTION OF THE PERFORMANCE IN Table I

R/Z	Precision			Recall		
	$> 80\%$	$50\% \sim 80\%$	$< 50\%$	$> 80\%$	$50\% \sim 80\%$	$< 50\%$
0/1.0	53.60%	45.20%	1.20%	88.80%	9.00%	2.20%
15/1.2	47.20%	50.12%	2.68%	82.97%	13.63%	3.41%
30/1.4	38.69%	47.20%	14.11%	69.10%	24.82%	6.08%
45/1.6	29.20%	37.80%	33.00%	55.31%	31.46%	13.23%
60/1.8	28.60%	34.80%	36.60%	49.19%	31.50%	19.31%
75/2.0	24.00%	31.20%	44.80%	41.68%	37.17%	21.15%

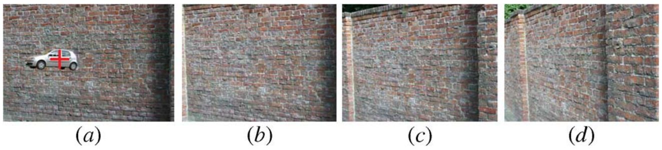


Fig. 14. Images used to test the user's ability to position an object correctly. The users are asked to insert the provided car [the car in (a)] into (b)–(d) according to (a). (a) An image (resolution: 1000 × 700) with a car inserted. (b)–(d) Real images (resolution: 880 × 680) having a certain homography relationship with (a) (without the inserted car). Images used in this experiment and their ground truth homographies are available in http://www.robots.ox.ac.uk/~vgg/research/affine/index.html.[2]

these extracted source objects are inserted into the target image (PASCAL 08 and OXFORD's affine) as the faked objects.

Also in our experiments, real images with visually plausible composites are either self-taken or are from the internet.

In our experiments, the precision and recall rates for a result $\mathbf{EST}$ (estimated) are defined as

$$\text{Precision} = \frac{|\mathbf{EST} \bigcap \mathbf{GT}|}{|\mathbf{EST}|} \times 100\% \tag{16}$$

$$\text{Recall} = \frac{|\mathbf{EST} \bigcap \mathbf{GT}|}{|\mathbf{GT}|} \times 100\% \tag{17}$$

where $\mathbf{EST}$ is the fake region we get, $\mathbf{GT}$ is the groundtruth fake region, and $|\mathbf{S}|$ denotes the number of nonzero pixels in $\mathbf{S}$. Note $\mathbf{EST}$ and $\mathbf{GT}$ are both binary images, in which the fake region is marked as 1 and 0 for other pixels.

B. Performance With Respect to Scales, Rotations, and Offsets

We first test our method on the PASCAL 2008 image dataset. Note that PASCAL does not have image pairs satisfying the $\mathbf{H}$ constraint. So we synthesize the second image from the existing one by rotation and zooming operation. Then the source objects are inserted into the target photos with offsets.

The average precision and recall rates with respect to different rotation angles and zooming factors are shown in Table I. Numbers in precision/recall columns are in the following format: (mean result with fixed feature&parameter)/(mean result with online feature&parameter selection), in short (fixedResult/onlineResult). Columns 4 and 5 are the improvement ratios in precision and recall, respectively, which are calculated as: (onlineResult-fixedResult)/fixedResult. Note the rotation angle (θ) here is equally divided into three parts: pitch $(\theta/3)$, roll $(\theta/3)$, and yaw $(\theta/3)$, and the offset is fixed at 10 pixels. Table II shows the detailed distribution of the performance. As shown, the precision and recall are generally decreasing when rotation/zooming grows. This is due to the limitation of the current state of the art feature descriptors in the case of wide baseline as also argued in [27]. The overall average precision and recall rates are 66.92% and 79.94%, respectively.

To evaluate the performance against different offsets, we first test the ability of typical users to position objects in an image pair from a simple user study. In this study, an image [Fig. 14(a)] with an inserted car is shown to 10 normal users (undergraduate students majored in computer science). These users were told to position and morph the same car into the distorted images [Fig. 14(b)–(d)] so that it looks as visually plausible as possible. In this experiment, all users achieve this task by using some photo editing software, i.e., Photoshop in our experiment. To evade detection of the algorithm proposed in the paper, an experienced attacker might be able to generate the tampered image pair by estimating the planar homography first and then insert the transformed object at exactly the same location. During this experiment, however, none of these users estimates the planar homography or is able to calculate the exact shape and position using cues. This observation partially reflects the fact that very few people have the knowledge of multiple view geometry, and the proposed method is applicable to authenticating duplications generated by general users.

After the users return their tampered photos to us, based on the ground truth transformation matrices $\mathbf{H}_i$ $(i = b, c, d)$ from Fig. 14(a) to Fig. 14(b)–(d), we compute the average offset in the x- and y-directions between the red cross $\mathbf{x}_a$ in Fig. 14(a) and the back-projected location $\mathbf{H}_i^{-1}\mathbf{x}_i$ from Fig. 14(b)–(d)'s red cross location $\mathbf{x}_i$. The statistics of these offsets are shown in Table III. Without surprise, it is harder to correctly position objects in an image pair with more distortions.

Next, performance with respect to different offsets is tested on the PASCAL 2008 dataset, as shown in Fig. 15. Based on the simple user study, the offset tested in this experiment is in the range of [0, 30] pixels, since the average resolution in PASCAL is 500 × 350. Note the the fake objects are inserted in ground truth shapes throughout our semisimulations. So when the offset equals 0, our method cannot find the fake region since the fake objects fully overlap with the same authenticate

TABLE III
OFFSET DISTRIBUTION (IN PIXELS) FROM TEN REAL USERS TO POSITION THE CAR IN Fig. 14(a) INTO Fig. 14(b)–(d)

Group	Mean	Standard Deviation	Min	Max
(a) and (b)	8.50	3.69	2.61	13.79
(a) and (c)	18.05	6.59	9.23	30.69
(a) and (d)	34.16	7.98	25.2	53.09

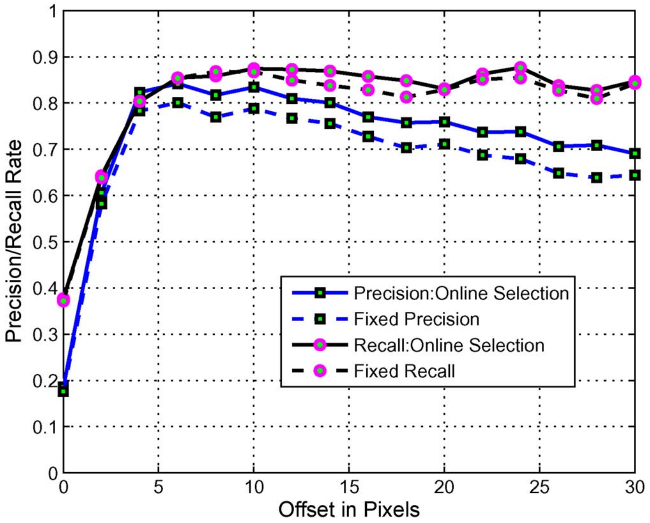


Fig. 15. Performance of our method against different offsets in pixels.

regions. Without surprise, online selection results are mostly better than those without online selection. Average precision and recall reach 71.36% and 83.59%, respectively.

C. Performance With Respect to Tough Image Pairs

Then we test our method on OXFORD's affine covariant feature images. We generate the fake image pair as follows: 1) obtain the target image pair $(\mathbf{A}, \mathbf{B})$, and the groundtruth $\mathbf{H}$ from the OXFORD database directly; 2) randomly select a source image $\mathbf{F}$ from IG02, and insert $\mathbf{F}$ into $\mathbf{A}$ at a random position $\mathbf{p} = (x, y, 1)^{\mathbf{T}}$; 3) generate $\mathbf{F}'$ with $\mathbf{F}' = \mathbf{HF}$, and then insert $\mathbf{F}'$ into B at the position $\mathbf{p}' = \mathbf{Hp} + \mathbf{d_{offset}}$, where $\mathbf{d_{offset}}$ is the offset between the inserted position and the groundtruth position. Finally, we get a pair of images, in which the fake region is inserted with $\mathbf{d_{offset}}$ offset away from the groundtruth position, but with groundtruth shapes.

In total, we have 40 pairs of target photos with known groundtruth $\mathbf{H}$ and 117 fake patches. So we generate 40 × 117 pairs of fake images. Note that during this test, groundtruth $\mathbf{H}$ is blind to our method, and it is only used for generating fake images. The $\mathbf{H}$ matrix used during our detection is calculated using the method described in the targeting step. So the only input is the synthesized fake image pair.

Table IV shows the statistics on the OXFORD dataset with a 10-pixel offset, while Table V shows the distribution of the performance. Globally, all target image pairs used in this experiment satisfy the $\mathbf{H}$ constraint. However, some groups contain moving objects (the driving car in GRAF, walking people in BOAT, and waving leaves in TREES), which do not satisfy the constraint. Precision/recall would be even higher if these images satisfy the $\mathbf{H}$ constraint strictly.

TABLE IV
PERFORMANCE ON OXFORD DATASET WITH THE OFFSET OF 10. DATA IN THIS TABLE ARE IN FORMAT: fixedResult/onlineResul$t$. FIXED RESULTS ARE GENERATED WITH THE CONSTANT FEATURE (GRAY) AND CONSTANT PARAMETER ($\lambda = 0.05$: THE MEAN VALUE OF OUR λ SPACE)

Group	Precision	Recall	Impr. P	Impr. R
BARK	43.37%/47.19%	42.53%/54.45%	8.81%	28.02%
LEUVEN	71.91%/77.02%	68.23%/79.75%	7.11%	16.89%
TREES	14.51%/16.28%	50.44%/37.45%	12.18%	-25.75%
GRAF	15.62%/16.67%	36.46%/61.34%	6.73%	68.25%
BOAT	19.69%/25.81%	47.61%/51.78%	31.04%	8.75%
UBC	65.55%/65.99%	78.57%/78.90%	0.66%	0.41%
BIKES	60.51%/87.48%	68.01%/84.08%	44.58%	23.62%
WALL	42.00%/45.04%	60.48%/68.44%	7.23%	13.15%

TABLE V
DISTRIBUTION OF THE PERFORMANCE IN Table IV

Group	Precision			Recall		
	$> 80\%$	$50\% \sim 80\%$	$< 50\%$	$> 80\%$	$50\% \sim 80\%$	$< 50\%$
BARK	22.40%	27.20%	50.40%	27.20%	31.60%	41.20%
LEUVEN	58.00%	25.20%	16.80%	66.00%	19.20%	14.80%
TREES	6.02%	10.04%	83.94%	28.00%	12.00%	60.00%
GRAF	0.00%	5.26%	94.74%	52.23%	13.77%	34.01%
BOAT	4.00%	12.40%	83.60%	37.60%	18.00%	44.40%
UBC	27.60%	49.20%	23.20%	60.80%	26.40%	12.80%
BIKES	78.40%	14.80%	6.80%	72.40%	16.80%	10.8%
WALL	25.96%	13.62%	60.43%	52.77%	22.55%	24.68%

Precision and recall are pleasing on most tough cases, reaching the highest 76.53% and 79.76%, respectively. Improvements in precision and recall are also obvious on most groups. Note that the BOAT group (Fig. 12 highlighted in magenta rectangle) is gray only. Although the faked object is in color, online feature selection contributes trivially, and the improvement mainly comes from the online parameter selection within this group.

As expected, the performance of our method degenerates in the case where the assumptions do not hold. First, $\mathbf{H}$ is not able to be accurately estimated due to: 1) the large viewpoint changes in GRAF and WALL; 2) the huge scales in BOAT and BARK; and 3) the large JPEG compression/blur in UBC and TREES. Second, moving textures in TREES, GRAF, and BOAT violate our assumption that the scenes are rigid. As shown in Fig. 16, some bad targeting results are shown in (c). The negative example of online selection happens in the recall rate of the TREES group as highlighted in the blue dashed rectangle in Fig. 12. This failure is due to the rich textures and "moving" leaves. At the end of Section II, we mentioned small false positives existing along object edges because of the imperfect estimation of the planar homography $\mathbf{H}$ and the interpolation in the warping step. In the TREES group, there are dense edges between the leaves and the sky. In addition, the intensity changes across these edges happen to be high. Moreover, the leaves are waving. Thus the result after subtraction contains large areas of false positives [Fig. 16(c)]. That is, the targeting step fails on such cases and online selection starts from a bad foreground as the red dashed rectangles in Fig. 16(c), although online selection did its job and made this wrong "foreground" more separable from the "background."

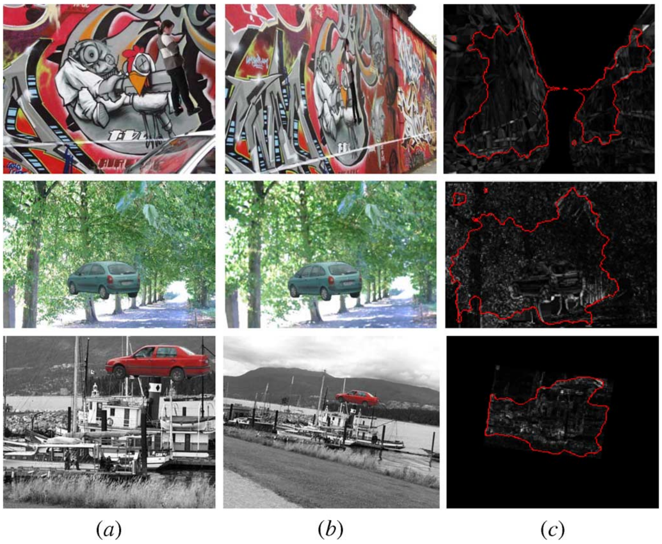


Fig. 16. Targeting results on GRAF, TREES, and BOAT. (a), (b) Input images. (c) The targeting results are marked as the red solid curve. The targeting step is sensitive when there are: 1) rich details; 2) high contrast across the details' edges; and 3) cases when the homography is hard to estimate.

D. Performance With Respect to Visually Plausible Images

At last, our method is tested on images with visually plausible composites. In this section, comparisons are made among the following results: 1) targeting results without segmentation in Section II; 2) results using automatic graph cut with fixed features and parameters; and 3) results using automatic graph cut with online feature/parameter selection.

As shown in Fig. 17, Rows 1 ~ 5 are examples with rotation and zooming that satisfy $\mathbf{H}$ constraint; Row 6 is a planar scene case which can also be detected using the proposed method. Note all the results shown in Fig. 17 are generated without resetting any parameters among different runs, which indicates we can get a pleasing result with the optimal parameters since the most important features/parameters are selected dynamically via the online selection framework.

The segmentation step takes Fig. 17(c) as inputs. Note that the boundaries can be extracted as long as (c) indicates a rough estimation for the fake region.

Note that the method in [12] will fail because the inserted balloon satisfies lighting conditions (row 1), and the correlation matching method in [5] is not applicable since the inserted objects are from unknown sources.

Results in Fig. 17(c) are improved with the segmentation step, and the results in (e) with online selection are better than those with fixed features and parameters (d). Note that the binary map might be slightly different through different runs since RANSAC is used. However, results in Fig. 17(e) are generally better than those in (d) with the same binary mask used during each run.

Also, we extend our method to the single-image case. The new algorithm is almost the same with the two-image case except: 1) single image performs a self-matching which finds the matches within itself; 2) the single-image case does not need segmentation since the result after targeting step is already pleasing; and 3) the region with zeros after subtraction is the fake region, rather than the nonzero region. Fig. 18 shows some results for the single-image case. Some of the duplications are hardly noticeable by human eyes, but our method works as well.

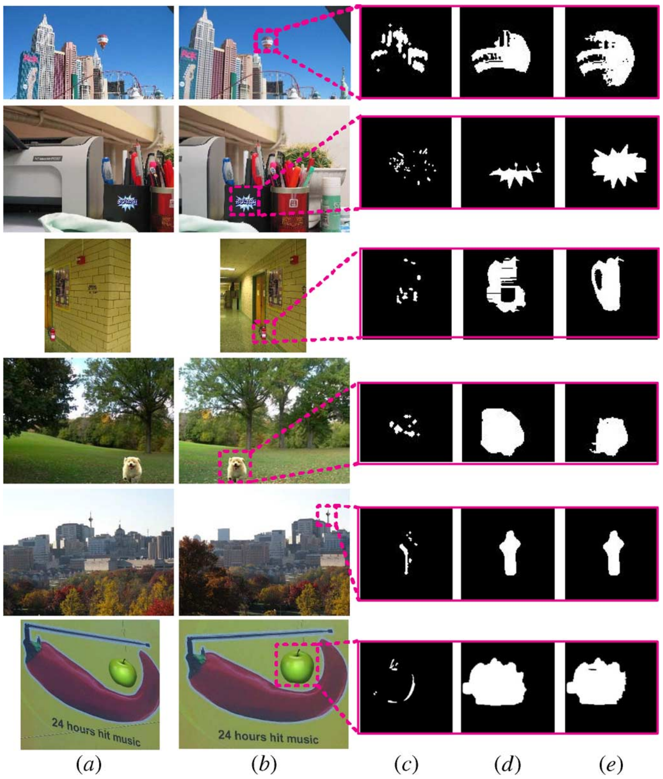
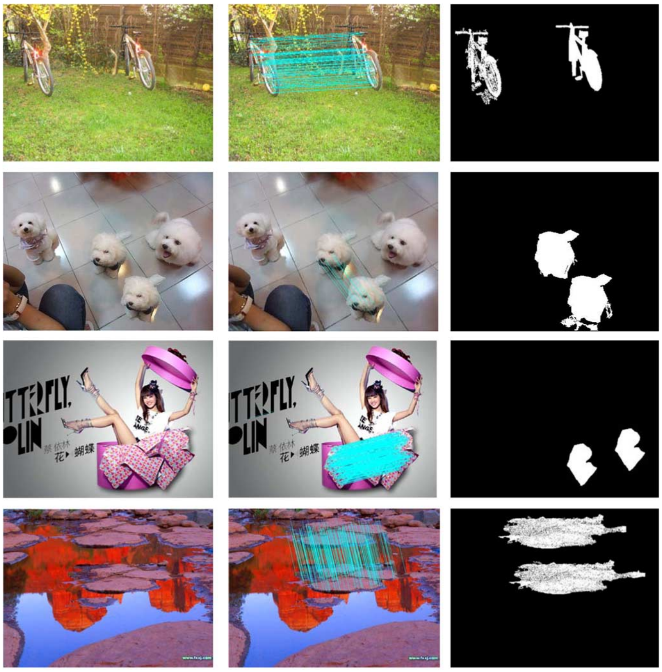


Fig. 17. Experimental results of our method. (a), (b) Original image pairs. (c) Results after the targeting step. (d) Automatic graph cut results without online feature/parameter selection. (e) Results with proposed method. Subpictures for the fake regions are marked with dashed magenta rectangles.

The last experiment is to compare our single-image method with [4]. Some results are shown Fig. 19. The first observation is that our method provides similar results as [4] in most cases, e.g., top two rows. However, when the test image has large regions of repeated textures, [4] might result in false alarms as shown in the last row of Fig. 19. In terms of time complexity, our method first estimates the translation using SIFT descriptors, which can be done efficiently using the binary code from the SIFT authors. Suppose we have S SIFT points extracted from the original image, the matching process can be done in $O(S^2)$. Then the duplicated regions are located based on direct subtraction, with the cost of $O(MN)$, where M and N are the width and height of the image, respectively. In [4], they extract the features in a $(M-B+1)(N-B+1)\times(BB)$ matrix which can be sorted in $O(MN\text{Log}(MN))$. Here B denotes the side length of the minimal segment square, which is typically much smaller than M or N. The translation (shift vector) in [4] can be calculated in $O(MN)$ in the best case and $O((MN)^2)$ in the worst case. So the time complexities of both methods are in the same magnitude. Based on our implementation, our method runs slightly faster than [4] since S is usually much smaller than MN.

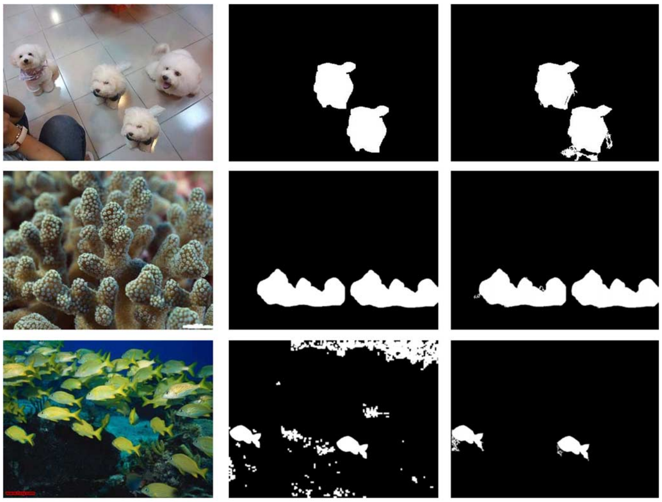
V. Conclusion

In this work, an automatic fake region extraction method was proposed using the planar homography constraint and graph cut.



(*a*) (*b*) (*c*)

Fig. 18. Experimental results for duplications within single image. (a) The original images with composites. (b) The self-matching pairs of the images. (c) The results denoting the fake region.



(*a*) (*b*) (*c*)

Fig. 19. Comparison between our single image method and [4]. (a) The test images. (b) The results by [4]. (c) Our results.

The photographic composites are first targeted with the geometrical constraint, and then extracted via graph cut. To improve the accuracy and automation, an online selection framework was adopted during the segmentation. The geometry-based targeting method is stable and effective, and the proposed segmentation method further improves the results. Results generated with online selection framework are generally better than those without it. Actually, the online feature selection acts like a preprocessing step, while the online parameter selection a postprocessing step for graph cut, which both improve the performance of graph cut. Note this segmentation technique can also be applied to other forensic detection methods, such as [8] and [19].

As argued in [4] that a single forensic tool considered separately may not always be reliable to provide sufficient evidence for all types of digital forgery, the fusion of various tools may give a comprehensive estimation. Our method may serve as one of the tools, and also has its limitations. For example, the present approach is not suitable for the cases when a pair of forged images have fake objects inserted almost at perfect positions and in perfect shapes, when a single faked image has no duplicate region, when the scenes contains nonrigid objects, and when few feature points (less than four pairs) can be found at the faked objects. These limitations are to be considered in our future work.

References

[1] I. J. Cox, M. L. Miller, and J. A. Bloom*, Digital Watermarking*. San Mateo, CA: Morgan Kaufmann, 2002.

[2] H. Liu, J. Rao, and X. Yao, "Feature based watermarking scheme for image authentication," in *IEEE Int. Conf. Multimedia and Expo*, 2008, pp. 229–232.

[3] H. Farid, "A survey of image forgery detection," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[4] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Research Workshop*, Cleveland, OH, Aug. 2003, vol. 8.

[5] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *ACM Multimedia and Security Workshop*, 2007, pp. 35–42.

[6] J. Kornblum, "Using jpeg quantization tables to identify imagery processed by software," *Digital Investigation*, pp. s21–s25, 2008.

[7] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting doctored jpeg images via dct coefficient analysis," in *Proc. Eur. Conf. Computer Vision*, 2006, pp. 423–435.

[8] H. Farid, "Exposing digital forgeries from jpeg ghosts," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 4, pp. 154–160, Dec. 2009.

[9] Z. Lin, R. Wang, X. Tang, and H. Y. Shum, "Detecting doctored images using camera response normality and consistency," in *Proc. Computer Vision and Pattern Recognition*, 2005, pp. 1087–1092.

[10] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Security Forensics*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[11] J. Fridrich, "Digital image forensic using sensor noise," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 26–37, Mar. 2009.

[12] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *ACM Proc. 7th Workshop on Multimedia and Security*, 2005, pp. 1–9.

[13] W. Zhang, X. Cao, J. Zhang, J. Zhu, and P. Wang, "Detecting photographic composites using shadows," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2009, pp. 1042–1045.

[14] M. Johnson and H. Farid, "Detecting photographic composites of people," in *Proc. Int. Workshop on Digital Watermarking*, Guangzhou, China, 2007.

[15] W. Wang and H. Farid, "Detecting re-projected video," in *Proc. Int. Workshop on Information Hiding*, 2008, pp. 72–86.

[16] A. Swaminathan, M. Wu, and K. Liu, "Non-intrusive forensic analysis of visual sensors using output images," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2006, vol. 5, pp. 401–404.

[17] A. Swaminathan, M. Wu, and K. Liu, "Image tampering identification using blind deconvolution," in *IEEE Int. Conf. Image Processing*, 2006, vol. 3, no. 1, pp. 2311–2314.

[18] A. Swaminathan, M. Wu, and K. Liu, "Component forensics for digital camera: A non-intrusive approach," in *Proc. Conf. Info. Sciences and Systems*, 2006, pp. 1194–1199.

[19] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. Society of Photo-Optical Instrumentation Engineers Conf. Series*, 2006, vol. 6072, pp. 362–372.

[20] M. K. Johnson and H. Farid, Metric Measurements on a Plane From a Single Image Department of Computer Science, Dartmouth College, TR2006-579, 2006.

[21] M. Liu, S. Chen, and J. Liu, "Precise object cutout from images," in *Proc. ACM Int. Conf. Multimedia*, 2008, pp. 623–626.

[22] B. Peng and O. Veksler, "Parameter selection for graph cut based image segmentation," in *Proc. British Machine Vision Conf.*, Leeds, U.K., 2008.

[23] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[24] W. Zhang, X. Cao, Z. Feng, J. Zhang, and P. Wang, "Detecting photographic composites using two-view geometrical constraints," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2009, pp. 1042–1045.

[25] R. Hartley and A. Zisserman*, Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[27] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors base on 3D objects," *Int. J. Comput. Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[28] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J. Comput. Vision*, vol. 27, no. 2, pp. 161–195, 1998.

[29] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[30] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, vol. 1, pp. 105–112.

[31] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 416–423.

[32] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 9, pp. 12–49, 1988.

[33] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *IEEE Int. Conf. Computer Vision Pattern Recognition (CVPR)*, San Diego, CA, vol. 1, pp. 430–436.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html

[35] M. Marszalek and C. Schmid, "Accurate object localization with shape masks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[36] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 416–431, Mar. 2006.

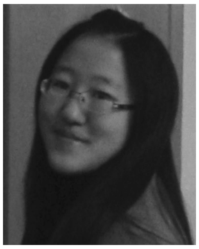
Wei Zhang was born in China, in 1986. He received the B.E. degree in software engineering from the School of Computer Software, Tianjin University (TJU), Tianjin, China, in 2008.

He is currently a second-year graduate student with the School of Computer Science and Technology, TJU. He has been working as a research assistant in the Computer Vision Laboratory since 2008. His research interests lie in the computer vision field, covering mainly image processing, multimedia forensic, and pattern recognition.

Xiaochun Cao received the B.E. and M.E. degrees both in computer science from Beihang University (BUAA), Beijing, China, in 1999 and 2002, respectively. He received the Ph.D. degree in computer science from the University of Central Florida, Orlando, in 2006, with his dissertation nominated for the university-level award for Outstanding Dissertation.

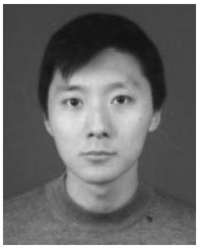
After graduation, he spent about two and half years at ObjectVideo Inc. as a Research Scientist. Since August 2008, he has been with Tianjin University, Tianjin, China, where he is currently Professor of Computer Science. He has authored and coauthored over 40 peer-reviewed journal and conference papers, and has been in the organizing and the technical committees of several international colloquia.

In 2004, Dr. Cho was a recipient of the Pierro Zamperoni best student paper award at the International Conference on Pattern Recognition.



Yanling Qu was born in Shandong Province, China, in 1985.

She is currently a second-year graduate student in the Department of Computer Science and Technology, Tianjin University, China. Her research interests lie in image processing, computer vision, and medical imaging processing.



Yuexian Hou received the B.Eng. degree in computer science, the M.Eng. degree in computer science, and the Ph.D. degree in signal and information processing from Tianjin University, Tianjin, China, in 1995, 1998, and 2001, respectively.

Currently, he is an Associate Professor at the School of Science and Technology, Tianjin University, and a Visiting Professor at the Knowledge Media Institute of the Open University, U.K. He has published more than 30 papers in academic journals and conferences. His current research interests include machine learning, natural language processing, and information retrieval.

Dr. Hou is a senior member of the China Computer Federation (CCF) and serves as a PC member of several international academic conferences.

Handong Zhao was born in March, 1988. He is currently working toward the B.S. degree at the School of Computer Science and Technology, Tianjin University, Tianjin, China.

His research interests include multimedia forensics, information security, particularly digital image authentication, and interpolation theory.

Chenyang Zhang was born in Taiyuan, China, in 1989. He is currently working toward the B.S. degree at the School of Computer Software, Tianjin University, Tianjin, China.

His research interests lie in image forensics, computer vision, and information security.