Detect Digital Image Splicing with Visual Cues

Zhenhua Qu[1], Guoping Qiu[2], and Jiwu Huang[1]

[1] School of Information Science and Technology, Sun Yat-Sen University, China
[2] School of Computer Science, University of Nottingham, NG 8, 1BB, UK
qzhua3@gmail.com, qiu@cs.nott.ac.uk, isshjw@mail.sysu.edu.cn

Abstract. Image splicing detection has been considered as one of the most challenging problems in passive image authentication. In this paper, we propose an automatic detection framework to identify a spliced image. Distinguishing from existing methods, the proposed system is based on a human visual system (HVS) model in which visual saliency and fixation are used to guide the feature extraction mechanism. An interesting and important insight of this work is that there is a high correlation between the splicing borders and the first few fixation points predicted by a visual attention model using edge sharpness as visual cues. We exploit this idea to develope a digital image splicing detection system with high performance. We present experimental results which show that the proposed system outperforms the prior arts. An additional advantage offered by the proposed system is that it provides a convenient way of localizing the splicing boundaries.

1 Introduction

Image splicing or photomontage is one of the most common image manipulation techniques to create forgery images. As shown in Fig. 1, by copying a spliced portion from the source image into a target image, one can create a composite scenery to cheat others. Helped by current state-of-the-art image editing software, even non-professional users can perform splicing without much difficulty. Although experienced experts can still identify not highly sophisticated forgeries, it is still a challenging issue to tackle this complicated problem by machines in a fully automated way.

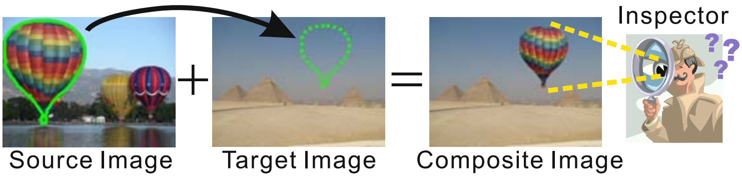


Fig. 1. The process of image splicing forgery

S. Katzenbeisser and A.-R. Sadeghi (Eds.): IH 2009, LNCS 5806, pp. 247–261, 2009.

Existing methods for detecting splicing images can be roughly classified into two categories: boundary based methods and region based methods. The boundary based methods detect the abnormal transient at the splicing boundaries, e.g., a sharp transition. Farid [1] used high order spectral(HOS) analysis to detect the high order correlation introduced by splicing in the harmonics of speech signals and later the idea was extended to image by Ng et al. [2]. By using the Hilbert-Huang transform(HHT) which empirically decomposes a signal into local harmonics and estimates the instantaneous frequency, Fu et al.[3] improved the accuracy from 72%[2] to 80%. Some researchers used the Wavelet analysis to characterizing the short-time transients in signals. Both the Lipsiz regularity [4] and phase congruency approach [5] can define a normalized measure of local sharpness/smoothness from the wavelet coefficients. Some methods further dealt with post-smoothing utilizing the camera response function (CRF)[6,7], or abnormality of local hue [8]. The region based methods generally rely on a generative model of the image and utilize the inconsistent system parameters estimated from the spliced and the original regions to identify the forgery. For images acquired with digital cameras, these generative models try to model lighting[9], optical lens characteristic[10], sensor pattern noise[11], and post-processing algorithm, such as color filter array interpolation[12,13]. For JPEG images, recompression artifacts are also useful features[14]. In this work, we focus on a boundary based method and assume, as in[2,3], that no further processing, e.g., blur or compression, has been applied to conceal the splicing boundary.

If there are sharp changes between a spliced region and its surrounding areas, such changes may be exploited to detect possible splicing. In many cases, humans can detect such changes and identify splicing effortlessly. However, to develop automatic algorithms to do the same task remains to be extremely difficult. Part of the difficulties comes from the fact that a natural image will consist of complicated edges of arbitrary magnitudes, orientations and curvatures. It is therefore hard to design an edge detector which can robustly distinguish the changes caused by the forgery splicing and the changes that are integral parts of the image signal.

For many years, researchers have been interested in trying to copy biological vision systems, simply because they are so good. The fact that humans can spot splicing in an image with relative ease implies that there must be some underlying mechanisms. Even though exactly how such mechanisms work still remains largely unknown, much research has been done to understand the human visual system (HVS)[15,16,17].

In this work, we exploit some results from research in HVS, especially those in the areas of visual saliency and visual fixation prediction [16,18], to develop a machine algorithms for splicing forgery detection. We present a novel three-level hierarchical system which provides a general framework for detecting splicing forgery. An innovative feature of the new systems is that it employs a visual fixation prediction algorithm to guide feature selection. Based on a very important discovery that there is a high correlation between the splicing boundaries and the first few visual fixation points predicted by a visual fixation prediction algorithm, we select the visual features from the first few fixation points (empirically about 5) to build a classifier to detect splicing. We introduce a normalized edge sharpness measure which is adaptive to variation in edge direction and curvature to provide a robust representation of edges.The experimental results show that the new scheme significantly outperforms existing methods on a public benchmark database. We will also show that an additional advantage of the new method over existing techniques is that it provides a convenient way to locate the splicing boundaries.

The rest of the paper is organized as follows: Section 2 introduces the three-level hierarchical detecting system. Section 3 describes the saliency guided feature extraction process. Section 4 and 5 deal with the feature fusion and localization problem respectively. Experimental results and conclusion are presented in Section 6 and 7.

2 The Proposed Image Splicing Detection System

As illustrated in Fig. 2, the proposed system use a detection window(DW) to scan across locations. At each location, the DW is divide into nine sub-blocks. To spotlight the "unusual" locations in a sub-block, a visual search is performed by regulating a bottom-up *Visual Attention Model*(VAM) with task-relevant top-down information. The bottom-up VAM used here is the Itti-Koch model[16]. It takes an image as input and construct a *Saliency Map* (SM) from low level feature pyramids, such as intensity, edge direction. The visual salient locations, or more formally called as fixations, are identified as the local maximums of the SM. In order to use the edge sharpness cues as top-down regulations to the VAM, , we extract two maps from each sub-block, the *Normalized Sharpness Map* (NSM) and the *Sharpness Conspicuous Map* (SCM). The NSM defines "*what*" is a splicing boundary by scoring the edge sharpness with a normalized value. The SCM is created by modulating the NSM with edge gradient. It implies "*where*" both the edge sharpness and edge gradient are large should be most conspicuous to an inspector and attract more attentions. The SCM is then inspected by the VAM to identify fixations which are the "unusual" spots within the SCM . Discriminative feature vectors are extracted from the most salient k fixations of the NSM to train a hierarchical classifier.

The hierarchical classifier is constituted by three types of meta classifier C_{low}, C_{mid}, C_{high}. C_{low} accept a single feature vector as input and outputs a probability p_{low}. The k p_{low}s from the same sub-block are send as a feature vector to C_{mid}. Analogously, the C_{mid} outputs a p_{mid} for a sub-block. Nine p_{mid}s of the DW are sent to a C_{high} for a final judgment.

3 Feature Extraction

In this section, we discuss how to gather discriminative features to identify if an image is spliced or not.

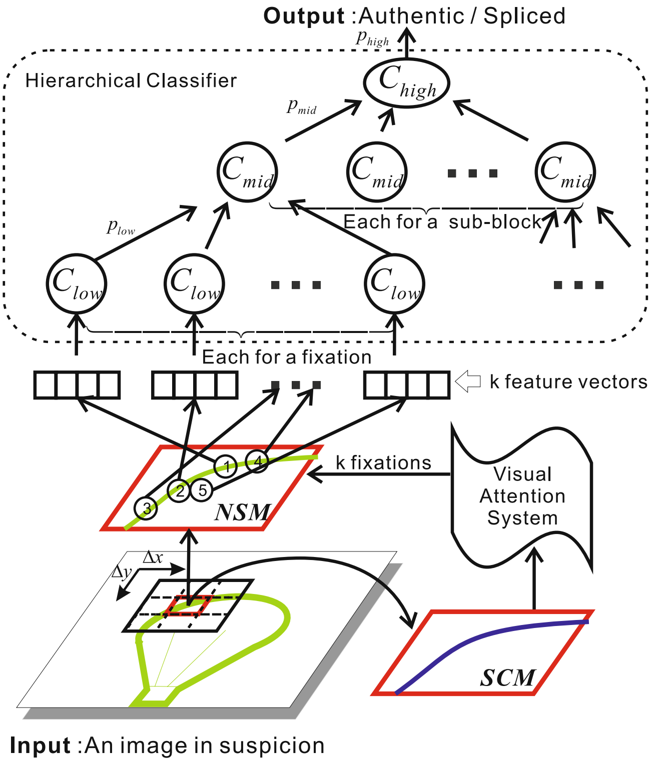


Fig. 2. Proposed system architecture

3.1 Edge Sharpness Measure

The key of detecting splicing forgery is to figure out the sharp splicing edges. The methods mentioned in Section 1 are not specified in analyzing splicing edges and thus have some limitations in practice. The HOS[2] and HHT[3] based detection methods are not good at analyzing small image patches due to the large sample size they often required. Wavelet or linear filtering based methods are limited by their finite filtering directions. When their filtering direction mismatches the splicing boundary which may have arbitrary directions, the filter's response will not correctly reflect the edge's property and thus increases the ambiguity of the spliced and the "natural" boundaries.

To derive a more specific solution which only accounts for the edge sharpness and adaptable to the variations of edge directions, we propose a non-linear filtering approach here based on Order Statistic Filter (OSF). An OSF can be denoted as $OSF(X, n, h \times w)$, where the input X is a 2-D signal and its output is the n'th largest element within a local $h \times w$ window. For example, $OSF(X, 5, 3 \times 3)$ performs a 2-D median filtering.

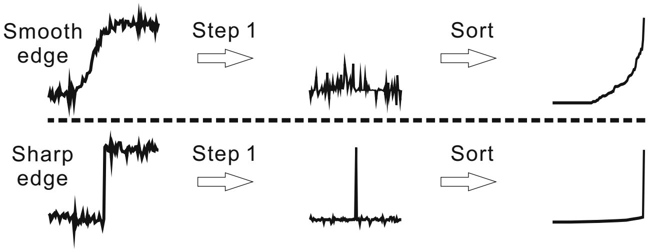
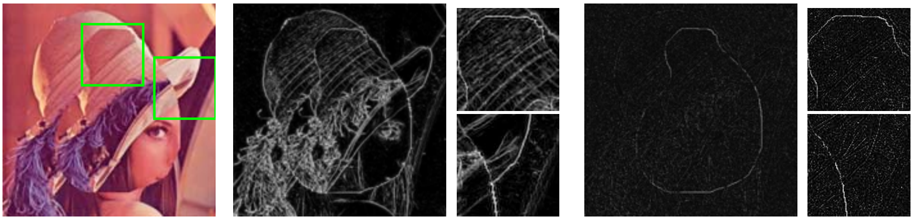


Fig. 3. The changes of sorted output values of sharp edges of step one are steeper than that of smooth edges

Described by Algorithm 1, our method involves three steps: In the first step, the horizontal/vertical abrupt changes are detected as impulse signals with a median filter. Then the output values of step one are sorted as shown in Fig. 3. We can see the sharp step edge has a much steeper peak than a smooth "natural" edge. This steepness provides us a mean to calculate a normalized measure of edge sharpness. Consequently, the Eq.(2) in step two and Eq.(3) in step three suppress those peak values with small local steepness which mostly correspond to the relatively smoother "natural" edges while the steep impulses which correspond to a step edge will be enhanced to be more noticeable. Figure 4 shows the effectiveness of our method.

Algorithm 1. Computing the normalized sharpness map (NSM)

```
Input: Image $I(x, y)$
1) For each color channel $I^c, c \in \{R, G, B\}$ of $I$, detect abrupt change with order filtering in horizontal and vertical direction. Only the horizontal direction is demonstrated.
    • Horizontal denoise: $I_h^c = OSF(I^c, 2, 1 \times 3)$.
    • Horizontal derivation: $D_h^c = \frac{\partial I^c}{\partial x}$.
    • Calculate the absolute difference:
      $E_h^c = abs\left[D_h^c - OSF\left(D_h^c, 3, 1 \times 5\right)\right]$    (1)
2) Combine all color channels $E_h = \sum_c E_h^c$ and the horizontal normalized sharpness is obtained by
      $\bar{E}_h = \frac{E_h}{E_h + OSF(E_h, 2, 1 \times 5) + \varepsilon}$    (2)
   The division here is entry-wise for matrix. $\bar{E}_v$ can be obtained similarly.
3) Combine horizontal and vertical directions $\bar{E} = \max(\bar{E}_h, \bar{E}_v)$ and obtain a final normalized sharpness map with
      $NSM = \frac{\bar{E}}{\bar{E} + OSF\left(\bar{E}, 10, 5 \times 5\right) + \varepsilon}$    (3)
Output: A normalized sharpness map ($NSM$).
```



(a) Spliced lena (b) Phase congruency (c) Proposed NSM

Fig. 4. Edge sharpness of a spliced Lena image given by phase congrency and the proposed NSM. Two regions with different contrast are magnified.

3.2 Visual Saliency Guided Feature Extraction

In the NSM, some edges are more suspicious than others because their sharpness or edge directions are quite abnormal when compared to their surroundings. To identify these locations, we use a VAM proposed by Itti et al [16] which can highlight the most "unusual" spots within an image. We use a publicly available implementation, the Saliency Toolbox [16], and introduce two alterations to the original model.

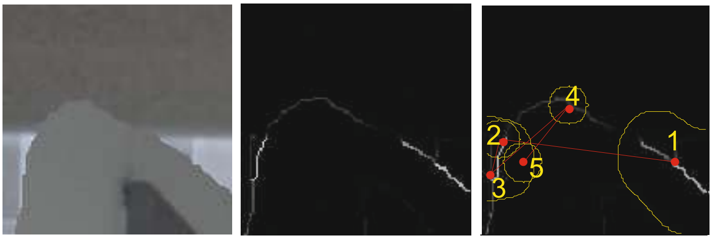
Sharpness Based Saliency. To utilize edge sharpness as a task-relevant information to guide the visual search, a *Sharpness Conspicuous Map*(SCM) rather than the original color image is used as the input of the *SaliencyToolbox*[16]. For more details about how task-relevant information will influence the attention model, please refer to [19]. The SCM is created by modulating the NSM with the edge's gradient magnitude, as follow

$$SCM = NSM \cdot \sum_c \nabla I^c \tag{4}$$

where $\nabla I^c = \sqrt{(\partial I^c/\partial x)^2 + (\partial I^c/\partial y)^2}$. The matrix multiplication here is also entry-wise. In practice, it suppresses the low level feature of specific edges. It means that those edges with high sharpness and also large gradient magnitude are more likely to be a splicing boundary.

With the SCM, the SaliencyToolbox generates an *Saliency Map*(SM) and the fixations are sequentially extracted as the SM's k largest local maximums, as illustrated in Fig. 5. For a spliced image block, most of the fixations will locate on the sharp splicing boundary. While for authentic image blocks, they may fall onto some "natural" sharp edges. The feature vectors extracted from a local patch at these locations will be used to classify the spliced and authentic images.

Localized Saliency. The VAM is applied to fixed-size local image blocks rather than the whole image here. To evaluate the influences of blocksize on the visual



(a) Splicing boundary (b) Generated *SCM* (c) Selected fixations

Fig. 5. Fixations extraction process. (b) is the *SCM* obtained from (a). The red dots in (c) label the location of fixations and the number indicates their order sorted by saliency. The yellow contour line indicates a prohibit region surrounding a fixation which prevent new fixations falling into the same region.

search process, we conduct four tests on spliced images of different size: $64 \times 64, 128 \times 128, 256 \times 256$ and whole image. The whole spliced image is selected from the Columbia image splicing detection dataset described in Sec. 6.1. The fixed-size spliced image blocks are cropped from them. In each tests, 180 spliced images are used to extract the first 10 fixations from their *SCM*s. The resulted fixations of each test are grouped according to their rank, say $k \in [1, 10]$, as a group $\mathcal{G}_k$. The efficiency of the visual searching process relies on the fixation correctly located on the splicing edges. It is quantitatively measured by the *Hit Rate* of $\mathcal{G}_k$ which indicates the co-occurrence of kth fixation with a splicing edge.

$$HR_k = M_k / N_k \tag{5}$$

where N_k is the number of fixations in $\mathcal{G}_k$, M_k is number of fixations which correctly "hit" a splicing edge in $\mathcal{G}_k$. Fig. 6 summarizes the variation of HR_k with fixation rank k and blocksize.

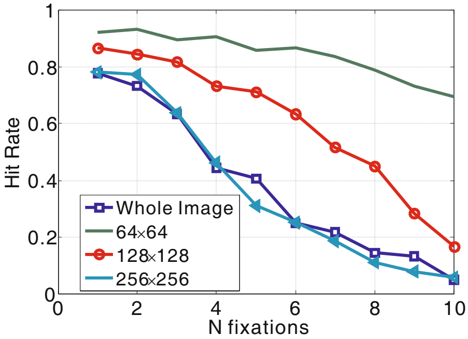


Fig. 6. Relation between the spliced image size and the hit rate of the kth fixation

When the blocks size is large, say 256×256 or whole image, the hit rate curve is lower and also declines steeper than those smaller blocksize. This indicates that a smaller blocksize will result in a higher fixation splicing-boundary co-occurrence probability, meaning that more discriminative features can be extracted from the fixations[1].

However, the blocksize cannot be too small, e.g., below 64x64, because the system will become inefficient with too many tiny blocks to be investigated. We decide to use 128x128 blocksize as a traded off between performance and efficiency.

Extract Feature Vectors at Fixations. At each fixation location, we draw a small local patch (11 × 11) centered at the fixation from the NSM. The histogram of this patch is extracted as a feature vector. It depicts the distribution of sharpness scores in that small local area. Since we extract k fixations for a sub-block and nine sub-blocks within a DW, there are $9 \cdot k$ feature vectors for representing a single DW.

4 Hierarchical Classifier for Feature Fusion

To classify a DW with its $9 \cdot k$ feature vectors, we design a hierarchical classifier [20] with a tree architecture as illustrated in Fig. 2. The top level classifier C_{high} has nine C_{mid} descendants each in charge of classifying a sub-block. Every C_{mid} has k C_{low} descendants each in charge of classifying a feature vector extracted at a fixation within the sub-block. The high level final decision is obtained by fusing the outputs of lower levels.

The use of this classifier structure has two benefits. Firstly the three meta classifiers are of low dimensional input which makes it easier to collect enough training samples to train the whole classifier. Secondly, by considering multiple sub-blocks together, the high level classifier C_{high} can make use of some of the structural information of neighboring sub-blocks to provide a context for making the decision.

we use LIBSVM [21] to implement support vector machines (SVMs) with a radial basis function(RBF) kernel as the meta classifiers. Each of them outputs a probability ranged in $[0, 1]$ rather than just a binary decision to its predecessor. Detailed training setups are presented in Section 6.2.

5 Localization

Different from an edge detector, the aim of this splicing detection system is not to find out the whole splicing boundary. We just label some of the most suspicious edges patches to indicate splicing. This is especially useful when further region based analysis or human inspection are needed. With the above hierarchical structure, localizing the splicing boundary is straight forward. For each

[1] Further analysis will be given in Section 6.3.

DW classified as spliced, we start a width-first-search from the root of tree. According to the output probability p_{mid}, p_{low} estimated for the C_{mid} and C_{low}, the sub-blocks and fixations which are classified as spliced are marked up. In the experimental results shown in Fig. 8, we give a concrete implementation by combining a more delicate sub-block search within each DW. It only labels the most suspicious DW of an image. The sub-blocks and fixations within this DW are treated with the same principle. These indications can be further utilized to extract an integral segment of splicing edge but not implemented this work.

6 Experimental Results

6.1 Image Database

We test our proposed algorithm on the Columbia image splicing detection dataset [7] including 180 authentic(auth) and 183 spliced (splc) TIFF images of different indoor/outdoor scenes. The spliced images are created using Photoshop software with no concealment. The database also provides an edgemask for each image. For a splc image, it labels the splicing boundary. For an auth image, it marks some most signification edges obtained from a segmentation algorithm.

6.2 Training

We split the database into two halves, and use one half for training (90 auth/90 splc) and the other for testing. The three meta classifier are trained with image patches drawn from the training set. According to the hierarchical structure, a low level meta classifier must be trained before its higher level predecessor. The feature extraction methods for each meta classifier is listed as follows:

The C_{low} is trained to classify 11×11 patches. By using the edgemasks, we randomly draw some 11x11 patches that locate on "natural"/splicing edges. A 10 dimensional local histogram obtained from the NSM of a patch is used as feature vector for C_{low}, as mentioned in Section 3.1.

The C_{mid} is trained to classify 128×128 sub-blocks. The 128×128 training and testing image are similarly selected using the edgemasks. For each sub-block, the first k fixations are extracted from its SCM using the SaliencyToolbox[16]. Then one 11×11 patch is drawn at each of these fixation locations and turned into a feature vector with the feature extraction procedure for patches. The trained C_{low} will turn the k feature vectors into k p_{low}s for training the C_{mid}.

The C_{high} is trained to handle 384×384 DWs. The selection strategy of image blocks is the same as C_{mid} except that the edge length contained in the inner ring of a DW, say $[33 : 352, 33 : 352]$, should includes at least 128 pixels. We divide the DW into nine 128×128 sub-blocks. After turning them into nine p_{mid} with the feature extraction procedure for sub-blocks and a trained C_{mid}, we packed the nine p_{mid}s up as a feature vector for training C_{high}.

At first, all the patches are randomly selected as the *Basic Set*. Then we adopt a bootstrapping strategy which intensionally adds some near-spliced auth samples into the basic training set and re-train the classifiers. Table 1 summarizes the detailed training setups.

Table 1. Training setups for meta classifiers. k is the number of fixations extracted in a 128×128 sub-block.

	C_{low}	C_{mid}	C_{high}
Image size	11×11	128×128	384×384
Feature dimension	10	k	9
Basic set (auth/splc)	1800/1800	1800/1800	540/540
Bootstrap set (auth)	NA	794	105

6.3 Classification Performance

Determining the Fixation Number *k*. There are two major parameters influence the classification performance of our system: the blocksize determined in Section 3.2 and the fixation number k extracted from each sub-block. We determine the rank of k by increasing k until the classification accuracy of C_{mid} stops growing. As shown in Fig. 7 , C_{mid}'s accuracy stop climbing after the first 5-6 fixations have been selected. Another curve is the classification accuracy when only the fixations ranked k of each block are used to train C_{mid}. Its steep declining from 6 evidents that the lower ranking fixations(7-10) are unreliable features for classification not only because they are not so salient in the sense of sharpness but also for their low co-occurrence with splicing boundaries as shown in Fig. 6. Thus they provide little extra performance gain when the first few fixations are already . An important insight given by this experiment is that the first few fixations contain the most discriminative information for detecting splicing boundaries. Consequently, we experimentally determine the number of selected fixation as $k = 5$.

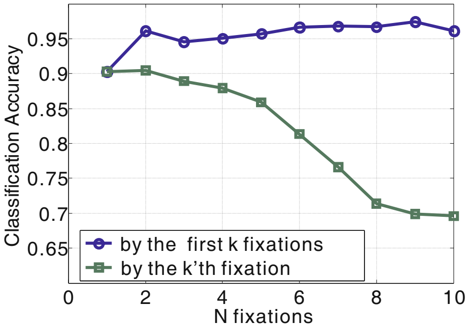


Fig. 7. Relation of fixation number and classification accuracy

Performance of Meta Classifiers. Table 2 shows the performance of the meta classifiers. The C_{low},C_{mid},C_{high} are trained and tested with the Basic set. Their accuracy shows that the feature extracted for each meta classifier is discriminative and induce reasonably good results. Note that performance of C_{high} also represents the overall performance of hierarchical classifier C_{hie}. For detecting splicing forgery, we should not just concern about classification accuracy but also the false positive rate. Because a splicing detection system shall take the whole images as a fake even if only a small part of it is classifier as spliced when the DW scanning across it. A high false positive rate classifier will hardly had any practical use. So we re-trained C_{mid} and C_{high} by adding a bootstrap set into the training set of Basic set as mentioned above and keep the testing set unchanged. This can further bring down the detection false positive rate to about 1.11%.

For a comparison with non-saliency based feature extraction methods, we also implemented the detection method proposed by Chen et. al[5] with wavelet and phase congruency based features. Since the original method was only for classifying 128 × 128 grayscale image blocks, we extend it to color image by putting the feature vectors obtained from each color channel together. And they did not provide a scheme to integrate higher level information to handle large size image, the performance is only compared in sub-block level and without bootstrap(C_{mid} without re-training). As shown in Table 2, the accuracy and true positive rate of the proposed method is higher and the false positive rate is lower than the reference method.

Table 2. Performance of meta classifiers. The performance of C_{mid} is compared to Chen's method in [5] as indicated by bold font.

	C_{low}	C_{mid}	C_{mid} re-trained	C_{high}	C_{high} re-trained	Reference method
Accuracy(%)	95.22	**96.33**	92.83	94.07	91.30	**89.39**
True positive rate(%)	93.67	**95.22**	86.78	92.22	83.7	**90.44**
False positive rate(%)	3.22	**2.56**	1.11	4.07	1.11	**11.67**

6.4 Splicing Detection and Localization

In detecting splicing forgeries in a whole image, the detector need to be scanned across locations. This is done by shifting the detecting window by $(\Delta x, \Delta y) = (64, 64)$. The step size will affect the detection accuracy and speed.

Fig. 8 shows the detection results of the system. With a total of 180 testing images, the 90 splc image are all correctly classified and only 7 auth image misclassified as splc. As shown in subfigure (j-l), the errors are mostly caused by the straight edges of tiny gaps, iron fences and window frames in a distance and with a high contrast.

The average hit rate of the fixations marked with yellow is about 84.6%. As shown in subfigure (a-f), most of the fixations fall onto a visually conspicuous segment of a splicing edge which suggests that the localization performance of the proposed system is reasonably good and is to some extent consistent with human's visual perception.

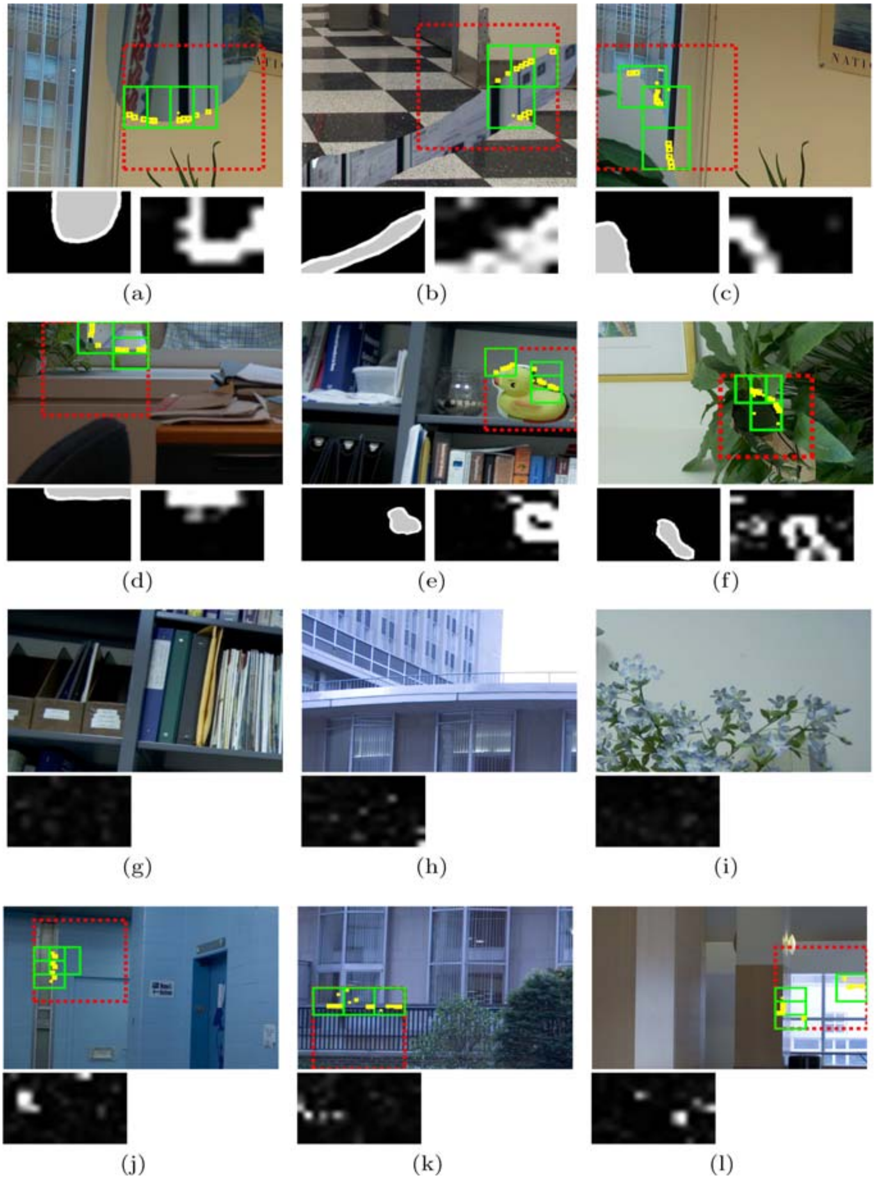


Fig. 8. Real Image Test. (a-f) and (g-i) are the results of spliced and authentic image respectively. (j-l) gives some examples of mis-classified images. For each image, a block-wisely evaluated p_{mid} map generated by the C_{mid} is given in the lower left of each subfigure. And for spliced images, the ground truth edgemask is given on the lower right conner. The red dash-line rectangle marks a DW with the largest p_{high} over the other DWs in the same image. The three green solid-line rectangles marks three sub-blocks with the three largest p_{mid}s in the DW. The tiny yellow dots mark the fixations within the sub-blocks and encode the p_{low} with its size.

6.5 Subjective Test

A natural question one may raise would be if the proposed HVS based splicing detection system is comparable to a human. We give some preliminary subjective testing results conducted on three male and three female testers to show how well can human performs on the same task. We randomly select 360(180 auth/180 splc) edge patches from the C_{mid}'s training and testing set[2]. Each of the 128 × 128-pixel image blocks is displayed on the center of a monitor without rescaling. None of the testers have ever seen them before. The testers are required to make judgments as fast as they can by clicking buttons. We also record the time they take to judge every block.

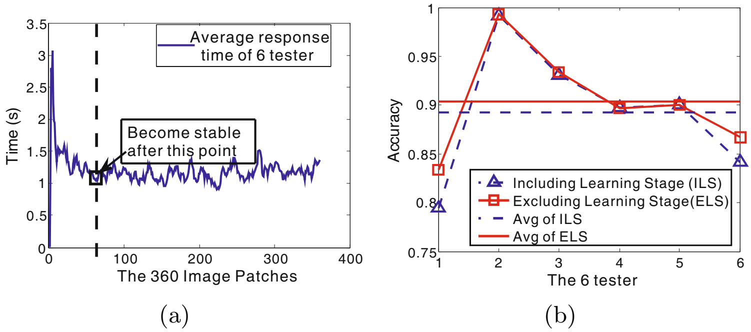


Fig. 9. Subjective testing results. (a) shows the average response time of six testers. The dash line indicates an observed learning stage in which the response time keeps on decreasing. (b) shows the individual and average accuracy of the six testers.

It is observed from Fig. 9(a) that there is a learning stage(about 60 images) before a tester gets familiar to use the system and their judgment time becomes stable (about 1 second per block) after that stage. From Fig. 9(b), we can see that the average performance of a human is about 90% which is lower than the proposed system. Only one of six testers outperforms our system with an accuracy of 99.43% (learning stage excluded). Due to a small number of testers, these observations remain to be further investigated.

7 Discussions and Conclusions

In this work, we described a fully automatic system for detecting digital image splicing forgeries based on the sharp splicing boundaries. The novelty of the proposed system includes the introduction of an OSF based edge sharpness

[2] The subjective test is confined to small block size to avoid the influence of image content information.

measure, a visual saliency guided feature extraction mechanism and also a hierarchical classifier into the solution of splicing detection problem. We show that a reliable hierarchical classifier can be trained with the discriminative features extracted from the first few fixations predicted with a visual attention model with edge sharpness as visual cues. The hierarchical classifier also provides a convenient way for localizing splicing boundaries. Experimental results based on a publicly available image database show that the proposed system achieves reasonably good performance and outperforms previous techniques reported in the literature.

A limitation of the proposed system is that the edge sharpness cues currently used will fail when concealing measures, such as blur, is applied. This may be improved by incorporating new blur-resistant feature as visual cues, such as CRF based method[7] or abnormality of local hue[8], into the framework to identify a blurred splicing boundary from surrounding "natural" boundaries. Another possible improvement is how to more effectively use edge structure information, such as fixation position or edge connectivity/curvature. A more sophisticate high level mechanism may help to further improve the system performance.

Acknowledgment

This work is supported by NSFC (60633030), 973 Program (2006CB303104). The authors would like to thank Dr. Yun-Qin Shi and Dr. Wei-Shi Zheng for helpful discussions and the anonymous reviewers for their useful suggestions.

References

1. Farid, H.: Detecting digital forgeries using bispectral analysis. Technical Report AIM-1657, AI Lab, MIT (1999)
2. Ng, T.T., Chang, S.F., Sun, Q.B.: Blind detection of photomontage using higher order statistics. In: ISCAS, vol. 5, pp. V688–V691 (2004)
3. Fu, D.D., Shi, Y.Q., Su, W.: Detection of image splicing based on hilbert-huang transform and moments of characteristic functions with wavelet decomposition. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 177–187. Springer, Heidelberg (2006)
4. Sutcu, Y., Coskun, B., Sencar, H.T., Memon, N.: Tamper detection based on regularity of wavelet transform coefficients. In: Proc. of IEEE ICIP 2007, vol. 1-7, pp. 397–400 (2007)
5. Chen, W., Shi, Y.Q., Su, W.: Image splicing detection using 2-d phase congruency and statistical moments of characteristic function. In: Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505 (2007), 65050R
6. Lin, Z.C., Wang, R.R., Tang, X.O., Shum, H.Y.: Detecting doctored images using camera response normality and consistency. In: Proc. of IEEE CVPR 2005, vol. 1, pp. 1087–1092 (2005)
7. Hsu, Y.F., Chang, S.F.: Image splicing detection using camera response function consistency and automatic segmentation. In: Proc. of IEEE ICME 2007, pp. 28–31 (2007)

8. Wang, B., Sun, L.L., Kong, X.W., You, X.G.: Image forensics technology using abnormity of local hue for blur detection. Acta Electronica Sinica 34, 2451–2454 (2006)
9. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. IEEE Trans. IFS 2(3), 450–461 (2007)
10. Johnson, M.K., Farid, H.: Exposing digital forgeries through chromatic aberration. In: ACM MM and Sec 2006, vol. 2006, pp. 48–55 (2006)
11. Chen, M., Fridrich, J., Lukáš, J., Goljan, M.: Imaging sensor noise as digital X-ray for revealing forgeries. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 342–358. Springer, Heidelberg (2008)
12. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Trans. SP 53(10 II), 3948–3959 (2005)
13. Swaminathan, A., Wu, M., Liu, K.J.R.: Optimization of input pattern for semi non-intrusive component forensics of digital cameras. In: IEEE ICASSP, vol. 2, p. 8 (2007)
14. He, J.F., Lin, Z.C., Wang, L.F., Tang, X.O.: Detecting doctored jpeg images via dct coefficient analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 423–435. Springer, Heidelberg (2006)
15. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE TPAMI 29(3), 411–426 (2007)
16. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. PAMI 20(11), 1254–1259 (1998)
17. Rybak, I.A., Gusakova, V.I., Golovan, A.V., Podladchikova, L.N., Shevtsova, N.A.: A model of attention-guided visual perception and recognition. Vision Research 38(15-16), 2387–2400 (1998)
18. Hou, X.D., Zhang, L.Q.: Saliency detection: A spectral residual approach. In: Proc. of IEEE CVPR 2007, pp. 1–8 (2007)
19. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision Research 45(2), 205–231 (2005)
20. Heisele, B., Serre, T., Pontil, M., Poggio, T.: Component-based face detection. In: IEEE CVPR 2001, vol. 1, pp. I657–I662 (2001)
21. Chang, C.C., Lin, C.J.: LIBSVM:a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`