Exposure Of Presence of Splicing Under Space-Varient Blurring Conditions

G.DURGAVENI
II Yr M.Tech Information Technology
Francis Xavier Engg college
Tirunelveli

Miss.J.MONICA ESTHER
Asst.Prof,Information Technology
Francis Xavier Engg college
Tirunelveli

***Abstract*— In this work ,we propose a new methodology to expose the presence of image splicing .Image splicing is a form of tampering in which an original image is altered by copying a portion from a different source.Our proposed work seems to have more imporatnce because of the lot and lot of existing availabilities of sophisticated image editing tools. So many number of methods exist in the present scenario for dealing with this same problem.But all of them are working only in the presence of uniform space invariantblur situations only.In our proposed approach too,we detect image splicing by means of blur as a cue.In contrast to all of the existing methods,we can expose the presence of splicing by evaluating inconsistencies in motion blur even under space_varient blurring situations With the no of few case studies for different scene situations and camera motions,we validate our proposed methodology.**

***Index Terms*— Image splicing, camera shake, motion blur, transformation spread function, scaling.**

I. INTRODUCTION

IVEN the potential that visual media has in influencing society, it is alarming to note that creation of fake images has become an effortless task with the wide availability of sophisticated softwares such as GIMP, Photoshop, Paint and the like. Such fake images are used for pushing political propaganda, sensationalizing news, and for even evidence tampering. Existing forgery detection techniques can be categorized as active or passive In the active approach, apriori information (watermark) is inserted into an image. However, such active embedding is also a clear limitation of these methods. In fact, there exist tons of images without any such prior information. Thus, it is not surprising that passive forgery detection methods have caught the attention of researchers.

Image splicing and cloning are two widely followed approaches for generating fake images. In this paper, we focus on the problem of detecting whether an image is spliced or not and also demarcate the spliced region, if the image is spliced. Image splicing involves more than one image, where a part of one image is copied and pasted onto another. An example of splicing is shown in Fig. 1. The people in the bottom-left part (marked in red) were inserted into the scene but this is difficult to discern by mere visual inspection of the image. There exist many works for detecting image splicing. Some of the commonly exploited cues for detecting splicing include stastistical properties of images, physical prop-erties such as lighting inconsistencies,

Fig. 1. Example of a spliced image.

image device characterictics such as the Camera Response Function (CRF) and sensor noise . Interestingly, inconsistency in image blurring can also act as a cue for splicing detection. The blur itself can be due to optical defocus or relative motion between camera and scene . In fact, Fig. 1 is one such example in which blur can be exploited to expose splicing.

A. Related Works

We have proposed methods based on the covariance structure of an image signal. In many situations, to arrive at a spliced image, one needs to apply geometric transformations such as scaling and rotation on the spliced region. These operations provide traces of resampling which can be used to detect the presence of splicing,proposed a method using lighting as a cue. The spliced region and background typically do not share similar lighting conditions. Hence, the inconsistency in estimated light source directions is a valuable cue for detecting splicing. However, finding the light source directions in a complex lighting environment (consisting of multiple light sources) from a single image is a challenging task in itself. The amount of light reflecting from a surface point is proportional to the angle between the surface normal at that point and the direction of light source. In the authors describe spherical parameter modeling of a complex lighting environment. They consider 2-D surface normals at occluding boundaries to estimate the parameters of the model. In a method to estimate 3-D surface normals for images containing persons is discussed. The parameters of the lighting environment are estimated by fitting 3-D models to persons heads.

We proposed a method based on the CRF which is a nonlinear function that maps input irradiance to image intensity. The CRF acts like a fingerprint for a camera. If the spliced region and background are from different cameras, there will exist inconsistencies in the CRFs of these two regions which can be measured using a cross-fitting framework .In CRF estimation from a single image is described. This method will fail if the spliced region and background happen to originate from the same camera.

Blur in images is normally considered a nuisance and it is common practice to devise methods to remove its effect. However, blur has its advantages too. It has been shown in that camera motion and depth map of a 3D scene can be inferred from motion blur. For a real-aperture camera model, the depth of the scene is related to the degree of optical defocus blur. utilized optical blur as a cue for splicing detection. Consider two objects in the image at similar depths; if the difference in the induced defocus blur is greater than a predefined threshold value, then it can be deduced that the image is spliced. Not with standing the importance of defocus blur, the relevance of motion blur is also growing by the day due to increasing ubiquity of hand-held imaging devices. This is also evident from the spate of recent works in motion deblurring . While the optical blur is space-variant only for 3D scenes, motion blur is more challenging as it can be space-variant even for flat scenes.

Interestingly, motion blur can also serve as a cue to detect the presence of splicing. Due to the random nature of incidental camera shake, it is unlikely that the blur introduced in the spliced region will be consistent with the rest of the image. Recently, have proposed a method based on the discrepancies in motion blur for splicing detection in static scenes by assuming uniform velocity for the camera.
For example, in the case of horizontal camera motion, the blur kernel will be of the form $k = \frac{1}{L}[1, 1, 1, \ldots L \text{ times}]$, where L is the length of the blur kernel. They characterize the blur kernel by a 2D vector with length and angle (direction of the kernel) as elements. They use spectral characteristics of image gradients to estimate blur. The given input image is divided into a number of overlapping blocks and corresponding blur kernels are estimated. These blur kernels are interpolated for every pixel in the image. Finally, the image is segmented into two regions depending on the inconsistency in the length and direction of the blur kernels. One of the segmented regions will possibly be the spliced region. It should be noted that the inconsistency criterion defined in [16] will work only for space-invariant blur arising from uniform camera motion.

B. Outline of Proposed Technique

In this paper, we propose an automatic passive splicing detection method based on motion blur. Following other works [14]–[16] we also assume that the scene is static and that the blur induced in the image is due to camera motion. This is a very relevant problem since camera shake is a common occurrence [25]–[31]. This is challenging too since an object can be inserted into a static scene with ease but is all the more difficult to detect especially in the presence of space-variant blur. An elementary version of our work can be found . The main advantage of our method over existing schemes is that it is applicable even to non-uniform blurring scenarios. Moreover, the blur itself can be of *arbitrary* shape. The actual induced blur depends both on camera motion and scene characteristics. Note that space-variant blur can be caused by simple rotational motion of the camera (even for a fronto-parallel scene). For scenes with varying depths, the motion blur induced will always be space-variant. Optical defocus blur, if present, is assumed to be negligible. We investigate three interesting and commonly prevalent situations that emerge due to the nature of the scene and type of camera motion. The three situations are different, arise naturally and have unique characteristics. We first consider a fronto-parallel scene and in-plane camera translation motion which can be characterized using the notion of the point spread function (PSF). The second is that of a fronto-parallel scene but with in-plane translations as well as rotations of the camera. Due to the space-varying nature of PSF, we show that several PSFs are needed to handle this situation. The third scenario involves a 3D scene and in-plane translation motion of the camera; the PSF is again space-varying but as a function of scene depth. The goal is to provide quantitative evidence based on motion blur to conclusively establish the act of splicing in each of these scenarios. Although other situations are also possible, such as a 3D scene with general camera motion, we limit the scope of our work to the above three scenarios.

In the first case, given that the scene is fronto-parallel, pure translational motion of the camera will displace all the pixels in the image by the same amount. Hence, the blur kernels though arbitrarily-shaped, should ideally be the same anywhere in the image. If a part of this image is replaced with a region from a different image, it is very unlikely that the blur kernel in that region will match with the rest of the image. This inconsistency can be used as a check for splicing. In fact, we can directly compare the blur kernels using cross-correlation as a measure of similarity.

Next, we again consider a flat scene, but allow for in-plane translations as well as rotations of the camera. Since this case involves space-variant blurring, several spatially separated PSFs picked randomly from the image enable description of camera motion. The blurred image can then be represented as a weighted average of warped versions of the original unknown image. The weights themselves depend on the amount of time the camera spent at a particular camera pose or homography. Even though the number of homographies can be large (as it depends upon the limits of possible transformations and the resolution of the motion parameters), the blurred image resulting from camera shake undergoes very few transformations from the possible set of allowable homographies. For example, if the allowable camera translations in the X and Y-directions is in the range $-5{:}1{:}5$ pixels and the rotations about Z-axis are in the range $-2{:}0.25{:}2$ degrees, then the total number of possible homographies will be $11 \times 11 \times 17$. However, since camera motion trajectory is quite sparse, the captured blurred image will have been caused only by a small subset of the above. This motivates us to enforce a sparsity constraint on the motion path of the camera.

We estimate the weights of these homographies also referred to as Transformation Spread Function (TSF) from a single blurred image. Interestingly, there exists a linear relationship between TSF (which is global) and the blur kernel (which is local). If we know the TSF, it can be used to describe the blur kernel at any location in the image. The inverse problem of determining the TSF given the blur kernels at various locations in the image is more involved. Since we consider a fronto-parallel scene, due to camera motion, every point in the scene should ideally be influenced by the same transformation. i.e., the same TSF should be valid for the entire image. In conjunction with the sparsity constraint, we show that the TSF can be recovered from the PSFs. At any location, the observed PSF is compared with the PSF predicted by the estimated TSF to reveal splicing.

In the last case, we consider a depth-varying scene and translational camera motion. A 3D scene will induce space-variant blurring wherein the blur kernels are a function of depth. In fact, the PSFs at different depths turn out to be scaled versions of each other, the scaling factor being the relative depth of the two scene points. In this case, we propose the use of Fourier-Mellin Transform (FMT) [33] to compare PSFs. Note that we do not assume prior knowledge of the depth map of the scene, yet we are able to compare the PSFs using the underlying scaling relationship that relates two PSFs at different depths.

The previous discussion assumes that we know the specific scenario that we are dealing with and where the authentic blur kernels come from. In practice, we do not have knowledge of either. We propose a strategy to automate this. Given a test image, our method first determines whether the image is representative of case (i), case (ii) or case (iii). It is based on the reasonable premise that the spliced region typically occupies a fraction of the total image area. Subsequently, we propose a scheme to automatically demarcate the spliced region. It is the blur kernel that serves as a cue in the inference process.

The paper advances the state-of-the art substantively and in several ways including automatic segmentation of blur kernels, computation of camera motion, automatic detection and demarcation of the spliced region, handling camera translations with and without depth in the scenes, as well as general camera motion for flat scenes. Knowledge of neither the suspect region nor the authentic area is assumed. We demonstrate the effectiveness of our method on several examples and also provide comparisons with existing methods.

The organization of the paper is as follows. In section II, we describe the motion blur model. In section III, we discuss different blurring scenarios depending on the type of camera motion and scene structure. In section IV, we propose suitable inconsistency criteria for each of these scenarios. We validate our approach on several synthetic and real examples in section V, and conclude with section VI.

II. Motion Blur

It is not uncommon for hand-held camera users to experience motion blur resulting from incidental camera shake. The blur in the image is due to the slow shutter speed of the camera relative to camera motion. Possible solutions to prevent the occurrence of blur include (i) increasing the shutter speed of the camera, and (ii) resorting to tripods. However, reduced shutter time will typically result in a noisy image while a tripod restricts mobility. In fact, in places such as heritage sites or museums, tripods are not even permitted .

During image capture, what we witness is an averaging of light intensities at the camera sensor. Due to camera motion, different scene points contribute to a pixel during exposure resulting in blur in the captured image. The degree of blur at a point in the image depends on the spatial location of the point, structure of the scene, and relative motion between scene and camera. Many works exist whose goal is to recover the latent image from one or more blurred images. These deblurring algorithms are roughly classified as blind and non-blind. The input to blind methods is either single or multiple blurred images. Non-blind methods have, in addition, prior knowledge about the blur kernel. Note that for the task of splicing detection, only a single observation is available and there is no prior knowledge of the blur kernel. Also our goal is not deblurring but verifying the authenticity of an image.

As discussed earlier, existing splicing detection methods model the blurred image as a convolution of the latent image with a uniform blur kernel i.e., the blur kernel is assumed to be flat and space-invariant. However, in reality, space-variant blurring is quite common and the blur can be quite arbitrary shape. For example, even in the case of a flat scene, simple camera rotation will result in non-uniform blurring. We now analyze this aspect in more detail.

Assume the origin to be the camera center and let $P = [X \;\; Y \;\; Z]^T$ be a 3D point in the static scene. The corresponding projected point in the image plane is given by homogeneous co-ordinates $\mathbf{x} = [x \;\; y \;\; 1]^T$ where $x = qX/Z, y = qY/Z$ and q is focal length of the camera. This relation can be represented in matrix form as a linear relationship $\mathbf{x} = KP$ where $K = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{bmatrix}$. At time τ (during exposure), the 3D point P gets transformed to $P_\tau = R_\tau P + T_\tau$ due to camera motion. Here, R_τ represents the rotation matrix at time instant τ and is a combination of the rotational matrices about the X, Y and Z axes. For example, the rotation matrix about Z-axis is given by $R_{Z_\tau} = \begin{bmatrix} \cos\theta_{Z_\tau} & -\sin\theta_{Z_\tau} & 0 \\ \sin\theta_{Z_\tau} & \cos\theta_{Z_\tau} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ where θ_{Z_τ} denotes the angle of rotation about the Z-axis. The vector $T_\tau = [T_{X_\tau} \;\; T_{Y_\tau} \;\; T_{Z_\tau}]^T$ represents translations along X, Y and Z axes, respectively, at time instant τ.

Let the projection of P_τ in the image plane be denoted by $\mathbf{x}_\tau = [x_\tau \;\; y_\tau \;\; 1]^T$. Therefore $\mathbf{x}_\tau = KP_\tau$. Let us assume that point P lies on a plane with normal N and at a distance d from the camera center. Therefore, any point Q lying on this plane will satisfy the relation $N^T Q = d$. We now derive the relation between P_τ and P using this planar constraint as

$$\begin{aligned} P_\tau &= R_\tau P + T_\tau \\ &= R_\tau P + T_\tau \frac{N^T P}{d} \end{aligned}$$

This yields the relation between $\mathbf{x}_\tau$ and $\mathbf{x}$ in the image plane as $\mathbf{x}_\tau = K \left(R_\tau + \frac{T_\tau}{d} \right) K^{-1}\mathbf{x}$ since $P = K^{-1}\mathbf{x}$. For a fronto-parallel scene where all the scene points are at the same distance d from the camera, the image points before and after camera motion (at time τ) are then related through homography H_τ as $\mathbf{x}_\tau = H_\tau \mathbf{x}$ where

$$H_\tau = K \left(R_\tau + \frac{1}{d} T_\tau [0\ 0\ 1] \right) K^{-1}$$

Let f be the image captured without camera shake. The transformed image due to camera motion at time instant τ will then be given by $g_\tau(H_\tau(\mathbf{x})) = f(\mathbf{x})$. Therefore, the blurred image can be modeled as the average of the transformed versions of f during the exposure time T_e. The blurred image intensity at a location $\mathbf{x}$ can then be expressed as

$$g(\mathbf{x}) = \frac{1}{T_e} \int_0 f(H_\tau^{-1}(\mathbf{x}))\, d\tau$$

The loss of order of transformation in the blurred image is of little consequence to the problem on hand.

There is an alternative way to express $g(\mathbf{x})$. Let D be the set of all possible transformations due to camera motion. This will be a 6D (translations and rotations about the three axes) space in general. Let $h_D : D \rightarrow R^+$ denote the transformation spread function (TSF) that maps elements in D to non-negative real numbers. The TSF $h_D(\lambda)$ represents the fraction of the exposure time T_e the camera spent at a particular transformation λ $(\lambda \in D)$. Hence $\int_{\lambda \in D} h_D(\lambda) = 1$. The blurred image can then be equivalently modeled as a weighted sum of the warped versions of f i.e.,

$$g(\mathbf{x}) = \int_{\lambda \in D} h_D(\lambda) f(H_\lambda^{-1}(\mathbf{x}))\, d\lambda \qquad (1)$$

Incidentally, the blurred image can be related to the latent image f through a space-variant blur kernel h as

$$g(\mathbf{x}) = \int f(\mathbf{x} - \mathbf{u}) h(\mathbf{x} - \mathbf{u}, \mathbf{u})\, d\mathbf{u} \qquad (2)$$

where $h(\mathbf{x}, \mathbf{u})$ denotes the PSF at the image point $\mathbf{x}$ and $\mathbf{u}$ is an independent variable. The PSF $h(\mathbf{x}, \mathbf{u})$ represents the amount of displacement undergone by image point $\mathbf{x}$ due to the homographies, weighted by the fraction of time spent at each homography during exposure [26] i.e.,

$$h(\mathbf{x}, \mathbf{u}) = \frac{1}{T^e} \int_0^{T_e} \delta(\mathbf{u} - \bar{\mathbf{x}}_\tau) d\tau \qquad (3)$$

Here, $\bar{\mathbf{x}}_\tau$ indicates displacement of point light source at $\mathbf{x}$ due to transformation H_τ while δ is the 2D dirac delta.

Importantly, we can write the PSF in terms of the TSF as

$$h(\mathbf{x}, \mathbf{u}) = \int_{\lambda \in D} h_D(\lambda) \delta(\mathbf{u} - (H_\lambda(\mathbf{x}) - \mathbf{x}))\, d\lambda \qquad (4)$$

The corresponding discrete form then becomes

$$h(i, j; m, n) = \sum_{\lambda \in D} h_D(\lambda) \delta_d(m - (i_\lambda - i),\ n - (j_\lambda - j)) \qquad (5)$$

where (i_λ, j_λ) denotes the coordinates of the point when a transformation H_λ is applied on a point $\mathbf{p} = [i\ \ j]^T$ and δ_d denotes the 2D Kronecker delta.

III. Scene, Motion and Blur

Although general camera motion consists of six degrees of freedom, argue that the cause of blur in an image is predominantly due to camera rotations. have shown that in practical scenarios, general (6D) camera motion can be reasonably approximated by 3D (in-plane translations and rotations). According to them, even small out-of-plane rotations can be modeled as in-plane translations. Following we too model camera motion to consist of in-plane translations and rotations. Thus, D represents a 3D vector space in our formulation. We now discuss three interesting cases of motion blur that stem from specific situations of camera motion and scene structure.

A. Planar Scene (a Scene With Insignificant Depth Variations)

- In-plane translation: In this case, the camera is translated on a plane parallel to the image plane. Hence, all the points in the image plane will be displaced by the same amount which will result in space-invariant blurring. Hence, the blurred image can be expressed as a convolution of the latent image with the blur kernel. The homography relation in the image plane will then be

$$\mathbf{x}_\tau = K \left(R_\tau + T_\tau \frac{N^T}{d} \right) K^{-1}\mathbf{x} \qquad (6)$$

  where $R_\tau = I$, $N^T = [0\ 0\ 1]$ and $T_\tau = [T_{X_\tau}\ \ T_{Y_\tau}\ \ 0]^T$ and d is the same for all the points. Hence, we get

$$\mathbf{x}_\tau = K \left(I + \frac{T_\tau}{d} [0\ 0\ 1] \right) K^{-1}\mathbf{x} \qquad (7)$$

  which further simplifies to $x_\tau = x + \frac{qT_{X_\tau}}{d}$ and $y_\tau = y + \frac{qT_{Y_\tau}}{d}$.
- In-plane translational and rotational motion: If the camera is rotated about the Z-axis, the points in the image plane do not inherit the same displacement. The image points near the axis of rotation have less displacement as compared to points that are away from the axis of rotation and this results in non-uniform (i.e., space-variant) blur. In addition, when there is in-plane translation, the homography relation in the image plane for a fronto-parallel scene becomes $\mathbf{x}_\tau = K \left(R_\tau + \frac{T_\tau}{d} [0\ 0\ 1] \right) K^{-1}\mathbf{x}$ where $R_\tau = R_{Z_\tau}$ and $T_\tau = [T_{X_\tau}\ \ T_{Y_\tau}\ \ 0]^T$.

B. General 3D Scene

- In-plane translational motion: In the case of a 3D scene, even when the camera motion is only in-plane translations, the displacement of pixels in the image plane will vary as a function of the depth of the corresponding point in the 3D scene. Due to parallax, points near to the camera will incur a large displacement in the image plane compared to points that are farther-off from the camera. This in turn results in non-uniform blurring.

The homography relation for points lying on a fronto-parallel plane at distance d_0 is by

$$H_\tau = K \begin{bmatrix} 1 & 0 & \frac{T_{X_\tau}}{d_0} \\ 0 & 1 & \frac{T_{Y_\tau}}{d_0} \\ 0 & 0 & 1 \end{bmatrix} K^{-1} \quad (9)$$

Correspondingly, the PSF at a location **x** with respect to reference depth d_0 is given by

$$h_0(\mathbf{x}, \mathbf{u}) = \frac{1}{T_e} \int_0^{T_e} \delta(\mathbf{u} - (H_\tau(\mathbf{x}) - \mathbf{x}))\, d\tau \quad (10)$$

where $x_\tau = x + \frac{qT_{X_\tau}}{d_0}$ and $y_\tau = x + \frac{qT_{Y_\tau}}{d_0}$. Thus, we get

$$h_0(\mathbf{x}, \mathbf{u}) = \frac{1}{T_e} \int_0^{T_e} \delta\left(u - \frac{qT_{X_\tau}}{d_0}, v - \frac{qT_{Y_\tau}}{d_0}\right) d\tau \quad (11)$$

For any point lying on a fronto-parallel plane at a different depth d, the homography (under in-plane translation) is

$$H_{\tau_d} = K \begin{bmatrix} 1 & 0 & \frac{T_{X_\tau}}{d} \\ 0 & 1 & \frac{T_{Y_\tau}}{d} \\ 0 & 0 & 1 \end{bmatrix} K^{-1} \quad (12)$$

Correspondingly, the blur kernel at depth $d = kd_0$ is given by

$$h_d(\mathbf{x}, \mathbf{u}) = \frac{1}{T_e} \int_0^{T_e} \delta\left(u - \frac{qT_{X_\tau}}{kd}, v - \frac{qT_{Y_\tau}}{kd}\right) d\tau \quad (13)$$

From equations (11) and (13) and using the scaling property of delta, the relation between the blur kernels corresponding to depth d_0 and d turns out to be

$$h_d(\mathbf{x}, \mathbf{u}) = k^2 h_0(\mathbf{x}, k\mathbf{u}) \quad (14)$$

i.e., $h_d(x, y)$ is just a scaled version of the PSF at $h_0(\mathbf{x}\ \mathbf{y})$ where k is the scale factor.

IV. Blur Inconsistency

In this section, we devise inconsistency tests for the different scenarios considered in section III. The inconsistency criterion varies from case to case. For a fronto-parallel scene and in-plane camera translations, we show that the notion of PSF suffices. For the second case, we estimate the TSF (global) which describes the motion of the camera and use it to infer inconsistencies in spatially varying blur kernels. For the third scenario, the orientation and scale of PSFs help to unveil splicing.

Given a test image, it should first be inferred to which scenarios this image belongs. For this purpose, we randomly select $N_p (\approx 15)$ number of patches from the test image and estimate the PSF at each patch using an existing blur estimation technique . The patches are required to be spatially spread-out so as to capture as much information as possible about the motion of camera through the PSFs. Note that some of the blur kernels may also come from the spliced region but these are very few in number since the spliced region typically occupies only a fraction of the total image area. We pick a PSF from this set and compute its cross-correlation with every other kernel in the set. If h_1 and h_2 are two PSFs, then the cross-correlation is defined as $\rho(\mathbf{t}) = \frac{\sum_{\mathbf{a}} h_1(\mathbf{a})\, h_2(\mathbf{a}+\mathbf{t})}{\sum_{\mathbf{a}} h_1^2 \sum_{\mathbf{a}} h_2^2}$ where **a** represents the spatial location of the blur kernel while **t** represents the shift. A match is reported if the cross-correlation value is above 0.8. This is repeated for every PSF. If the number of matches for any kernel exceeds N_p, we conclude that the test image is representative of case (i) and this kernel is chosen as the reference PSF; else, the blur kernels are clearly space-varying and we proceed to employ the Fourier-Mellin transform (FMT) which is typically used for image registration and yields the rotation, scale and translation parameters governing the geometric relation between two images. We use FMT to compute the relative orientation of each kernel with respect to every other blur kernel. Depending upon the angle returned by FMT, we perform grouping such that PSFs within a group have similar orientation. If the cardinality of the largest group (set) is less than N_p, we conclude that the space-varying PSFs are not scaled versions of each other and hence the test image must belong to case (ii); else, we conclude that it represents case (iii) and the PSF with the largest support (from this set) is chosen as the reference PSF.

A. Case (i): Planar Scene and In-Plane Translations

For a static planar scene and in-plane translational camera motion, the displacement undergone by all the image pixels will be the same. Therefore, the TSF and PSF will be identical. In fact, all the pixels in the image should ideally have the same PSF. The reference PSF is extracted as discussed above. By comparing the PSFs at all the spatial locations with the reference PSF, inconsistency, if any, can be deduced.

Clearly, the above task necessitates the need for accurately estimating the blur kernel or the PSF. Within the context of image deblurring, many works have been proposed to estimate PSF from a single image. Most of these methods consider salient edges to estimate PSF. We use the blind deconvolution method proposed for estimating PSF. They have comprehensively compared their method with several state-of-the-art techniques for estimation of blur kernel and have demonstrated superior performance vis-a-vis others. The method is quite fast and has problems only in image regions with little or no texture. We also tested this method on several examples and found it to work quite satisfactorily. . Interestingly, their work reveals that salient edges are not always helpful in accurate estimation of PSF. They argue that if the width of the selected edge is less than the blur kernel width, it will, in fact, degrade the estimated kernel. They define a metric for selecting useful edges Here, ∇ denotes the gradient, B is the blurred image, and $N_h(x)$ is a window centered at pixel x. Those edges are ignored whose r value is less than a threshold value τ_r. The final edges are selected by combining the gradient threshold with the value of r. A coarse PSF is estimated by minimizing an objective function subject to Gaussian regularizer. In the second phase, a sparse

To identify and demarcate the spliced region, the entire image is divided into non-overlapping patches and the PSF is estimated at these patches .These PSFs are com-pared with the reference PSF using cross-correlation. Since PSF estimation is blind, there is a possibility of shift in the resultant PSF with respect to the original PSF. This is because these shifts will get compensated in the recovered latent image so as to satisfy the convolution model. Therefore, we propose to use cross-correlation [31] which is a shift-invariant metric to compare PSFs. The cross-correlation profile is thresholded to arrive at a rough estimate of the spliced borders. By selecting overlapping patches around this boundary, we arrive at a finer demarcation of the spliced region.

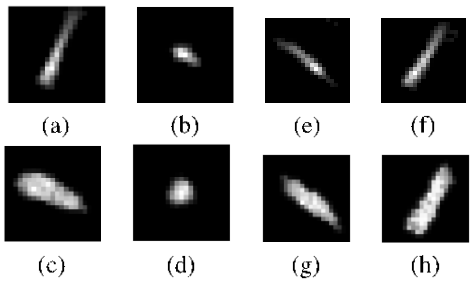


Fig. 2. (a) and (b) Kernels from authentic and spliced regions, respectively. (c) and (d) Reconstructed kernels corresponding to Fig. 2(a) and (b), respectively. (e) and (f) Both the kernels are from the authentic region. (g) and (h) Reconstructed kernels corresponding to Fig. 2(e) and (f), respectively.

B. Case (ii): Flat Scene and In-Plane Translations With Rotations

For this scenario, the PSFs will be space-varying. However, the TSF will remain valid over the entire image since there are no depth variations. TSF estimation is very important because it gives information about the motion of the camera. We now describe how knowledge of PSFs can be used to infer TSF. For this purpose, let us revisit equation (5) which describes the relation between TSF and PSF. For each blur kernel, this equation can be represented in matrix-vector form as $h_{\mathbf{p}_l} = M_l h_D$, $l = 1, 2 \ldots K$, where $h_{\mathbf{p}_l}$ indicates the blur kernel at location p_l while M_l is an interpolation matrix, whose entries are computed from the location of the blur kernel and by bilinear interpolation. If the size of the blur kernel is $m \times n$, then after rearranging in vector form $N_h \times 1$ ($N_h = m.n$), the matrix M_l will be of size $N_h \times N_T$ where N_T denotes the number of possible transformations. To generate the values of matrix M, we find the displacement of point $\mathbf{p}$ due to each transformation $\lambda \in D$ and a unit entry is made at the row which corresponds to the co-ordinates of displacement.

Assume that we have 'K' authentic blur kernels. The procedure to get these kernels is discussed later. Arranging the K blur kernels selected in the form of a column vector $\bar{h}$, the relationship between the blur kernels and the TSF can be written as $\bar{h} = Mh_D$, where the size of M will be $K N_h \times N_T$. As discussed earlier in section IB, even though the dimension of possible transformations can be quite large, the latent image will actually have undergone only a few transformations since the motion due to camera-shake is sparse. This leads to the incorporation of a sparsity constraint on the TSF through the L_1 norm. The TSF can be estimated by minimizing the cost

$$\_\bar{h} - Mh_D\_^2 + \lambda_s \_h_D\_1. \qquad (15)$$

This is an L_1 norm regularized least squares problem and we solve it using LASSO (Least Absolute Shrinkage and Selection Operator). More specifically, we use the *nnLeastR* function of SLEP package [35] to solve this. The value of λ_s is chosen such that equation (15) gives minimum error for non-negative values of TSF.

C. Automatic Estimation Of TSF

Note that TSF estimation requires that the PSFs come from the authentic region. From the N_p randomly picked blur kernels, we need to eliminate outliers (those belonging to the spliced region) to enable accurate estimation of TSF. We exploit the following PSF compatability criterion for this purpose. For a given blur kernel h_1, one can determine (from equation 5) the set of transformations corresponding to each non-zero weight of h_1. We represent this set of homographies by S_1. Note that S_1 is a super-set of the actual homographies responsible for h_1. In a similar manner, we determine the set of homographies S_2 for a kernel h_2 at a different location. Let the intersection of S_1 and S_2 be a set denoted by S_{12}. Using the transformations in S_{12}, we attempt reconstruction of h_1 and h_2. Let the recomputed kernels be $\hat{h}_1$ and $\hat{h}_2$, respectively. If h_1 and h_2 had originated from the same TSF, then the supports of h_i and $\hat{h}_i$, $i = 1, 2$ would be compatible i.e, wherever h_i is non-zero, $\hat{h}_i$ will also have a non-zero entry. On the other hand, if one of the two kernels is from a spliced region, then the above compatibility is lost. We randomly choose one blur kernel and test its compatibility with the remaining $N_p - 1$ kernels. We repeat this test for each kernel. Since the number of kernels from spliced region is expected to be few in number, this test allows us to segment out kernels that are outliers since the effective number of compatibility failures will be higher for a spliced kernel. The resultant set of compatible (authentic) blur kernels is then used to arrive at the TSF.

The above procedure is best explained with an example. Let us first consider the case where the PSFs come from authentic [Fig. 2(a)] and spliced region [Fig. 2(b)]. We display the reconstructed PSFs [Figs. 2(c) and (d)] using the intersection of homographies S_1 and S_2 of the kernels in Fig. 2(a) and (b). Note that the kernels are not compatible. This is quite in contrast to the case when both the kernels come from authentic region [Fig. 2(e) and (f)]. The reconstructed kernels from the intersection set are given in Fig. 2(g) and (h). Observe that the intersection of their homographies S_1 and S_2 is very well able to explain both the PSFs.

From the estimated TSF, we can predict the blur kernel at any point in the image using equation (5). To identify and demarcate the spliced region, we adopt the same procedure as described for case (i) except that PSFs at non-overlapping patches obtained using [25] are compared with PSFs *predicted* at those locations by the estimated TSF. Post thresholding of cross-correlation profile, overlapping patches are used to carve out a finer boundary of the spliced region.

These steps are summarized in Algorithm 1.

Algorithm 1

1. Randomly select N_p patches from across the image
2. Estimate the PSFs of the selected patches using [25]
3. Eliminate outlier blur kernels using the compatibility test described in this section
4. Estimate true TSF using only the authentic kernels
5. Estimate PSF across non-overlapping patches using [25]
6. Deduce PSF at these locations from the estimated TSF of step 4
7. Compare both the PSFs by finding their cross-correlation value
8. Detect and demarcate the spliced region by thresholding the cross-correlation profile

D. Case (iii): Depth-Varying Scene and In-Plane Translations

In the case of a 3D scene, the blur is always space-variant. Even for simple translational motion of the camera, the pixel motion of scene points near to the camera will be faster as compared to faraway points. However, as derived in equation (14), there exists an elegant scaling relationship between PSFs at different depths which plays a vital role in defining inconsistency. If the PSF is from a spliced region, then it should not be possible to express this PSF as a scaled version of the reference (h_{ref}) PSF in the authentic region. The reference PSF is obtained as described in the beginning of section IV. We again resort to FMT to compare the PSFs. Let h_i be a blur kernel at any image location. The kernels h_{ref} and h_i are given as input to FMT. If the rotation angle is less than a threshold, we warp h_i by the angle output by FMT to yield $\hat{h}_i$ such that the relative angle between PSFs h_{ref} and h_i is close to zero. Next, we warp h_i by the scale factor to yield $\hat{h}_i$. Finally, we use cross-correlation between h_{ref} and $\hat{h}_i$ for determining whether h_i belongs to authentic region. On the other hand, if the resultant rotation angle is greater than threshold, we can straightaway infer that the region around h_i is spliced. As before, the comparison between h_{ref} and PSF obtained using [25] is carried out initially at non-overlapping patches in the image. The rough splicing boundary obtained from the cross-correlation profile is then refined to yield the actual spliced region.

V. Experimental Results

We verify the performance of our approach on several synthetic and real examples for the following situations. Case (i): Flat scene and in-plane camera translations. Case (ii): Flat scene with in-plane camera translations and rotations. Case (iii): 3D scene and in-plane camera translations. The examples are chosen such that it is difficult to visually decipher whether the image was spliced or not. The cross-correlation threshold was set as 0.8 to determine whether two blur kernels are similar. We constructed a database of more than thirty images representing different scenarios which also included few untampered images. Unless otherwise stated, the actual and detected boundaries are shown in green and red, respectively.

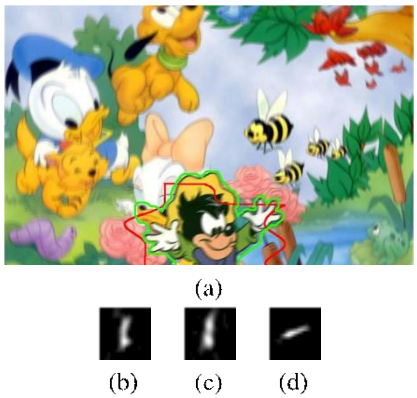


Fig. 3. Case (i). (a) Spliced region demarcated by our approach. (b) and (c) PSFs randomly selected from authentic region. (d) PSF from spliced region.

For the simulated examples, the original images were obtained from the web and these were blurred synthetically corresponding to each of the three different scenarios considered in the paper. For the real cases, we used a Canon 60D hand-held camera for capturing the images.

We have also compared our method which perform splicing detection using blur as a cue. We give a few representative comparison examples to underscore how our method is able to advance the state-of-the art. The technique proposed in uses inconsistencies in defocus blur based on DCT coefficients as a cue for detecting forgeries. Specifically, it analyzes the differences in degree of blur to detect the forged region by assuming the blur to be Gaussian. The blur threshold value is chosen relative to the global blurring in the entire image. The method is a recent work that assumes uniform space-invariant motion blur and characterizes the blur kernel by its length and orientation. We employed the modified Hausdorff distance [36] to quantify the accuracy of our method.

A. Flat Scene and In-Plane Translations

We first discuss some simulations followed by real examples. Deciphering the specific scenario (i.e, case (i)) followed by reference PSF estimation are both carried out as described earlier in the beginning of section IV.

Synthetic case: In Fig. 3(a), we show a spliced image from our dataset. The authentic and spliced regions were synthesized with different translational blurs. In fact, we copied a patch from another translationally blurred image to mimic the effect of splicing.

The spliced image of Fig. 3(a) was given as input to our method. The size of the image was 1024×646 pixels while the size of the spliced region was 350×290 pixels. The given image was divided into non-overlapping patches of size 121×121 pixels and for each patch the PSF corresponding to the center of the patch was estimated .The maximum size of the blur kernel was chosen as 21×21 pixels. The patch size should be greater than the maximum size of the expected blur kernel to reliably estimate the PSF.

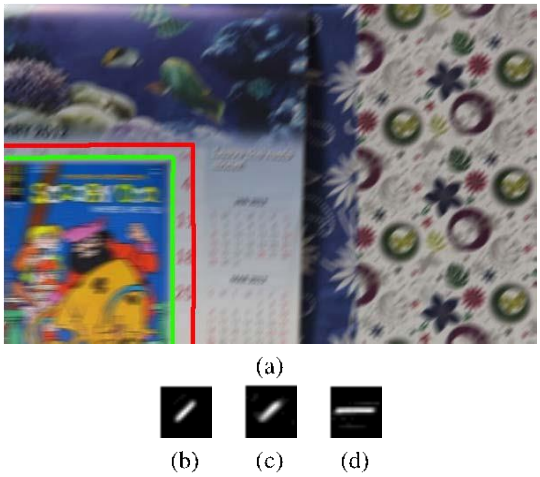


Fig. 4. Case (i). (a) Real spliced image. (b) and (c) PSF from genuine region and the reference PSF, respectively. (d) PSF from spliced region.

In Fig. 3(b) and (c), we show two PSFs at random locations from the authentic region. Note they look quite alike since the blur is space-invariant. In contrast, the PSF from the spliced region [Fig. 3(c)] looks starkly different. Its cross-correlation value with the reference PSF was only 0.47 which is much below the threshold value. The demarcated boundary of the spliced region in the center bottom of the image has been correctly detected in Fig. 3(a). Using the modified Hausdorff distance, the deviation of the estimated contour from the true contour was found to be 1.8%.

Real case: A Canon 60D camera was mounted on a portable stand to induce translation motion blur along horizontal and vertical directions. We captured two images with different amounts of motion blur and arrived at the spliced image [Fig. 4(a)] in our dataset. Analogous to the synthetic case, the image was divided into non-overlapping patches (of size 101×101 pixels) and the PSFs corresponding to the centers of these patches were computed These PSFs were compared with the reference PSF and the cross-correlation values were used to detect spliced PSFs using a threshold of 0.8. In Fig. 4(b)–(d), we show the reference PSF and the PSFs corresponding to authentic and spliced region. Note the similarity between Fig. 4(b) and (c) (correlation $=$ 0.94) while the wrong PSF is distinctly different (correlation $=$ 0.3). The spliced region has been correctly demarcated by our method as shown in Fig. 4(a). The contour deviation error for this example is 1.69%.

In Fig. 5(b) and (c), we give some comparison results with [16] and [15], respectively. Since case (i) has space-invariant motion blur, the method works satisfactorily. On the other hand, fails since it is designed for optical blur.

B. Flat Scene and In-Plane Translations With Rotations

In this section, due to presence of rotational component, the pixels near the axis of rotation suffer relatively less displacement which results in space-variant blurring. Given a test image, the scheme outlined in section IV was followed

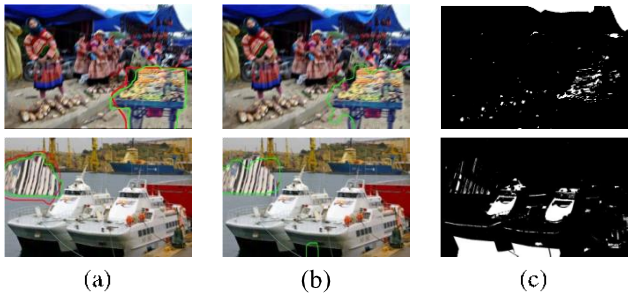


Fig. 5. Comparison results for case (i) obtained using (a) our method, (b) [16], and (c) [15], respectively.

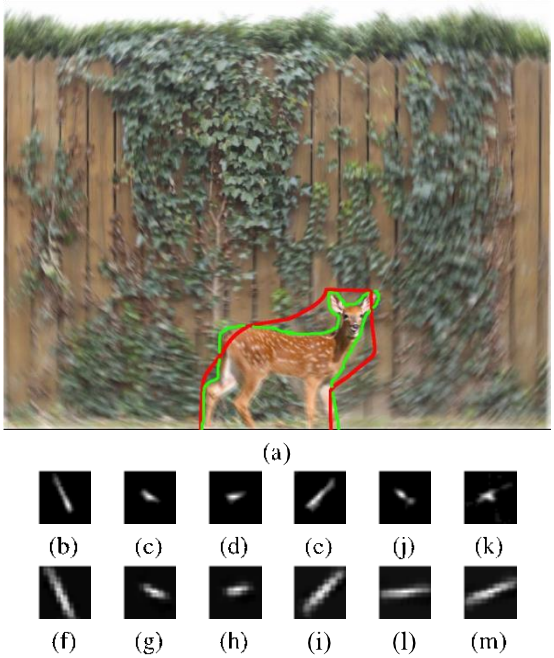


Fig. 6. Case (ii) (a) Demarcation of spliced region. (b)–(e) Authentic PSFs obtained by [25]. (f)–(i) Authentic PSFs corresponding to (b)–(e) predicted by TSF. (j) and (k) Spliced PSFs obtained by [25]. (l) and (m) Spliced PSFs corresponding to (j) and (k) predicted by TSF.

to first establish that the image belongs to case (ii). The compatibility test described in section IV-C was used to obtain authentic kernels to estimate the TSF.

Synthetic case: We blurred two different images from our dataset with different (but known) TSFs. The TSFs were chosen such that they produce both in-plane translational and rotational blur. Each weight in the TSF corresponds to a particular homography. These two blurred images were used to arrive at the spliced image shown in Fig. 6(a). The maximum size of the blur kernel was assumed to be 21×21 pixels. Using the compatibility test, authentic PSFs were estimated at spatial locations that were spread-out across the image to compute the camera motion. As expected, the blur was found to be space-varying. From the authentic PSFs, we estimated the underlying TSF.

The test image of Fig. 6(a) was next divided into non-overlapping patches of size 121×121 pixels and the PSF was estimated at each patch .We also derived the

..

Fig. 7. Case (ii). (a) and (b) Additional results on synthetic examples from our dataset using the proposed method.

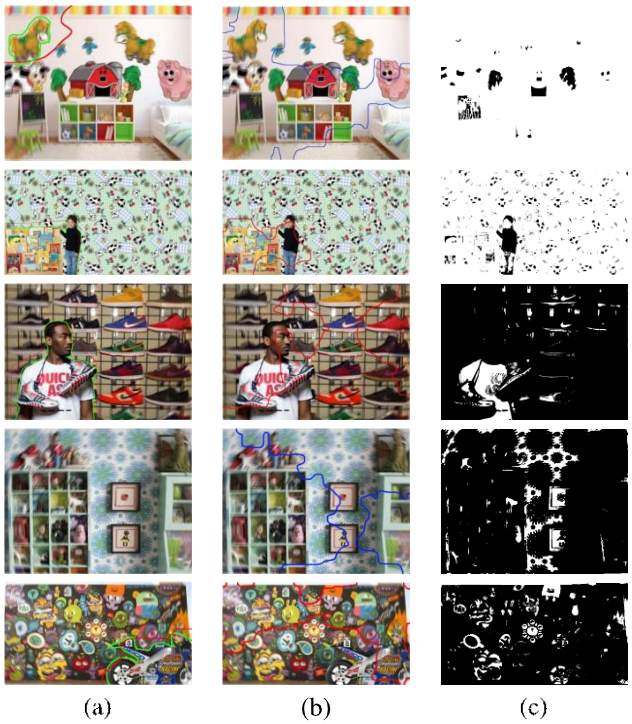


Fig. 8. Case (ii) Outputs obtained using (a) our method, (b) [16], and (c) [15], respectively.

expected PSFs at these locations using the estimated TSF and compared them with the directly estimated PSFs Thresholding of the correlation profile yields a rough outline of the spliced region. This is further refined with overlapping patches to arrive at the final spliced boundary shown in Fig. 6(a). We have given the PSFs obtained using [25] and the corresponding PSFs as predicted by the TSF in both authentic and spliced regions in Fig. 6(b)–(i) and (j)–(m), respectively. Note that the estimated and true PSFs in the authentic region are quite close. The percentage deviation from the actual splicing boundary is 1.3%.

Two more challenging and synthetically spliced examples are shown in Fig. 7(a) and (b). Following the procedure discussed in the previous example, the detected spliced regions indeed represent correct demarcation. The deviation errors are 4.39% and 1.7%, respectively.

Since the examples in case (ii) involve space-variant blurring, we compared the performance of our method with that of [15] and [16]. While we implemented carefully on our own, the authors of [16] mailed us the results for their method. Few representative comparison results from our dataset are shown in Fig. 8 for the best possible threshold value for each method. The method of outputs a binary image corresponding to two different levels of blur while outputs the estimated demarcated boundary. Observe that our method emerges a clear winner. The outputs obtained using and shown in Fig. 8 are clearly wrong and bring out the inability of these methods to detect splicing under space-variant blurring conditions. The performance of encounters difficulties when blur kernels have arbitrary shape. Furthermore, its performance degrades severely in the presence of space-variant blur. The technique was able to identify the spliced region correctly in only one example (second row) and performs poorly on all the other images. Since this technique is primarily based on PSF comparison, the inconsistency criteria defined ,fails to work when the blur is space-varying. We also tested these methods on the image in the fourth row of Fig. 8 which incidentally is untampered, although it has space-varying blur. While our method was able to rightly conclude that the given test image was authentic, the comparison methods falsely ended-up dividing the image into authentic and spliced regions.

Real case: We captured two images of flat scenes with different camera shakes (comprising of in-plane translations and rotations). One such spliced image from our dataset is shown in Fig. 9(a). By following the same procedure as outlined in the synthetic case, outlier PSFs were removed and only the authentic blur kernels were used to estimate the TSF. The image patches were of size of 121×121 pixels while the maximum size of the PSF was assumed to be 31×31 pixels. Some of the authentic PSFs are shown in Fig. 9(b)–(e). Note that even in the authentic region, the blur exhibits significant differences underscoring its space-variant nature. Again, following the steps discussed in the synthetic case for detection and demarcation of the spliced boundary, the final result with the spliced region marked is shown in Fig. 9(a) and has a deviation of 1.6%.

Some more real examples are shown in Fig. 10(a) and (b). We choose patches of size 121×121 pixels and the maximum

..

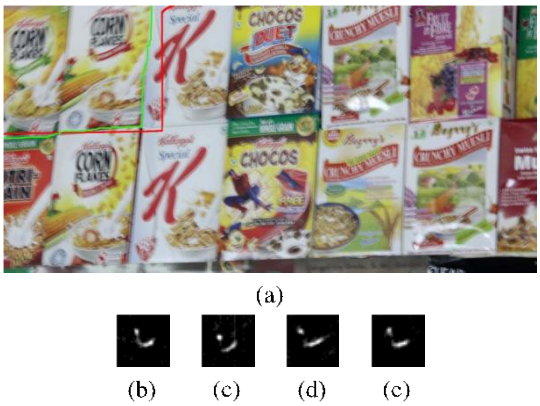


Fig. 9. Case (ii). (a) Real splicing example and output of our method. (b)–(e) PSFs from authentic region.

Fig. 10. Case (ii). (a) and (b) Results obtained using our method on real examples.

size of the blur kernel was assumed to be 31×31 pixels. Note that the spliced region has been correctly detected yet again by our method. The deviation errors for Fig. 10(a) and (b) are 4.9% and 1.6%, respectively.

C. 3D Scene and In-Plane Translations

In this section, we consider the scenario of a 3D scene and translational motion of the camera. For this case, the scaling relationship of equation (14) lies at the core of detecting splicing. Yet again given a test image, the methodology outlined in section IV was used to infer that the image represents case (iii). This was followed by estimation of reference PSF.

Synthetic case: A 3D scene from our dataset with known (bi-layered) depth map was first considered as shown in

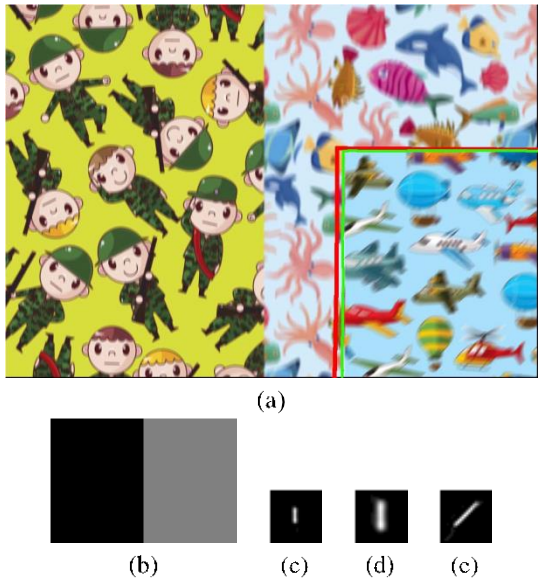
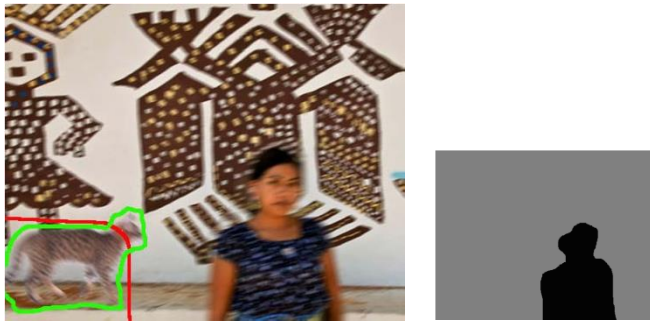


Fig. 11. Case (iii). (a) A bi-layered scene. (b) Depth map. (c) and (d) Authentic and reference PSFs, respectively. (e) PSF from spliced region.

Fig. 11(a). From the figure, notice that the left-half is near to the camera while the right-half is farther away from the camera. We chose an arbitrary translational PSF to induce blur on the left-half of the image. Using the scaling relation, the corresponding PSF for the right-half was generated. From these PSFs, the blurred image was generated according to the depth map of Fig. 11(b). We used a different image for splicing to arrive at the spliced image of Fig. 11(a).

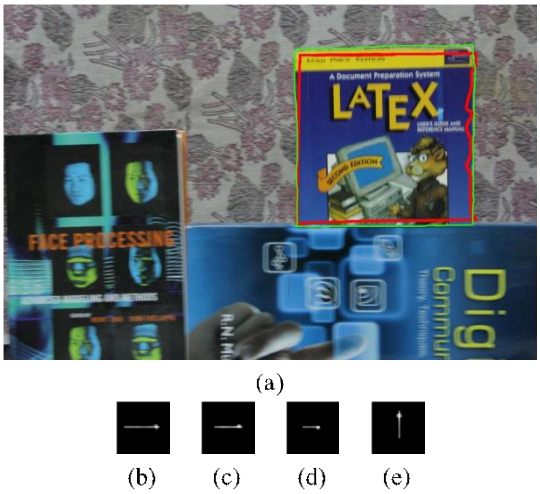
The test image was divided into non-overlapping blocks of size 121×121 pixels and the PSF (maximum size of 31×31 pixels) was estimated for each block using yet again the blur estimation technique of [25]. Following the procedure discussed in section IV, these PSFs were compared with the reference PSF using FMT. The threshold value with respect to angle was chosen as 8 degrees. If the angle returned by FMT is greater than this threshold, we infer that the patch is spliced. The reference PSF is shown in Fig. 11(c). In contrast, the PSF deduced to be from the spliced region [Fig. 11(e)] is distinctly different. We gave two PSFs (one each from authentic and spliced region [Fig. 11(c) and (d)] as input to FMT. The relative angle between these two PSFs was found to be 46 degrees. On the other hand, a comparison of two authentic PSFs [Fig. 11(b) and (c)] gives an angle close to zero degrees while the scale factor was 0.53. We also checked the scaling relationship of each PSF with the reference PSF (i.e, verified the similarity by comparing the cross-correlation after re-scaling to be sure of the authenticity of a patch). The detected spliced region is shown in Fig. 11(a). The estimated contour is very close (only 0.8% deviation) to the true contour.

We show yet another example in Fig. 12(a) with depth map as given in [Fig. 12(b)]. The person standing near to the camera is authentic while the cat in the left-bottom corner was spliced. When this image was given as input to our method, the spliced region could be correctly demarcated



(a) (b)

Fig. 12. Case (iii). (a) Output by the proposed technique. (b) Depth map of the 3D scene.



(a)

(b) (c) (d) (e)

Fig. 13. Case (iii). (a) Splicing in a real 3D scene. (b)–(d) PSFs estimated at different layers. (e) PSF from spliced region.

(a)

(b) (c) (d)

Fig. 14. Case (iii). (a) The output of our method for a real outdoor 3D scene. (b) and (c) PSFs obtained from authentic region. (d) PSF from the spliced region.

(a) (b) (c)

Fig. 15. Outputs for the case of 3D scene with camera translations. (a) Our method, (b) [16], and (c) [15], respectively.

(5.1% deviation error). The method were again found to perform poorly on these images as the blur was space-varying.

Real case: A real spliced example from our dataset is shown in Fig. 13(a). A 3D scene was captured with the help of a moving translational stage to induce purely translation blur in the image. The depth of the scene varied from 15 cm to 50 cm. Patches of size 101×101 pixels were selected at different locations while the maximum size of the blur kernel was taken to be 31×31 pixels. The left-bottom corner (the face processing book) was placed nearest to the camera, the right-bottom (digital communication book) was lying around the middle depth layer, while the background region (blanket) was farthest from the camera. Among the randomly estimate PSFs across the image, the ones corresponding to these layers are shown in Fig. 13(b)–(d). The parallax effect is very clear in this case. We observe that the PSFs are just scaled versions of one another. We also note that the PSF for the spliced region is very different from the blur kernels estimated in the authentic region. Again, we employed FMT for comparing PSFs estimated at non-overlapping patches in the image with the reference PSF. If the relative angle between PSFs was more than 8 degrees, the PSF was adjudged as belonging to the spliced region. If not, inverse warping was applied on one of the PSFs with the FMT scale factor and this was compared with the reference PSF using cross-correlation. The final result shown in Fig. 13(a) with the spliced region marked reveals the correctness of our approach (contour deviation $= 1.8\%$).

Lastly, we show the example of Fig. 14 from our dataset. Here we selected patch size of 121×121 pixels and the maximum kernel size assumed was 21×21 pixels. By following the procedure described in earlier examples, the final result shown in Fig. 14(a) was arrived at with the spliced region marked. The PSFs corresponding to authentic and spliced regions are shown in Fig. 14(b)–(d). The algorithm is quite correct in its inference since the two individuals in the rear indeed belong to a different image and the contour deviation is 1.6%. Some comparison results for this case are shown in Fig. 15. Since case (iii) again consists of space-variant blur, the methods of [15] and [16] perform poorly.

The splicing boundaries in our examples were arbitrary in shape which renders automatic detection of spliced areas quite difficult. Yet, the estimated spliced contours were close to the actual boundary. When quantitatively evaluated, the deviation from the true contour was less than 5%. Some deviation is to be expected since (i) the boundaries are blurred, and (ii) our method is patch based.

VI. Conclusion

We conluded a passive approach to unveil image splicing by using motion blur as a cue. Since the motion blur induced at a point depends on the depth of the corresponding scene point, its location in the image, as well as camera motion, any inconsistency in motion blur reveals the presence of splicing. Three interesting scenarios were considered and suitable inconsistency criteria were proposed. The method is automatic in the sense that neither the spliced region nor the authentic region is assumed to be known. The approach was validated on several synthetic and real examples and comparisons were also given to demonstrate its efficacy over existing techniques. In future, we plan to extend this method for the more general case of a 3D scene involving both in-plane translation and rotational camera motion. This extension is non-trivial since the notion of TSF is valid for a plane.

References

[1] M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proc. IEEE Int. Conf. Ind. Inf.*, Aug. 2005, pp. 709–716.

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

[3] B. Mahdian and S. Saic, "A bibliography on blind methods for identifying image forgery," *Signal Process., Image Commun.*, vol. 25, pp. 389–399, Jul. 2010.

[4] A. Propescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[5] T. T. Ng and S. F. Chang, "A model for image splicing," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Oct. 2004, pp. 1169–1172.

[6] W. Wang, J. Dong, and T. Tan, "Effective image splicing detection based on image chroma," in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 1257–1260.

[7] W. Chen, Y. Q. Shi, and W. Su, "Image splicing detection using 2-D phase congruency and statistical moments of characteristic function," *SPIE Proc.*, vol. 6505, pp. 65050R-1–65050R-5, Jan. 2007.

[8] M. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. 7th Workshop Multimedia Security*, 2005, pp. 1–10.

[9] E. Kee and H. Farid, "Exposing digital forgeries from 3-D lighting environments," in *Proc. IEEE Int. Workshop Inf. Forensics Security*, Dec. 2010, pp. 1–6.

[10] M. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 450–461, Sep. 2007.

[11] T. T. Ng, S. F. Chang, and M. P. Tsui, "Using geometry invariants for camera response function estimation," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2007, pp. 1–8.

[12] Y. F. Hsu and S. F. Chang, "Camera response functions for image forensics: An automatic algorithm for splicing detection," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 816–825, Dec. 2010.

[13] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[14] X. Wang, B. Xuan, and S. Peng, "Digital image forgery detection based on the consistency of defocus blur," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2008, pp. 192–195.

[15] D. Y. Hsiao and S. C. Pei, "Detecting digital tampering by blur estimation," in *Proc. 1st Int. Workshop Syst. Approaches Digital Forensic Eng.*, Nov. 2005, pp. 264–278.

[16] P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting discrepancies in motion blur," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 443–452, Jun. 2011.

[17] M. Purnachandrarao and A. N. Rajagopalan, "Harnessing motion blur to uncover splicing," in *Proc. IEEE Int. Conf. Image Processing*, Melbourne, Australia, Sep. 2013, pp. 4507–4511.

[18] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 529–538, Sep. 2008.

[19] S. Lin, J. Gu, S. Yamazaki, and H. Y. Shum, "Radiometric calibration from a single image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jul. 2004, pp. 938–945.

[20] C. Paramanand and A. N. Rajagopalan, "Unscented transformation for depth from motion-blur in videos," in *Proc. IEEE Workshop 3rd Dimensional Inf. Extraction Video Anal. Mining*, Jun. 2010, pp. 38–44.

[21] S. Dai and Y. Wu, "Motion from blur," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[22] C. Paramanand and A. N. Rajagopalan, "Inferring image transformation and structure from motion-blurred images," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 73.1–73.12.

[23] C. Paramanand and A. N. Rajagopalan, "Depth from motion and optical blur with an unscented Kalman filter," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2798–2811, May 2012.

[24] S. Chaudhuri and A. N. Rajagopalan, *Depth From Defocus: A Real Aperture Imaging Approach*. New York, NY, USA: Springer-Verlag, 1999.

[25] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 157–170.

[26] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Process.*, vol. 17, no. 2, pp. 105–116, Feb. 2008.

[27] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 491–498.

[28] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 171–154.

[29] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006.

[30] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–10, Aug. 2008.

[31] Z. Hu and M. H. Yang, "Good regions to deblur," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 59–72.

[32] C. S. Vijay, C. Paramanand, and A. N. Rajagopalan, "HDR imaging under non-uniform blurring," in *Proc. 12th Int. Conf. Comput. Vis.*, Oct. 2012, pp. 451–460.

[33] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for transla-tion, rotation, and scale-invariant image registration," *IEEE Trans. Image Process.*, vol. 5, no. 8, pp. 1266–1271, Aug. 1996.

[34] (2010). *When Tripods are not Permitted* [Online]. Available: http://www.shutterbug.com/content/when-tripods-are-not-permitted

[35] (2013, Mar. 23). *Sparse Learning with Efficient Projections* [Online]. Available: http://www.public.asu.edu/∼jye02/Software/SLEP/

[36] M. P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. 12th IAPR*, Oct.

