Efficient Change Detection for Very Large Motion Blurred Images

Vijay Rengarajan, Abhijith Punnappurath, A.N. Rajagopalan
Indian Institute of Technology Madras
Chennai, India
{*ee11d035,ee10d038,raju*}*@ee.iitm.ac.in*

Guna Seetharaman
Information Directorate, AFRL/RIEA
Rome, NY USA
guna@ieee.org

Abstract—**In this paper, we address the challenging problem of registration and change detection in very large motion blurred images. The unreasonable demand that this task puts on computational and memory resources precludes the possibility of any direct attempt at solving this problem. We address this issue by observing the fact that the camera motion experienced by a sufficiently large sub-image is approximately the same as that of the entire image itself. We devise an algorithm for judicious sub-image selection so that the camera motion can be deciphered correctly, irrespective of the presence or absence of occluder. We follow a reblur-difference framework to detect changes as this is an artifact-free pipeline unlike the traditional deblur-difference approach. We demonstrate the results of our algorithm on both synthetic and real data.**

Keywords-**very large images; motion blur; change detection;**

I. INTRODUCTION

Feature-based approach is commonly used in image registration. There are several methods for feature extraction such as SIFT, SURF, ORB and MSER (Lowe et al. [1], Bay et al. [2], Rublee et al. [3], Matas et al. [4]). These algorithms are primarily designed to work on small to medium-sized images. Memory requirement is an important factor to consider when employing these approaches for high resolution images. Huo et al. [5] showed that SIFT features require a prohibitively huge amount of memory for very large images. Another drawback of feature-based approaches while working on large images is incorrect feature matching due to the occurrence of multiple instances of similar objects across the image (Carleer et al. [6]). Coarse-to-fine strategies for feature matching are followed by Yu et al. [7] and Huo et al. [5] to enable matching.

Within the scope of the problem tackled in this paper, there is yet another deterrent in adopting feature-based approach and that is blur. Motion blur is a common occurrence in aerial imagery where the imaging vehicle is always on the move. In addition to geometric matching, photometric matching becomes essential in such a scenario. Feature-based approaches are not designed to handle the presence of blur and fail to reliably detect features in the presence of blur. A traditional approach to handle this situation is to first deblur the observation, and then pass on the resultant image to the change detection pipeline where it is compared with a clean reference image after feature-based registration.

A number of approaches already exist in the literature to perform deblurring. Blind deconvolution methods recover a sharp image from the blurred image with an unknown blur kernel under the assumption of space-invariant blur. Fergus et al. [8] and Xu et al. [9] estimate space-invariant kernels and deblur the image. Space-variant blur based approaches include that of Gupta et al. [10] who model a motion density function to represent the time spent in each camera pose and to generate spatially varying blur kernels and eventually restore the deblurred image using a gradient-based optimisation. Whyte et al. [11] define a global representation for space-variant blur similar to the point spread function for space-invariant blur to restore the motion blurred image using MAP approach. Hu et al. [12] estimate weights for each camera pose in a restricted pose space using a backprojection model while deblurring is carried out by employing a gradient-based prior. Paramanand et al. [13] estimate a transformation spread function to determine the depth map, which is used to deblur the image. Leveraging gradient sparsity, Xu et al. [14] proposed a unified framework to perform both uniform and non-uniform image deblurring.

An issue with such a deblur-difference framework is that it must deal with the annoying problem of artifacts that tend to get introduced during the course of deblurring. A more serious issue within the context of this paper is that none of the deblurring methods are designed to handle very large images. Furthermore, the deblurring methods would fail if the occluder was not static since the image will then be governed by two independent motions.

In the problem of change detection, the goal is to detect the difference between a reference image with no artifacts and an observed image which is blurred and has viewpoint changes as well. We develop a unified framework to register the reference image with the blurred image and also to detect occlusions simultaneously. The occluder is not constrained to be static. To address the issue of image size, we reveal that the camera motion can be elegantly extracted from only a part of the observation. For reasons discussed earlier, we follow a reblur-difference pipeline instead of a deblur-difference pipeline. While Punnappurath et al. [15] also followed a reblur-difference strategy, our work is more general and, in fact, subsumes their work. Specifically, we

use an optimisation framework with partial non-negative constraint which can handle occlusions of any polarity, and we efficiently tackle the issue of large image dimension. In addition, our algorithm can also deal with dynamic occluders. In our approach, the estimated camera motion is used to reblur the reference image to photometrically match it with the observed image, and thereby detecting the changes.

Our main contributions in this paper are as follows.

- We develop a scheme to automatically select good sub-images from the given observation to enable reliable estimation of the camera motion.
- We propose a memory efficient and computationally tractable registration scheme to estimate the camera motion from the selected sub-image, irrespective of the presence or absence of occlusions in the sub-image.
- We advocate a reblur-difference pipeline for geometric as well as photometric registration of the reference image and the blurred observation for robust change detection.

In Section II we briefly discuss the motion blur model and develop an optimisation framework to simultaneously solve for camera motion and detect occlusions. In Section III, we discuss an efficient algorithm for registration of very large images. Section IV demonstrates our experimental results and we conclude in Section V.

II. Blur, Registration and Occlusion

In this section, we briefly discuss motion blur model in a camera. We then show how to invoke an optimisation framework to simultaneously register the reference image with the blurred image as well as detect occlusions, if any.

A. Motion Blur Model

Each pixel in a digital camera embeds a sensor which collects photons from the scene. A digital circuit provides the intensity value based on the number of photons received. All the pixels are exposed for a finite amount of period T_e which is the exposure time of the image. The resultant intensity at each pixel is the average of all intensities that the pixel sees during the exposure period. Let us denote the camera path during the image exposure period by $\mathbf{p}(t)$ for $0 \leq t \leq T_e$. The camera motion has six degrees of freedom, three for rotations and three for translations. Let $\mathbf{f}$ represent the image observed by the camera during an infinitesimal amount of time. Let $\mathbf{g}$ be the image observed by the camera with an exposure time T_e. Let the number of pixels in the image be N, so that $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{N \times 1}$. Then, we have

$$\mathbf{g} = \frac{1}{T_e} \int_0^{T_e} \mathbf{f}_{\mathbf{p}(t)} \, dt. \qquad (1)$$

where $\mathbf{f}_{\mathbf{p}(t)}$ is the image observed by the camera due to the pose $\mathbf{p}(t)$ at a particular time t.

When there is no motion, the camera observes the same scene during the entire exposure time, and hence a clean image without any blur is observed. In this case, $\mathbf{p}(t) = \mathbf{0}$ for all $0 \leq t \leq T_e$, and $\mathbf{g} = \mathbf{f}$. Thus $\mathbf{f}$ represents also the image seen by the camera with no motion during the exposure time T_e. In the presence of camera motion, the sensor array records different scenes at every instant during the exposure time. The resultant image thus embodies blur in it, and we have $\mathbf{g} \neq \mathbf{f}$.

We discretise the continuous model in (1) with respect to a finite camera pose space $\mathcal{P}$. We assume that the camera can undergo only a finite set of poses during the exposure time. Let us define $\mathcal{P} = \{\mathbf{p}\}_{i=1}^{|\mathcal{P}|}$ as the set of possible camera poses, where $|\cdot|$ denotes cardinality. We can write (1) equivalently as

$$\mathbf{g} = \sum_{\mathbf{p}_k \in \mathcal{P}} \omega_{\mathbf{p}_k} \, \mathbf{f}_{\mathbf{p}_k} \qquad (2)$$

where $\mathbf{f}_{\mathbf{p}_k}$ is the warped reference image $\mathbf{f}$ due to the camera pose $\mathbf{p}_k$. Each scalar $\omega_{\mathbf{p}_k}$ represents the fraction of exposure time that the camera stayed in the pose $\mathbf{p}_k$. Thus we have $\sum_{\mathbf{p}_k} \omega_{\mathbf{p}_k} = 1$ if the camera takes only the poses from the defined pose set $\mathcal{P}$. The weights of all poses are stacked in the pose weight vector $\boldsymbol{\omega}$. Since the averaging effect removes the time dependency of the continuous camera path $\mathbf{p}(t)$, this discretisation model is valid. We assume that the scene is far enough from the camera such that planarity can be assumed. We follow a homography-based approach to generate the warps for different camera poses.

B. Joint Registration and Occlusion Detection

We now consider the problem of estimation of camera poses during exposure. Given a reference image $\mathbf{f}$ which is captured with no camera motion, and a blurred image $\mathbf{g}$ arising from an unknown camera motion, the following problem can be posed to solve for the camera motion.

$$\widetilde{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}} \|\mathbf{g} - \mathbf{F}\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1 \text{ subject to } \boldsymbol{\omega} \succeq 0 \quad (3)$$

Here $\mathbf{F}$ is the matrix which contains the warped copies of the reference image $\mathbf{f}$ in its columns for the camera poses in $\mathcal{P}$. In the whole pose space, the camera can be moved through only a small set of poses. This is prioritised as the ℓ_1 norm in (3) which promotes sparsity of the pose weight vector. The above problem seeks the sparsest non-negative pose weight vector which satisfies the relation between reference and blurred images. Matrix-vector multiplication $\mathbf{F}\boldsymbol{\omega}$ is an equivalent form of (2).

This model, however, does not accommodate occluding objects in the observation $\mathbf{g}$ although this is quite often the case in aerial surveillance systems. To handle this, let $\mathbf{g}_{\text{occ}}$ be the observed image captured with blur and occlusions. We model the occlusion as an additive term to $\mathbf{g}$ to give $\mathbf{g}_{\text{occ}} = \mathbf{g} + \boldsymbol{\chi}$. The occluded image $\boldsymbol{\chi}$ can take both positive and negative values since the occluded pixels can have intensities
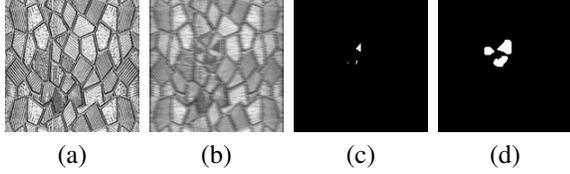
Figure 1. Detected changes using [15] (c) and our algorithm (d), between the reference image (a) and blurred and occluded observation (b).

Figure 2. Shown are some of the reference (top row) and blurred (bottom row) images used in our experiments in Section III.

greater or lesser than the intensities purely explained by blur. This model can then be written as

$$\mathbf{g}_{\text{occ}} = \begin{bmatrix} \mathbf{F} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\chi} \end{bmatrix} = \mathbf{A}\boldsymbol{\xi}. \qquad (4)$$

Here $\mathbf{A}$ is a combined dictionary of warped reference images to represent blur and the $N \times N$ identity matrix to represent occlusions, where $\boldsymbol{\xi}$ is the combined weight vector, the first $|\mathcal{P}|$ elements of which represent the pose weight $\boldsymbol{\omega}$ and the remaining N elements represent the occlusion vector $\boldsymbol{\chi}$. To solve this under-determined system, we leverage the prior information about the camera motion and occlusion, viz. the sparsity of the camera motion in the pose space and the sparsity of the occlusion in the spatial domain. Thus we impose ℓ_1 norm prior on $\boldsymbol{\xi}$. We estimate the combined weight vector by solving the following optimisation problem.

$$\widetilde{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} \|\mathbf{g}_{\text{occ}} - \mathbf{A}\boldsymbol{\xi}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 \text{ subject to } \mathbf{C}\boldsymbol{\xi} \succeq 0, (5)$$

where $\mathbf{C} = \begin{bmatrix} \mathbf{I}_{|\mathcal{P}|} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$. As mentioned earlier, the occlusion vector can take both positive and negative values. To take care of this, Punnappurath et al. [15] modify the signs of the ones of the identity matrix based on the minimum and maximum elements in each row of $\mathbf{F}$. Our method is hands-off, and we neatly impose non-negativity constraint *only* on the elements of the pose weight vector. Fig. 1 demonstrates how our method works well even if the occluder intensity is close to that of background, unlike [15].

III. REGISTRATION OF VERY LARGE IMAGES

Building the matrix $\mathbf{A}$ in (4) is a crucial step in our problem. The occlusion part of the matrix $\mathbf{I}_N$ can be stored and processed efficiently since it is a diagonal matrix. The first part of the matrix $\mathbf{F}$ contains the warped versions of $\mathbf{f}$ for all the poses in $\mathcal{P}$. Though the reference image $\mathbf{f}$ operates in the intensity range [0-255] and requires only an unsigned 8-bit integer for each pixel, this is not the case for the storage of the warped versions. The pixel values of the warped image $\mathbf{f}_{\mathbf{p}_k}$ can take floating-point values due to bilinear interpolation during its generation. A round-off during the interpolation makes the equality in (2) only approximate, and hence it might lead to a wrong solution. A single warped image needs Nd bits of storage memory for operation, where d is the number of bits required to store a floating-point
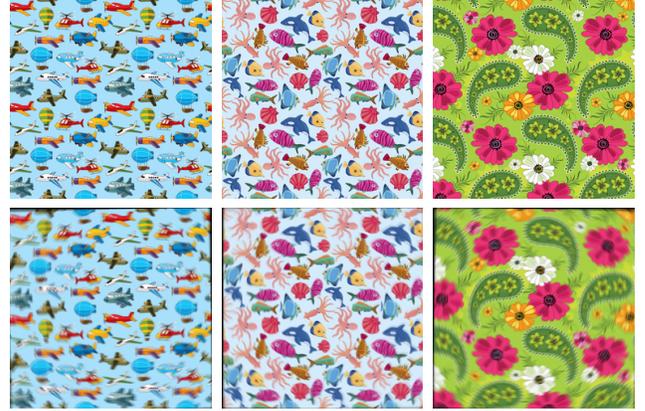
number. For even a 25 mega-pixel image with 5000 rows and 5000 columns and with $d = 32$ bits, a warped image requires $5000 \times 5000 \times 32$ bits, that is 95.3 megabytes. If all three colour channels are used, this value will triple. Storing all warps for the pose space as the matrix $\mathbf{F}$ thus warrants a huge amount of memory allocation which is infeasible in practical situations.

A. Pose Weight Estimation from Sub-images

Our solution to the large image problem stems from the interesting observation that all the pixels in an image observe the same camera motion during the exposure period. We leverage this fact to estimate the pose weight vector from a *subset* of pixels in the image. Let $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ represent a portion of the reference and blurred images, respectively. The sub-image size is $S \times S$, and $\mathbf{f}^{(S)}, \mathbf{g}^{(S)} \in \mathbb{R}^{S^2 \times 1}$. We call these, respectively, as reference sub-image and blurred sub-image. We will ignore the presence of occlusion in this discussion for clarity. The relation in (2) holds for $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ as well i.e.

$$\mathbf{g}^{(S)} = \sum_{\mathbf{p}_k \in \mathcal{P}} \omega_{\mathbf{p}_k} \mathbf{f}_{\mathbf{p}_k}^{(S)} \qquad (6)$$

The estimated pose weight vector $\boldsymbol{\omega}$ will be the same irrespective of whether we use $\mathbf{f}$ and $\mathbf{g}$ or $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ in (3). We propose to estimate the camera motion using only the sub-images thus effectively circumventing the issue of memory storage.

To verify our proposition, we now perform experiments to estimate camera motion from sub-images of large synthetically blurred images. We simulate five different continuous camera paths for a predefined set of discrete translation and rotation ranges. We use a set of five images for this experiment. We thus have a set of five reference images $\mathbf{f}$ and 25 blurred images $\mathbf{g}$. Some of the reference and blurred images are shown in Fig. 2. We pick a pair of $\mathbf{f}$ and $\mathbf{g}$, and for a given S we pick the sub-images $\mathbf{f}^{(S)}$ and
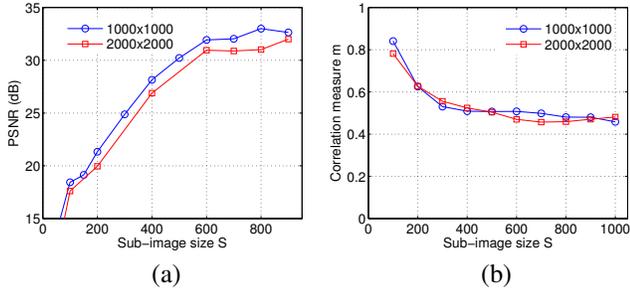
Figure 3. (a) PSNR in dB, and (b) correlation measure (for different sub-image sizes S). Original image sizes are 1000×1000 pixels (blue circle) and 2000×2000 pixels (red square).

$\mathbf{g}^{(S)}$. Using these two images, we estimate the pose weight vector $\boldsymbol{\omega}$ using (3). Since the motion involves combinations of rotations and translations, direct comparison of original and estimated motion vectors may not lead to a correct measure of error. Hence we measure the success of our estimation by reblurring. We warp $\mathbf{f}$ using the poses in $\mathcal{P}$, and perform a weighted average using the estimated weights $\widetilde{\boldsymbol{\omega}}$, resulting in a reblurred reference image. We then calculate the reconstruction PSNR of the reblurred reference image with respect to the original blurred image $\mathbf{g}$. If the motion estimation from the sub-image is correct, then the reblurred image will be close in appearance to the original blurred image resulting in a high PSNR. We repeat this experiment for different values of S. The variation of PSNR with respect to S is shown in Fig. 3(a) for image sizes of 1000×1000 pixels and 2000×2000 pixels.

For small values of S, the variation of motion blur within the sub-image will be small and will approximately tend to mimic space-invariant blur. Hence solving (3) results in a wrong pose weight estimate which results in a poor PSNR between the reblurred and blurred images. The PSNR increases as S increases since the blur variation inside the sub-image also increases. We observe that the PSNR value stabilises after a particular value of S. Beyond this point, any further increase in S results only in marginal benefits in terms of correct estimation of pose weights. The size of the sub-image is an important factor in estimating the true camera motion. Too small an S renders the notion of space-variant blur inside the sub-image invalid, and results in a wrong pose weight estimate. Too large an S will kindle storage and processing problems. In the following subsection, we formulate a method to automatically choose good sub-images for reliably estimating the camera motion.

B. Choosing a Good Sub-image

It is important to devise an automatic method to select a sub-image of a particular size at a particular location from the given large blurred observation. We develop a measure that would indicate the quality of the selected sub-image for estimating the camera motion. Given a pair of reference and blurred sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ of size S, we randomly select γ scattered locations across the image. We crop small patches, $\mathbf{f}_k^{(S)}$ and $\mathbf{g}_k^{(S)}$, from $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ respectively, for $k = 1$ to γ. We approximate the blur to be space-invariant in these patches, and estimate blur kernels using (3) allowing the pose space to contain only in-plane translations. Let us denote these blur kernels by $\mathbf{h}_k$ for $k = 1$ to γ.

If the selected sub-image has sufficient variations in blur across it, then each of these blur kernels will be different as they are quite spread out spatially. Hence a comparison of these estimated kernels is a good way to decide the suitability of the sub-image for motion estimation. We advocate the use of normalised cross-correlation of the kernels for this decision. Normalised cross-correlation between two 2D kernels $\mathbf{h}_i$ and $\mathbf{h}_j$ is given by

$$\mathbf{NCC}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{corr}(\mathbf{h}_i, \mathbf{h}_j)}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2}. \qquad (7)$$

Values of the matrix $\mathbf{NCC}$ lie in $[0, 1]$. We use the maximum value of this matrix as our measure to compare the blur kernels, i.e.,

$$\text{Correlation measure } m(\mathbf{h}_i, \mathbf{h}_j) = \max \; \mathbf{NCC}(\mathbf{h}_i, \mathbf{h}_j) \quad (8)$$

Note that $m(\mathbf{h}_i, \mathbf{h}_j)$ attains a peak value of 1 if the two blur kernels are same. If the sub-image size is small, then there will not be sufficient blur variations across it, and our measure value will be close to 1. If the kernels are dissimilar, then m takes values close to 0.

Fig. 4 shows four blur kernels of the patches that are extracted randomly from sub-images of sizes $S = 100, 300$ and 600. The patch size used is 41×41 and $\gamma = 4$. Blur kernels corresponding to space-invariant blur will appear the same irrespective of the spatial point. For a small sub-image of size $S = 100$, it can be clearly observed that the four kernels are similar. Hence the camera motion cannot be correctly explained by this sub-image. For $S = 300$, the blur kernels are more dissimilar, and for $S = 600$, they look completely different. Thus, higher values of S describe the motion better. From these four blur kernels, $\binom{4}{2}$ (six) measure values m are estimated for every pair. Fig. 3 (b) shows the plot of mean $\overline{m}$ of these six values with respect to the sub-image size. The curve falls with increase in sub-image size as expected due to continuous decrease in kernel similarity. A synonymity can be observed between the plots in Figs. 3 (a) and (b). Correlation measure (Fig. 3 (b)) decreases initially with increasing S and stays almost constant after a certain value of S. Similarly, the reconstruction PSNR (Fig. 3 (a)) stabilises after it reaches a particular sub-image size. Based on these observations, we define a threshold $T_m = 0.6 \, \overline{m}_{100}$, where $\overline{m}_{100}$ is the correlation measure for $S = 100$, to accept or reject a sub-image for motion estimation. If $\overline{m}_{S_0}$ for a sub-image of a specific size S_0 is less than this threshold, we decide that
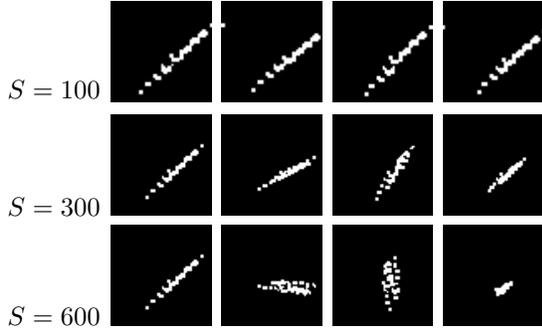
Figure 4. Estimated blur kernels for different sub-image sizes S. The blur kernels are displayed as binary images with non-zero values shown in white colour.
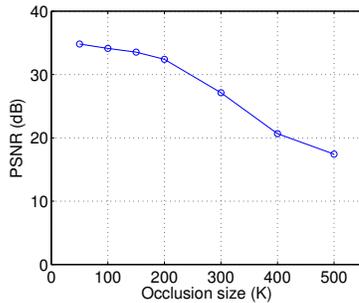

Figure 5. (a) PSNR in dB for $S = 600$ and different occlusion sizes K.

the quality of the selected sub-image of size S_0 is good, and that the camera motion can be estimated from it.

C. Presence of Occlusion

A natural question to ask is how well our algorithm fares when there is occlusion in the selected sub-image itself. We add a random occlusion patch of size $K \times K$ to the reference image $\mathbf{f}$. We blur this image using the generated camera motion path, the resultant image being $\mathbf{g}_{occ}$. We slice the sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}_{occ}^{(S)}$, from $\mathbf{f}$ and $\mathbf{g}_{occ}$ respectively. We do not restrict the position of the sub-image with respect to the occlusion. Therefore, the sub-image can either include the full occlusion or a part of the occlusion or be devoid of the occlusion completely. Our combined dictionary $\mathbf{A}$ in (4) tackles both the presence of blur and occlusion simultaneously. If occlusion is present either fully or partially, it would be accommodated by the weights of the identity matrix in $\mathbf{A}$. If there is no occlusion present, then the occlusion weight vector will be zero. Thus, irrespective of the absence or presence (complete or partial) of the occluder in the sub-image, our formulation can elegantly handle it.

We next discuss the effect of the size of the occlusion for a chosen sub-image size S. We consider the worst case of the occlusion being present completely inside the chosen sub-image.

We solve the optimisation problem in (5) with $\mathbf{f}^{(S)}$ and $\mathbf{g}_{occ}^{(S)}$ to arrive at the combined pose weight and occlusion

Algorithm 1:

Inputs: Reference image $\mathbf{f}$, blurred and occluded image $\mathbf{g}$.

Init: Pick β sub-images $\mathbf{f}^{(100)}$ based on Hu et al. [16]. Extract γ blur kernels from each sub-image, calculate $\overline{m}$ for $\binom{\gamma}{2}$ pairs, and find the mean $\overline{m}$. Average the β correlation values to get $\overline{m}_{100}$.

Let $S = 200$. leftmargin=*

1) Pick a sub-image of size S. Extract γ blur kernels. Calculate $\overline{m}_S$ by averaging out the correlation values found out for all kernel pairs. If $\overline{m}_S < 0.6\ \overline{m}_{100}$, goto Step 4. Else, choose a different sub-image of the same size at a different location.

2) If a particular S is chosen α times at different locations, declare this S to be unsuitable to estimate motion. Update $S \leftarrow S + 100$. Goto Step 1.

3) If $S > S_{\max}$, declare blur to be space-invariant. Use one of the estimated blur kernels itself as the camera pose weight vector. Goto Step 5.

4) Estimate pose weight vector and occlusion weight vector for the selected sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ using (5). If the number of non-zero elements in the occlusion weight vector $\|\boldsymbol{\chi}\|_0 > \frac{2}{3}S^2$, then go to Step 1.

5) Reblur the original reference image $\mathbf{f}$ using the estimated pose weight vector $\widetilde{\boldsymbol{\omega}}$.

6) Detect the changes by differencing the reblurred image and the observation $\mathbf{g}$.

weight vectors. Using the estimated $\widetilde{\boldsymbol{\omega}}$, we reblur the large reference image $\mathbf{f}$. We compare this reblurred image with the large blurred image $\mathbf{g}_{occ}$ ignoring the values in the occlusion region, since this comparison is to verify the success of our motion estimation. Fig. 5 shows how the PSNR varies with respect to the value of K for $S = 600$. We note that our algorithm tackles the presence of occlusion quite well. The motion estimation is correct, and thus, PSNR values are good even when the occluder occupies up to half the sub-image area.

Algorithm 1 shows our complete framework to choose good sub-images automatically, estimation of motion and change detection. In our experiments, we use the following values in Algorithm 1: $\alpha = 5, \beta = 6, \gamma = 4, S_{\max} = 900$. Eq. (5) requires a sparse matrix-vector multiplication with order less than $O(N(S^2 + |\mathcal{P}|))$ and a projection onto subspace $O(S^2 + |\mathcal{P}|)$. We use SLEP package [17], which has a convergence rate of $O(1/k^2)$ for the kth iteration.

IV. EXPERIMENTS

We first evaluate the performance of our algorithm using a synthetic example. A reference image of size 2188×1315 pixels is shown in Fig. 6(a). To simulate blur incurred due to camera shake, we manually generated camera motion with

a connected path in the 6D pose space and initialised the weights. The synthesised camera motion was then applied on the same scene taken from a different view point with synthetically added occluders to produce the blurred and occluded image in Fig. 6(b).

To evaluate the proposed method, we followed the steps outlined in Algorithm 1, selected four sub-images (based on Hu et al. [16]) of size 100×100 pixels, and calculated $\overline{m}$ independently for each sub-image. The average value, $\overline{m}_{100}$, was computed. Next, we picked a sub-image of size 200×200 pixels and calculated $\overline{m}_{200}$. The four kernels computed within the sub-image bore a large degree of similarity indicating that the space-varying nature of the blur was not being captured at this size. This step was repeated for five different sub-images of size 200×200 pixels, but the values of $\overline{m}_{200}$ they yielded were approximately equal to $\overline{m}_{100}$ revealing that only a bigger sized sub-image can encapsulate the space-varying camera motion. Our algorithm converged for a sub-image of size 500×500 pixels where the computed $\overline{m}_{500}$ was less than $0.6\ \overline{m}_{100}$. The selected sub-images of size 500×500 pixels from the focused, and the blurred and occluded input images are shown in Figs. 6(c) and 6(d), respectively. The position of these sub-images have been indicated by red boxes in Figs. 6(a) and (b). Note that the selected sub-image, incidentally, does not contain any occlusion. To handle pose changes between the two images, we first coarsely aligned the reference image and the blurred and occluded image at a lower resolution using a multiscale implementation similar to [15]. As noted by Gupta et al. [10] and Whyte et al. [11], 6D camera motion with small out-of-plane rotations can be explained by a 3D pose space itself. Hence, we use the following ranges while estimating the motion: in-plane translations: $[-12{:}1{:}12]$ pixels, in-plane rotations: $[-3°{:}0.5°{:}3°]$. Fig. 6(e) shows the reference image reblurred using the estimated $\widetilde{\omega}$. The detected occlusions shown in Fig. 6(f) are found by comparing the blurred and occluded observation (Fig. 6(b)) with the reblurred reference image (Fig. 6(e)).

Next, we demonstrate the applicability of the proposed method on real images. The roof-top images in Figs. 7(a) and (b) which represent the reference image and the blurred and occluded observation, respectively, were captured using an 18 MP Canon DSLR camera. Observe that in addition to significant blur and change in viewpoint, the occlusion (a cyclist in this case) has its own independent motion. While the combined motion of camera and object would confound blind deblurring algorithms, our joint formulation for estimating camera motion and occlusion can elegantly handle this challenging scenario even when the sub-image includes the occluder. The detected occlusion is shown in Fig. 7(c) which is correct.

For our next experiment, we use the publicly available VIRAT database (Oh et al. [18]) which is a benchmark for video surveillance and change detection. Two frames
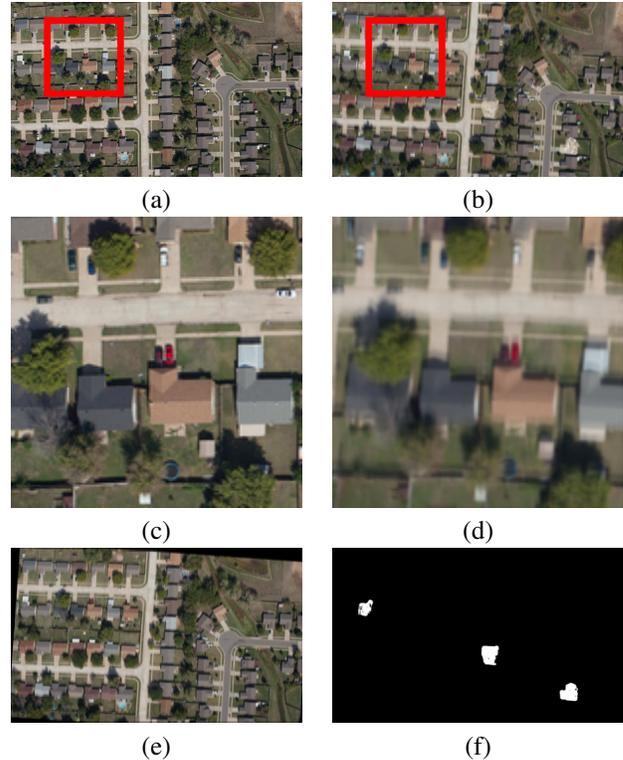

Figure 6. (a) Reference image, (b) synthetically blurred and occluded observation from a different view point, (c) sub-image from (a), (d) sub-image from (b), (e) reference image reblurred using the estimated camera motion, and (f) detected occlusion.

corresponding to the reference image and the occluded image (shown in Figs. 8(a) and (b), respectively) were manually extracted from an aerial video. The frames are at the resolution of the original video i.e., 720×480 pixels. Since the resolution is low, we run our algorithm directly on the whole image instead of a sub-image. The detected occlusion is shown in Fig. 8(c). Although, strictly speaking, the images are not high resolution at all, the purpose is to demonstrate the efficiency of our method for aerial imaging. Also, this example illustrates how the proposed method elegantly subsumes the work in [15] for the case of low resolution images.

A final real example is shown in Fig. 9. The two images in Figs. 9 (a) and (b) were captured from the same view point but with a small time lapse using a Google Nexus 4 mobile phone which has an 8 MP camera. Observe how even small occluders with intensities close to the background are correctly detected by our algorithm (Fig. 9(c)). This example threw up small spurious non-connected occlusions in the bottom half of the image due to movement of leaves, and these were removed by simple post-processing.

To perform quantitative assessment of our method, we computed the following metrics which are well known in the change detection community: percentage of correct classification (PCC), Jaccard coefficient (JC) and Yule coefficient (YC) [19]. For the real experiments, the ground-truth

| (a) | (b) | (c) |

Figure 7.   (a) Reference image, (b) real blurred and occluded observation, and (c) detected dynamic occluder.

| (a) | (b) | (c) |

Figure 8.   (a) Reference image, (b) real blurred and occluded observation, and (c) detected occlusion.

occlusion was assessed by asking different individuals to locally mark out the occluded regions as per their individual perception. The efficacy of our algorithm is further evident from the values in Table I. A value greater than 0.5 is considered good for JC and YC, and a value close to 100 indicates high performance for PCC.

Table I
QUANTITATIVE METRICS FOR OUR RESULTS IN FIGS. 5 TO 8.

Fig.	PCC	JC	YC
5	99.7874	0.8508	0.9443
6	99.9409	0.7816	0.8377
7	99.7195	0.6218	0.9211
8	99.3736	0.4557	0.8729

The maximum size of the images considered in this paper was 18 MP. Despite our best attempts, we could not find any database containing image pairs of very large sizes (of the order of 100M) that could be used for testing. Nevertheless, the framework proposed here has the potential to handle even very large images. Due to file size constraints, we have included only the downscaled images in the pdf, and not the original high resolution images.

V. CONCLUSIONS

We proposed a framework to detect changes in blurred images of very large sizes. Traditional deblurring techniques fail to cope with large image sizes, while feature-based approaches for change detection are rendered invalid in the presence of blur. We developed an optimisation problem which would perform registration in the presence of blur and detect occlusions simultaneously. We devised an algorithm to choose good sub-images from the large observations to estimate the camera motion thus alleviating issues related to memory and computational resources. As future work, it would be interesting to relax the constraints of using images taken with the same camera with same resolution and focus settings, by assimilating ideas from areas such as domain adaptation. We shall also consider accommodating illumination variations into our framework.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
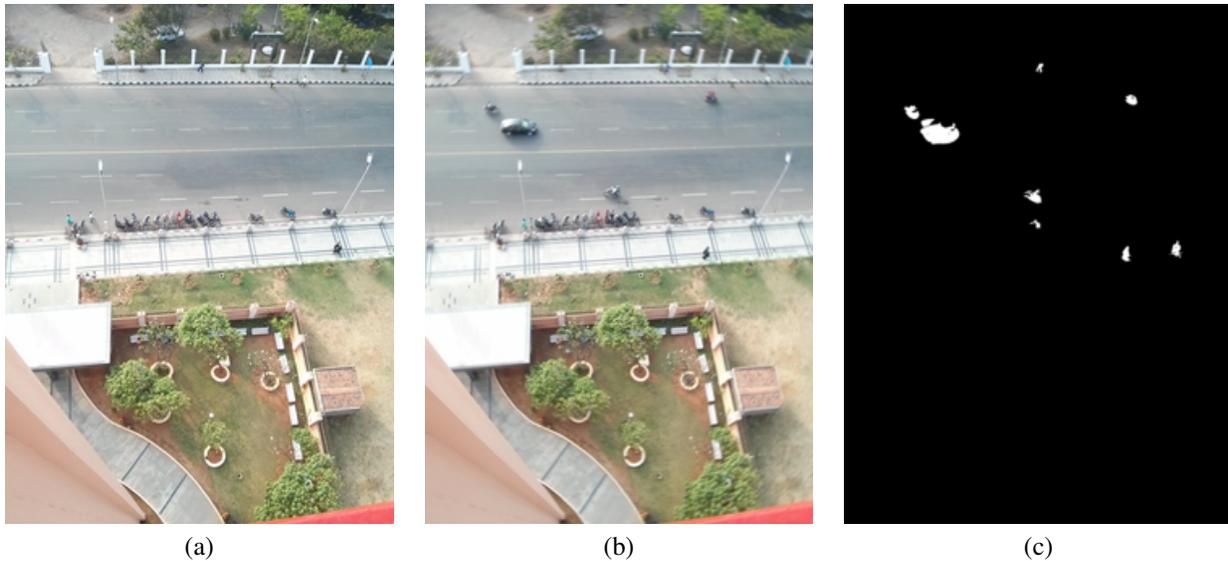
<div align="center">(a)   (b)   (c)</div>

Figure 9. (a) Reference image, (b) real blurred and occluded observation, and (c) multiple occluders found.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Proc. ICCV*. IEEE, 2011, pp. 2564–2571.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[5] C. Huo, C. Pan, L. Huo, and Z. Zhou, "Multilevel sift matching for large-size vhr image registration," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 2, pp. 171–175, 2012.

[6] A. Carleer, O. Debeir, and E. Wolff, "Assessment of very high spatial resolution satellite image segmentations," *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 11, pp. 1285–1294, 2005.

[7] L. Yu, D. Zhang, and E.-J. Holden, "A fast and fully automatic registration approach based on point features for multi-source remote-sensing images," *Computers & Geosciences*, vol. 34, no. 7, pp. 838–848, 2008.

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

[9] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. ECCV*, 2010, pp. 157–170.

[10] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. ECCV*, 2010, pp. 171–184.

[11] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012.

[12] Z. Hu and M.-H. Yang, "Fast non-uniform deblurring using constrained camera pose subspace." in *Proc. BMVC*, 2012, pp. 1–11.

[13] C. Paramanand and A. N. Rajagopalan, "Non-uniform motion deblurring for bilayer scenes," in *Proc. CVPR*. IEEE, 2013, pp. 1115–1122.

[14] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proc. CVPR*, 2013, pp. 1107–1114.

[15] A. Punnappurath, A. Rajagopalan, and G. Seetharaman, "Registration and occlusion detection in motion blur," in *Proc. ICIP*, 2013.

[16] Z. Hu and M.-H. Yang, "Good regions to deblur," in *Proc. ECCV 2012*. Springer, 2012, pp. 59–72.

[17] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/ jye02/Software/SLEP

[18] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. CVPR*. IEEE, 2011, pp. 3153–3160.

[19] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, 2005.