Blurred/Non-Blurred Image Alignment using Sparseness Prior

Lu Yuan[1] Jian Sun[2] Long Quan[1] Heung-Yeung Shum[2]

[1]The Hong Kong University of Science and Technology [2]Microsoft Research Asia

Abstract Aligning a pair of blurred and non-blurred images is a prerequisite for many image and video restoration and graphics applications. The traditional alignment methods such as direct and feature-based approaches cannot be used due to the presence of motion blur in one image of the pair. In this paper, we present an effective and accurate alignment approach for a blurred/non-blurred image pair. We exploit a statistical characteristic of the real blur kernel - the marginal distribution of kernel value is sparse. Using this sparseness prior, we can search the best alignment which produces the sparsest blur kernel. The search is carried out in scale space with a coarse-to-fine strategy for efficiency. Finally, we demonstrate the effectiveness of our algorithm for image deblurring, video restoration, and image matting.

1. Introduction

Image alignment or registration is a fundamental task for many multi-image and video applications, e.g., image stabilization, image enhancement, video summarization, panorama and satellite photo stitching, medical imaging, and many graphics applications. However, existing methods are applied only to *good* images without motion blur. In this paper, we study the problem of aligning two images, one blurred and one non-blurred, as illustrated in Figure 2. The problem arises in many practical capturing situations, for instance changing relative motion between the camera and the scene, fast panning of the video camera which often gives blurred frames, and varying the exposure times [7, 22] for hand-held camera in low-light conditions.

Aligning a blurred/no-blurred image pair is non-trivial. For a spatially invariant blur, the blurred image can be represented as a convolution of a blur kernel and an original sharp image. Usually, the real blur kernel is complex, not simply a gaussian or a single direction motion. The presence of the blur make it difficult to directly apply two existing types of image alignment approaches: direct approaches [13] and feature based approaches [2]. Direct approaches minimize pixel-to-pixel dissimilarities. But this measurement is infeasible if the blur kernel is large, e.g., 40-80 pixels. One can downsample the input images to reduce the blur effect, but in practice it is hard to use more than two or three levels of a pyramid before important details start to be lost. Feature based approaches have trouble extracting features in the blurred image. For an arbitrary blur kernel, features such as corners, or SIFT features [12], are not blur invariant.

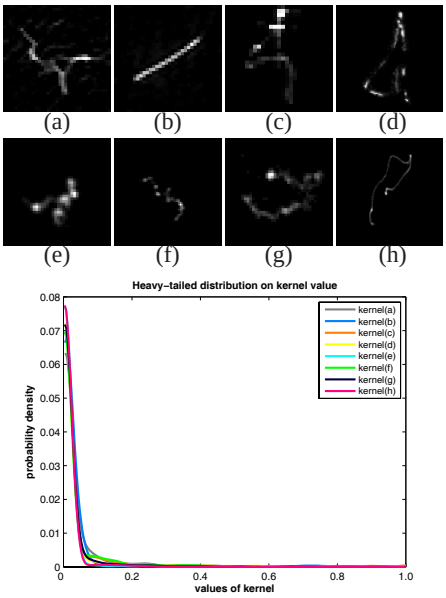


Figure 1. Kernel distributions. Top: eight real kernels. (a-c) are from Fergus et al. [4], and (d) is from Raskar et al. [15], and (e-h) are from Anonymous [22]. Bottom: The histograms of kernel magnitude are shown in different colored curves.

The main difficulty is that we do not know the blur kernel or the motion between the two images. If the blur kernel is known, we can do non-blind deconvolution to obtain a deblurred image to apply the previous approaches to. However, directly estimating an accurate blur kernel from the blurred image is challenging despite recent significant progress in single image deblurring [4]. If two images are well aligned (up to a translation), the work in [22] demonstrated that a very accurate kernel can be estimated from a blurred/non-blurred image pair.

The key is whether it is possible to align a blurred and non-blurred image pair without correspondence. If so, what is the necessary prior information, and what are the required assumptions?

1.1. Related work

A general tutorial to the literature of image alignment could be found in [19]. There are two approaches explicitly account for blur. The first approach is to deblur. By limiting the blur kernel to be one dimensional in [16, 17], both align-

ment and deblurring can be solved simultaneously for a blurred/non-blurred pair or two blurred images with different blur directions. The motion direction is pre-computed using the monotonicity of the sharpness of the recovered image as a function of the estimated direction. The work in [1] assumes that the motion blur is caused by a constant velocity affine motion, which is estimated by a tracking that combines area-based and contour based deformable models. Motion blur has also been explicitly taken into account in [8]. Based on single direction and constant velocity assumption, it simultaneously estimates motion blur and deformation. Instead of deblurring images, the matching cost is defined by further blurring two regions using commutativity of the kernels.

The second approach is to derive features that are invariant to blur or invariant to the combination of both blur and geometric transforms. The work in [6] proposes the first moment-based invariants to blurring by a centrally symmetric blur kernel. The descriptors are invariant to translation. This work is extended in [6] and [5] to moment-based descriptors that are invariant to the combination of 2D blur-rotation and N-D blur-rotation. However, these invariants are mainly limited to centro-symmeric kernels, e.g., Gaussian-like blur, out-of-focus blur, and atmospheric turbulence blur. So far, there is no blur-invariant feature for an arbitrary shape kernel.

1.2. Our approach

We propose a simple but effective alignment method that uses the statistics of the blur kernel. We assume that the blur kernel is spatially invariant, caused by motion between the camera and the scene. We exploit an unique characteristic of most real blur kernels: sparseness, i.e., most values in the kernel are zero. Blur kernels have this characteristic because the relative motion between camera and scene is a continuous path in 3D in most cases. This is particularly true for the hand-held cameras. If there is no saturation, the trajectory of a point light source recorded on the 2D sensor should be exactly the blur kernel. Based on this simple statistic, we can use the kernel sparseness priors to measure what a good kernel should be. Thus, we can simultaneously perform the alignment and the blur kernel estimation by a coarse-to-fine "brute force" search.

2. Kernel Sparseness Prior

Natural image statistics have been used in many applications, such as image super-resolution [20], denoising [18], inpainting [11], transparency separation [10], and single image deblurring [9]. These works mainly exploit the sparseness of the distribution of image derivatives. Inspired by these works, we propose to study the statistics of the blur kernels, and to use these statistics as the prior for alignment with a blurred image.

2.1. Kernel statistics

We collected a number of real kernels, as shown in Figure 1 (a-g). The brightness value in the kernel suggests the duration of the camera exposure. The brighter a kernel value is, the longer the camera exposes. As we can see, all kernels tend to be sparse - most values in the kernel are zeros (black) and the non-zeros form curve-like pathes. Figure 1 (b) plots the histograms of the kernel values. The histograms peak at zero and fall off faster than a Gaussian distribution. The long tails show the high-kurtosis statistics of these distributions. Following [4], we fit the kernel value distribution to a mixture of two exponential distributions:

$$p(k_i) \propto \left(w_1 e^{-k_i/\beta_1} + (1-w_1) e^{-k_i/\beta_2} \right), \qquad (1)$$

where k_i is the ith element in the kernel $\mathbf{k}$, and w_1 and β_1, β_2 are parameters of two exponential distributions. Assuming i.i.d, the probabilistic distribution of the kernel $\mathbf{k}$ is $P(\mathbf{k}) = \prod_i p(k_i)$. To measure the sparseness of a given kernel, we compute the minus log of distribution $P(\mathbf{k})$ which is normalized by the kernel size Z as a sparseness measurement:

$$E_{sparse}(\mathbf{k}) = -\frac{1}{Z} \sum_i \ln p(k_i). \qquad (2)$$

The parameters in the distribution $P(\mathbf{k})$ can be estimated using example kernels by Maximum Likelihood (ML) estimation. Using eight kernels in Figure 1, our estimated parameters are: $w_1 = 0.6, \beta_1 = 0.01, \beta_2 = 0.03$.

2.2. Kernel estimation

The work in [22] demonstrated that if the blurred image and non-blurred image are well aligned, a very accurate blur kernel can be estimated using two input images. We briefly review the approach.

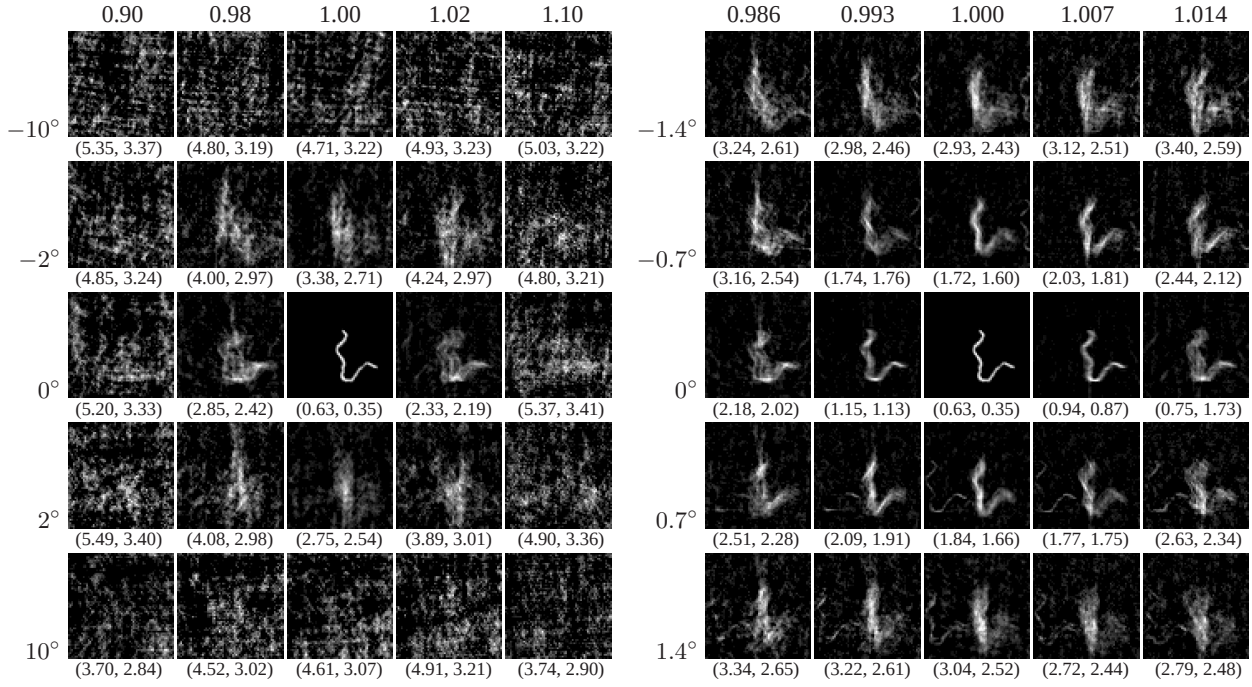
First, the non-blurred image $\mathbf{I}$ is pre-multiplied by a scale factor, if necessary, to compensate for the average intensity difference between two images. If two images are aligned up to a translation (because the image translation only results in a translation of the kernel), the blurred image $\mathbf{B}$ is the convolution of the image $\mathbf{I}$ and the blur kernel $\mathbf{k}$: $\mathbf{B} = \mathbf{I} \otimes \mathbf{k}$, where $\otimes$ is the convolution operator. Then, the kernel $\mathbf{k}$ is computed by minimizing the following objective function:

$$\mathbf{k}^* = \arg\min_{\mathbf{k}} ||\mathbf{B} - \mathbf{I} \otimes \mathbf{k}||^2 + \lambda ||\mathbf{k}||^2, \quad \text{subject to } k_i \geq 0, \text{ and } \sum_i k_i = 1, \qquad (3)$$

where $\sum_i ||k_i||^2$ is a Tikhonov regularization term to stabilize the solution and the parameter λ is fixed at 3. A Landweber method [3] is used for the optimization. The algorithm is fast when using FFT, taking about 8 to 12 seconds for a 64×64 kernel and a 800×600 image. To make

(a) blurred image

(b) no-blurred image



(c) coarse sampling

(d) fine sampling

Figure 2. A synthetic example. (a) and (b) the blurred and non-blurred images. (c-d) estimated kernels using the blurred image and a number of transformed non-blurred images, with different rotations and scales. $[0.90, 0.98, 1.00, 1.02, 1.10]$ (in horizontal) $[-10^\circ, -2^\circ, 0^\circ, 2^\circ, 10^\circ]$ (in vertical) are the sampled scales and rotations in the coarse level. Two numbers (in brackets) under each kernel are the sparseness value and the entropy. The kernel sparseness and entropy are good measures of alignment accuracy.

the optimization robust, the kernel estimation is run in a coarse-to-fine manner. To further suppress the error and noise in the estimation, a hysteresis thresholding is applied in scale space so that the global shape of the kernel at a fine level to be similar to the shape at its coarser level. More details can be found in [22].

2.3. Empirical evaluation of kernel sparseness

The kernel estimation method given above is theoretically optimal only when the two images are aligned up to a translation. Unfortunately, this is not true in practice. Let us first assume that the motion between the two images is a similarity transformation: $\mathbf{x}' = [s\mathbf{R} \quad \mathbf{t}]\ \mathbf{x}$, where $\mathbf{x} = (x, y, 1)$ is the *homogeneous* 2D coordinate. $\mathbf{R}$ is an orthonormal rotation matrix, s is an arbitrary scale factor, and $\mathbf{t}$ is 2D translation. Since an image translation only results in a translation of the kernel and our kernel sparseness measurement is translation-invariant, we need not consider the 2D translation $\mathbf{t}$. Therefore, the alignment parameter space consists of only a 2D rotation characterized by an angle and a scaling factor for the similarity transformation.

Now, we want to investigate the relationship between the sparseness of the estimated kernel and the accuracy of the kernel estimation under different similarity transformations between the blurred/non-blurred image pair.

We designed the following experiment. First, we take one original (non-blurred) image and blur it with a given synthetic kernel (shown on the top-right corner of Figure 2 (a)) to obtain the blurred image shown in Figure 2. Then, we transform the original non-blurred image using different

rotation angles and scaling factors to generate a series of transformed but non-blurred image families. Finally, we run the kernel estimation method described in the above section for each pair of blurred and transformed non-blurred images. Figure 2 (c-d) show estimated kernels with different transformations at coarse and fine sampling levels.

It is interesting to observe that the closer the alignment parameters are to the true transformation, the sparser the estimated kernel is. The true alignment $(0^\circ, 1.000)$ produces the sparsest kernel! The sparseness E_{sparse} for each kernel is quantitatively evaluated by Equation (2), and is shown in the first number in the bracket on the bottom of each kernel in Figure 2. The best kernel has the smallest sparseness measure of 0.63. The measurement E_{sparse} consistently reveals the sparseness of the kernel.

An alternative measure for the sparseness of a distribution $p(x)$ is to use the entropy: $-\int_x p(x) \ln p(x)$. In Figure 2, the second number in the bracket is the entropy of the histogram of the kernel values. The entropy also tells which kernel we should choose since it measures the disorder or randomness of a distribution. The sparsest kernel has the minimal entropy. Both the sparseness value and the entropy are good measurements for the sparseness characteristic of a real kernel.

3. Alignment using Sparseness Prior

Motivated by the above empirical results, we propose an alignment approach using the kernel sparseness prior. We first describe the algorithm with a similarity transform between the two images, then we evaluate the alignment accuracy, and finally we extend the similarity transform to the affine transform.

3.1. Algorithm

To be efficient, we use scale space representations for both image space and parameter space. We build two pyramids $\{\mathbf{B}^l\}_{l=1}^L$ and $\{\mathbf{I}^l\}_{l=1}^L$ for the blurred and non-blurred images, where L is the pyramid level. Let $[\Delta\theta, \Delta s]$ and $[\theta_-, \theta_+]$, $[s_-, s_+]$ be the search intervals and ranges for the rotation angle and the scale. Our basic algorithm for the similarity transformation is from level 1 to L:

- compute a family of kernels using $\mathbf{B}^l$ and the transformed $\mathbf{I}^l$ from Eqn. (3) in the current search ranges. Calculate the sparseness values by Eqn. (1) or entropies of these kernels.
- obtain the best alignment $[\theta_{opt}, s_{opt}]$ resulting in the kernel with minimal sparseness value E_{sparse} or entropy.
- (optional) repeat the above steps once or twice by reducing the search ranges and sampling intervals by half.
- for the next level $l+1$, set the sampling intervals as $[\Delta\theta/2, \Delta s/2]$, and the search ranges as $[\theta_{opt} \pm 2\Delta\theta] \times [s_{opt} \pm 2\Delta s]$.

The initial search ranges are determined by the maximal possible relative motion between two images and the initial sampling intervals are dependent on the required alignment accuracy. Figure 4 shows a real example using four levels coarse-to-fine search. Figure 4 (a) is a blurred image with a long exposure and Figure 4 (b) is a non-blurred but noisy image with a short exposure. Their size is 791×1156. In this example, we set the initial search ranges to $[-45^\circ, 45^\circ] \times [0.7, 1.3]$ and the initial sampling intervals to 1° and 0.04.

The top of Figure 4 shows a fraction of the kernels estimated during the search. For each kernel, the rotation angle and the scale are on the top, and the sparseness value and the entropy are on the bottom. We mark the sparsest kernel at each level with a red square. With the estimated blur kernel, we can obtain the deblurred image in Figure 4(c) using the Richardson-Lucy (RL) deconvolution algorithm [14] (Matlab's deconvblind routine). The ripple-like ringing effects in the deblurred image is a common side effect of most deconvolution approaches. The fundamental reason is that the partial high frequencies in the blurred image lost. Please see [22] for more discussion. Figure 4(d) is the aligned image using the computed rotation and translation. The images in Figure 4 (c-d) are automatically aligned and we do not need to consider translation. The total computation time are about 7 minutes, which is efficient with the finest sampling intervals of 0.02° and 0.001.

3.2. Alignment accuracy

The accuracy of the estimates can be evaluated for the synthetic examples as we know the ground truth. For the example in Figure 2, we search for the sparsest kernel using very fine search intervals $[0.001^\circ, 0.0001]$ at the finest level to test the maximal accuracy that can be achieved by our approach. The resulting alignment parameters are $[-0.009^\circ, 1.0002]$ using the sparseness value, and $[-0.013^\circ, 1.0004]$ using the minimal entropy, with respect to the ground truth parameters $[0.0^\circ, 1.0]$. For this 678×376 example, the alignment accuracy is sub-pixel.

To measure the accuracy for the real example in Figure 4, we manually select a number of corresponding features in the deblurred image and the aligned non-blurred image, as showed in Figure 3 (a). The corresponding features are further refined to sub-pixel accuracy by correlation [21]. The alignment error is measured by the coordinate differences between corresponding feature pairs. Figure 3 (b) shows the coordinate differences (Δx, Δy) for 11 corresponding feature pairs. We have also implemented a naive alignment approach: first deblur the blurred image using Fergus et al.'s single image deblurring algorithm [4] (http://people.csail.mit.edu/fergus/), then use the SIFT [12]

#	Ours (pixel)	SIFT (pixel)
1	(0.6, 0.2)	(5.7, 6.8)
2	(0.0, -0.3)	(5.9, 1.2)
3	(0.2, -0.4)	(3.1, -3.9)
4	(0.5, -0.8)	(-7.7, -12.5)
5	(-0.1, -0.9)	(4.6, 7.9)
6	(0.1, -0.6)	(7.0, 6.5)
7	(-0.5, -0.7)	(-2, -2.7)
8	(0.2, -0.6)	(0.1, 2.0)
9	(0.7, -0.5)	(0.8, 13.9)
10	(-0.3, -0.3)	(2.2, 2.3)
11	(0.1, 0.2)	(5.9, 6.9)

Figure 3. Comparison of alignment errors. (a) 11 manually marked features in our deblurred image. (b) alignment errors of our approach and a naive approach (single image deblurring [4] + SIFT [12]). Our alignment is significantly more accurate.

algorithm with RANSAC to align the deblurred image and the non-blurred image. In Figure 3(b), we can see that our alignment errors are less that a pixel. The mean and standard derivation of our alignment error are $0.31, 0.50$ and $(0.23, 0.24)$ which are very small compared with the native approach, which produces the mean $(4.13, 6.11)$ and the standard derivation $(2.58, 4.22)$. The errors of the naive approach are quit large and unacceptable for most alignment applications.

3.3. Extension for affine transformation

To model an affine transformation, we replace the scaled rotation matrix by a non-singular linear transformation followed by a translation: $\mathbf{x}' = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{x}$. The 2×2 matrix $\mathbf{A}$ is the composition of two fundamental transformations, rotations and non-isotropic scalings. The matrix $\mathbf{A}$ can always be decomposed as $\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi)$, where $\mathbf{R}(\theta)$ and $\mathbf{R}(\phi)$ are rotations by θ and ϕ respectively, and $\mathbf{D}$ is a diagonal matrix $diag(s_1, s_2)$. This decomposition follows directly from the SVD: writing $\mathbf{A} = \left(\mathbf{U}\mathbf{V}^{\mathrm{T}}\right)\left(\mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}\right) = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi)$, since $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. Therefore, alignment with an affine transformation requires a search in a 4D parameter space.

To reduce the search range, we first estimate an initial affine matrix using a SIFT based alignment with RANSAC at low-resolution image pairs. Although the initial alignment could be very inaccurate, it provides a reasonable starting point. Then, we apply our coarse-to-fine algorithm to search four physical parameters $[\theta, \phi, s_1, s_2]$.

Figure 5 shows an affine example. For this example, the search intervals in four levels are: ($\Delta\theta = \Delta\phi$ = 1.5, 0.32, 0.08, 0.02), ($\Delta s_1 = \Delta s_2$ = 0.06, 0.016, 0.004, 0.001). The top row of Figure 5 shows estimated kernels in different levels. The four parameters associated with the sparsest kernel are: $[\theta, \phi, s_1, s_2] = [-9.82°, -23°, 1.063, 0.926]$. The final matrix $\mathbf{A} = \begin{bmatrix} 1.0352 & 0.2102 \\ -0.1292 & 0.9246 \end{bmatrix}$. Figure 5 (c-d) are the deblurred image and the aligned image using our estimated kernel and affine transformation. The mean and standard derivation are (0.6, 0.8) and (0.39, 0.45). The computation time is about 40 minutes for this example. Although searching in 4D is expensive, it is still affordable with our coarse-to-fine search in both image space and parameter space.

4. Applications

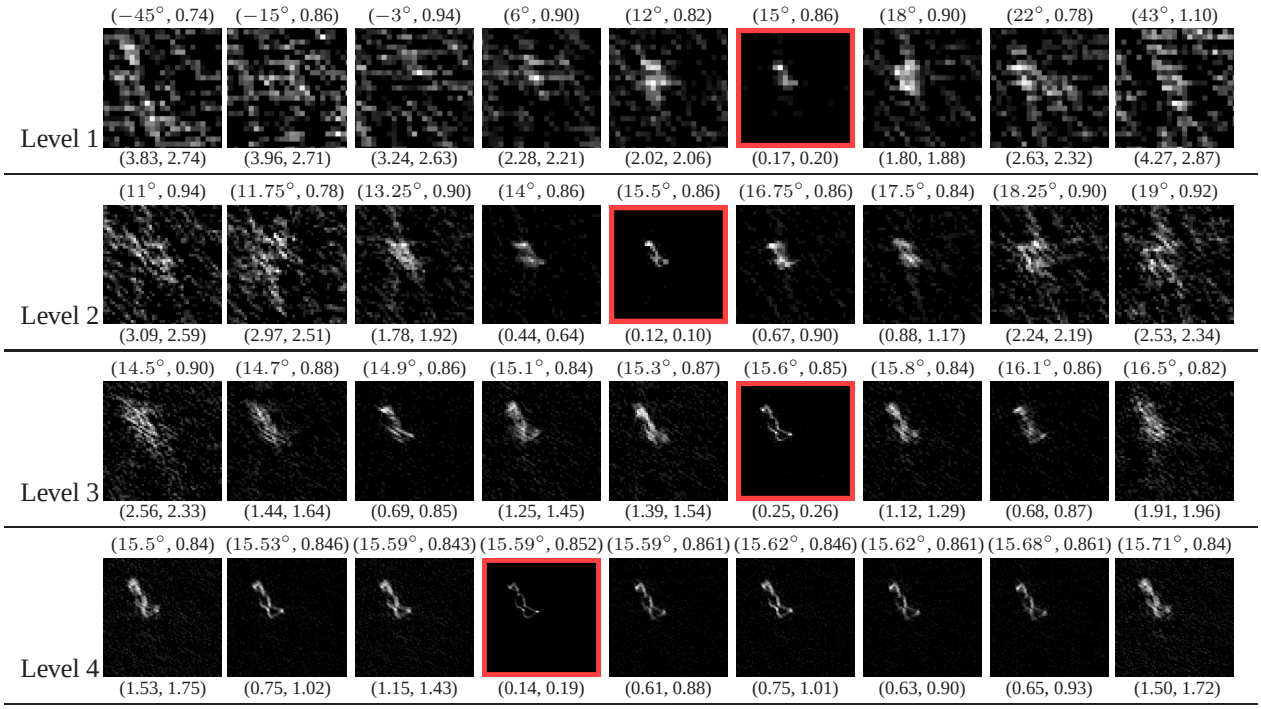
We apply our approach to three applications requiring alignment of blurred/non-blurred image pair.

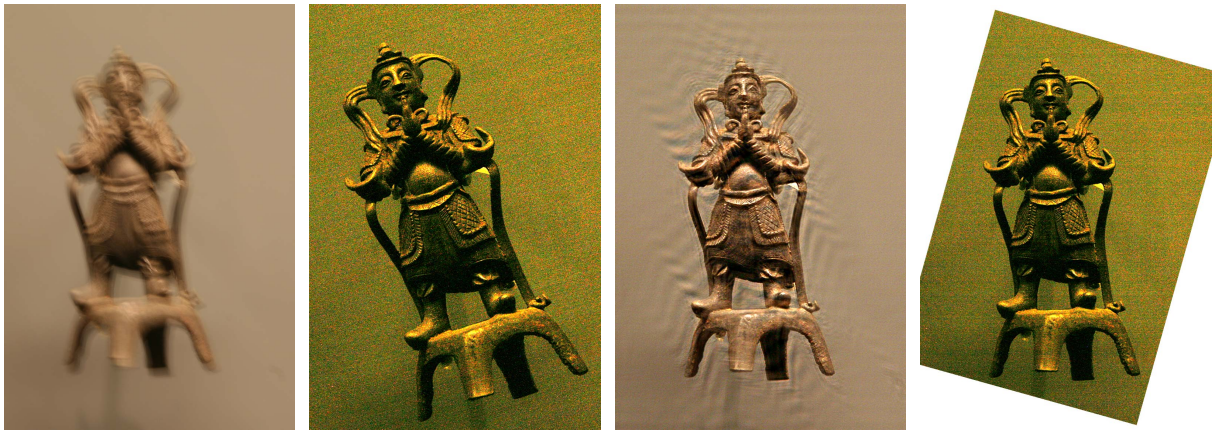
Image deblurring. Taking satisfactory photos under low-light condition using a hand-held camera is challenging. [22] presented an image deblurring approach by combining a blurred/nosiy image pair. The blurred image is taken with a slow shutter speed and a low ISO setting. It has the correct color, intensity and a high SNR, but is blurry due to camera shake. The other is an underexposed and noisy image with a fast shutter speed and a high ISO setting. It is sharp but very noisy due to insufficient exposure and high camera gain.

Impressive deblurring results can be obtained using the approach proposed in the work [22], using two well aligned images. With our alignment approach, we can relax this restriction and make two-image deblurring more practical. Figure 6 shows a real image deblurring example.

Video restoration. When the user pans a video camera with varying speed, blurred/non-blurred image pairs can occur. The top row of Figure 7 shows three frames in a video. The middle one is a non-blurred image since the user briefly stopped panning. Blur is inevitable in some frames despite using a high-end, progressive scan, 3CCD video camera. The bottom row of Figure 7 shows our alignment results and estimated kernels for two blurred frames. Using our technique, we are able to restore a sharp video sequence.

Image matting. Image matting can also benefit from our alignment. Figure 8 (a) is a blurred image. Due to the motion blur, the boundary of the foreground object is semi-transparent - alpha mattes. It is very difficult, if not impossible, to extract the blurred foreground from the single image. This matting problem becomes easier if we take a second non-blurred image, as show in Figure 8 (b). The foreground object and binary mask are interactively extracted from the non-blurred image. Then, we align two images and compute the blur kernel. If the foreground boundary is solid, i.e., the semi-transparency is only caused by the blur, the alpha matte of the blurred foreground can be obtained by simply convoluting the blur kernel with the foreground mask extracted from the non-blurred image. Figure 8 (c-d)





(a) blurred image (b) noisy image (c) deconvolution image (d) registered image

Figure 4. A real example. Top: a fraction of kernels at four levels in the coarse-to-fine searching. The kernel sizes at each level are [24, 48, 96, 160]. The kernel in the red square is the best one at each level. Bottom: (a-b) blurred/noisy image pair. (c) the non-blind image deconvolution result using the obtained sparest kernel. The ripple-like ringing effect, especially in the textureless regions, is mainly caused by the deconvolution. (d) aligned image using the transformation associated with the sparsest kernel.

shows the estimated alpha matte and the composting result.

5. Conclusion

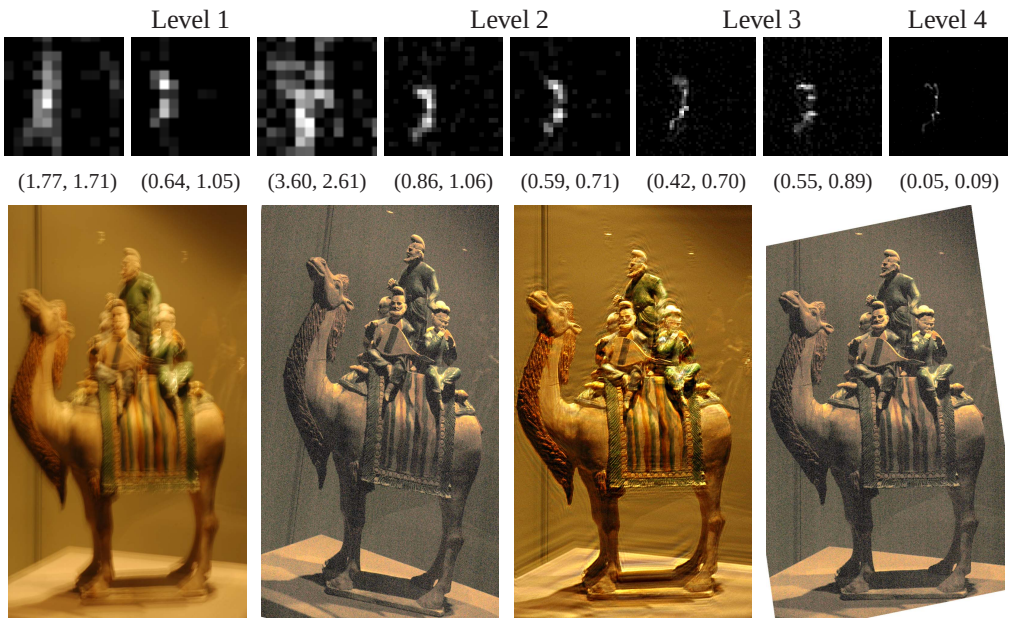
In this paper, we present an approach to align a blurred/non-blurred image pair based on the motion blur kernel statistics. We demonstrated that the sparseness prior can be effectively used to measure the goodness of a kernel so that we can search the alignment parameters without correspondence. An efficient coarse-to-fine search algorithm is also presented. The proposed alignment approach is applied to three applications: image deblurring, video restoration, and image matting.

Handling more complex motion, e.g., perspective transformation or non-global motion, is one of our future works. Locally applying our current approach is one possible solution. Another interesting problem is to align two blurred images using the kernel sparseness prior.

References

[1] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*.

[2] M. Brown and D. Lowe. Recognizing panoramas. In *ICCV*, 2003.



(a) blurred image (b) noisy image (c) deconvolution image (d) registered image

Figure 5. An affine example. Top: example kernels. The sparseness value and the entropy are in the brackets. Bottom: input blurred/noisy image, deblurred image using the computed sparsest kernel, and the noisy image aligned using our estimated affine transformation.

Figure 6. Image deblurring example. Top: input burred image and noisy image (1504×955). Bottom: aligned noisy image, and deconvolution result using the approach in [22].

[3] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer Academic, 2000.

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph. (SIGGRAPH)*, 25(3):787–794, 2006.

[5] J. Flusser, J. Boldys, and B. Zitova. Moment forms invariant to rotation and blur in arbitrary number of dimensions. *PAMI*, 25(2):234–246, 2003.

[6] J. Flusser and T. Suk. Degraded image analysis: An invariant approach. *PAMI*, 20(6):590–603, 1998.

[7] J. Jia, J. Sun, C.-K. Tang, , and H.-Y. Shum. Bayesian correction of image intensity with spatial consideration. In *ECCV*,

Figure 7. Video restoration. The top row shows three frames extracted from a 720×576 video. The bottom row shows the registered result using the sharp image in the middle of the first row and the estimated kernels.

(a) blurred image (b) non-blurred image (c) alpha matte (d) composition result

Figure 8. Image matting. (a) the blurred image is taken by intentionally moving the camera. (b) the non-blurred image is normally captured. Image alignment and kernel estimation are done simultaneously by our approach. Then, we manually cut out the binary mask of the toy from the non-blurred image. The alpha matte in (c) is generated by convoluting the mask and the estimated blur kernel (at the top-right corner). (d) is the final composited image on a new background.

2004.

[8] F. P. Jin, H. and R. Cipolla. Visual tacking in presence of motion blur. In *CVPR*, 2005.

[9] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006.

[10] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. In *ECCV*, 2004.

[11] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *ICCV*.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *IJCAI*, 1981.

[14] L. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79:745, 1974.

[15] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph. (SIGGRAPH*, 25(3):795–804, 2006.

[16] A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in different directions. *IEEE Workshop on Applications of Computer Vision*, 2000.

[17] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recogn. Lett.*, 26(3):311–317, 2005.

[18] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[19] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Computer Vision*, 2(1):1–104, 2006.

[20] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *SCTV*, 2003.

[21] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Comput. Vision Graph. Image Process.*, 35(2), 1986.

[22] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph. (SIGGRAPH)*, 26(3):1–10, 2007.