SENSOR-ASSISTED IMAGE DEBLURRING OF CONSUMER PHOTOS ON SMARTPHONES

Wei Jiang *Dongqing Zhang* *Heather Yu*

American Media Lab, Futurewei Technologies, Inc., Bridgewater, NJ 08807
{wei.jiang, dongqing.zhang, heatheryu}@huawei.com

ABSTRACT

Hand-held mobile phone photography usually suffers from motion blur. Without information about the camera motion, it is difficult for traditional methods to remove such blur. We study sensor-assisted single image deblurring on modern smartphones. Information about the camera motion is obtained from the smartphones' built-in sensors such as accelerometers, gyroscopes, and magnetometers. Addressing the characteristics of mobile photo capture, we propose a camera model to estimate the point spread function (PSF) based on the 3D camera orientation computed from the fused sensor data. A convenient framework is developed for automatically calibrating the camera and sensors using a series of consecutively captured photos, without requiring any extra device. In addition, to accommodate users' sensitivity in artifacts over human faces, a face-adaptive deblurring framework is proposed. Features measuring image quality are computed over the automatically detected face regions, based on which the appropriate deblurring algorithm is selected. We evaluate our approach over 500 consumer photos featuring various levels of blur and illumination, where our algorithm shows clear advantages compared with state-of-the-art traditional blind and non-blind deconvolution methods in both deblurring quality and speed.

Index Terms— image deblurring, mobile, sensors

1. INTRODUCTION

With the widespread prevalence of modern smartphones, mobile photography has become an inseparable part of our everyday life. For normal users, it is hard to hold a cell phone camera steady, and the captured photos usually suffer from motion blur. Single-image motion deblurring has been extensively studied recently. However, without information about the camera motion, removing blur from general smartphone photos is difficult and remains an unsolved problem.

The blur process caused by camera shake during capture is generally modeled as a latent target image $\mathbf{x}$ convolved with a blur point spread function (PSF) $\mathbf{k}$: $\mathbf{y} = \mathbf{x} \bigotimes \mathbf{k}$. The task is to recover the sharp target $\mathbf{x}$ from the blurred version $\mathbf{y}$. Since this is an ill-posed problem, additional assumptions are required. One popular approach is to use prior knowledge about the latent image or the blur PSF. For example, Fergus *et al.* [5] used a mixture of Gaussian to fit the heavy-tailed natural image prior. Levin *et al.* [13] used the sparsity regularization and proposed a faster approximated marginal likelihood optimization algorithm [14]. Shan *et al.* [22] used sparse priors for both the latent image and the blur PSF. A normalized sparse regularization scheme was exploited in [11]. A framelet and curvelet system was used in [2] to obtain the sparse representation. Denoising and deblurring are jointly optimized in [23]. To better handle noises, Zhong *et al.* [27] applied a set of directional filters at different orientations and reconstructed the PSF using inverse Random transforms obtained from the filtered images.

Another group of methods explicitly or implicitly detect salient edges. For example, shock filter was used to generate a sharpened reference map for PSF estimation [3]. Joshi *et al.* [6] predicted sharp edges by first locating step edges and then propagating the local intensity extrema towards the edge. By analyzing the scale of blurry edges, a texture-removal strategy was proposed in [24] to guide edge selection. Based on the step-edge properties in unnatural representation, Xu *et al.* [25] developed a generalized L_0 sparse expression for fast implicit high sparsity-pursuit regularization.

The hardware-based approaches try to provide high-quality results by acquiring information about the camera motion from other sources. For example, by using a pair of images (one well exposed but blurred and one underexposed but sharp), multichannel blind deconvolution was used [26]. Better PSF was estimated by tracking camera motion using data from an auxiliary high-speed yet low-resolution video camera [1]. By using a hybrid system consisting of a DSLR camera and a set of inertial sensors, Joshi *et al.* [8] physically measured the full 6D camera motion to accurately compute spatially-varying blur PSFs.

Although some good results can be achieved, existing methods generally cannot perform satisfactorily on real consumer photos captured by smartphones. First, methods requiring additional hardware supports other than the smartphone itself is not practical for general users. Second, consumer photos are usually large in size, *e.g*, with 8-plus mega pixels. Most previous methods work on less than 1-mega pixel images, and the process already takes minutes or hours on high-end PCs. This is because to avoid the trivial delta

solution of naive MAP inference, marginalizing the posterior needs expensive approximation methods. Using sparse priors also results in costly non-convex objective functions. Third, most smartphone photos have small to medium-size PSFs with relatively simple shapes since the shutter speed is usually pretty fast (in milliseconds). Fig. 6 gives some examples of such PSFs. Compared with synthetically generated blurry images or images taken by DSLR cameras with manually-set long exposure time, it is more difficult to estimate accurate PSFs over real consumer photos. Fourth, users usually do not have the knowledge or interests in parameter tuning. A practical system should eliminate such requirements. Finally, users' tolerance about artifacts is generally quite low, especially over human faces. A practical algorithm should gracefully fail. That is, if we cannot effectively remove the blur, the system should not generate a result less pleasant than the original image.

In this paper, we study sensor-assisted single-image deblurring for general smartphone users. With the increasing popularity, more and more modern smartphones are equipped with various sensors, *e.g.*, gyroscopes, accelerometers, and magnetometers. Information about the camera motion can be obtained from such sensors, which can be further used to compute the blur PSF. Different from the expensive measuring apparatus [8], using smartphone sensors has the following issues. First, the camera needs to be calibrated without any additional device requirements. Second, different from cases of navigation or modeling from videos, where there are consecutive frames and constant time stamps for both frame sampling and sensor reading, there exists inconsistent time delay between the sensor output and the captured photo. Third, the gyro drift needs to be corrected online at high speed.

We propose an efficient smartphone-based image deblurring framework. Fig. 1 shows the overall system. Information from gyroscopes, accelerometers, and magnetometers are fused at high speed, which can alleviate the gyro drift. The blur PSF is then estimated based on the 3D camera orientation computed from the fused sensor data. To accommodate the inconsistent time delay between the sensor output and the captured photo, we assume constant camera motion during photo capture and model the effect caused by such a time delay by a 2D Gaussian term. This is reasonable because of the fast shutter speed of modern smartphones. We only consider 3D camera orientation because the inertial-sensor-based online camera translation estimation is prone to error, and real-time drift correction is not as accurate as offline solutions like [8]. For calibration, the camera parameters can be computed using only a set of consecutively captured photos by the smartphone. Addressing users' sensitivity in artifacts over human faces, we further propose a face-adaptive deblurring method. We compute several features to measure the image quality over the automatically detected face regions, based on which the appropriate deblurring algorithm is selected for each face. Evaluation is conducted over 500 real consumer photos from 7 users featuring various levels of blur and illumination. Experiments show clear advantages of our method in both deblurring quality and speed compared with state-of-the-art conventional deblurring algorithms.

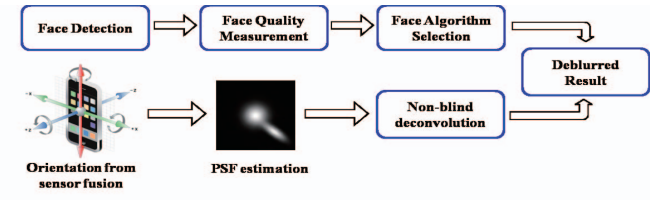


Fig. 1. The overall framework of our system.

2. SENSOR-ASSISTED PSF ESTIMATION

In a pinhole camera the relationship between an image point $\mathbf{x}$ in homogeneous coordinates and the corresponding point $\mathbf{X}$ in real-world 3D coordinates can be described as:

$$\mathbf{x} = C\mathbf{X}, \;\; \mathbf{X} = \lambda C^{-1}\mathbf{x}.$$

C is the intrinsic camera matrix, and λ is an unknown scaling factor. Most cameras have square pixels, and the principle point is very close to the image center. So C is given by:

$$C = \begin{pmatrix} f_x & 0 & w/2 \\ 0 & f_y & h/2 \\ 0 & 0 & 1 \end{pmatrix}, \qquad (1)$$

where w and h are the width and height of the image, respectively. Parameters f_x and f_y are unknown, which need to be recovered through the calibration process.

We consider 3D camera rotations only. This is because the inertial-sensor-based online camera translation estimation is prone to error. The accelerometer data have to be integrated twice to obtain camera translations and real-time drift correction is not as accurate as offline solutions like [8].

2.1. Orientations from sensor fusion

The accelerometer provides the gravity vector, and the magnetometer works as a compass. The information from both sensors suffices to calculate the device orientation. However, the magnetometer suffers from noise, jittering and temporal magnetic influences, and accelerometers are prone to vibration and noise. Also, both sensors are slow to respond. The gyroscope provides the angular rotation speeds of the device for all three axes, which are quite accurate with a very short response time. However, to get the actual orientation, we need to integrate such outputs over time. Small errors can accumulate during integration, which results in the gyro drift.

To alleviate both the gyro drift and the noisy orientation, the output from gyroscope is applied only for orientation changes in short time intervals, and the magnetometer/accelerometer-based orientation is used as the support information over long periods of time. Intuitively,

this is similar to low-pass filtering of the magnetometer/accelerometer signals and high-pass filtering of the gyroscope signals. Fig. 2 gives the sensor fusion framework. More discussions can be found in [28]. The output of the framework will be the 3D camera orientation $\mathbf{\Theta}(t) = (\theta_x(t), \theta_y(t), \theta_z(t))$ at time t. This simple yet effective approach is fast to respond (similar to gyro's speed), which suits our needs. In a common case, the user usually opens the camera, takes a photo, and then turns off the camera. There is no time for complicated sensor fusion computations.

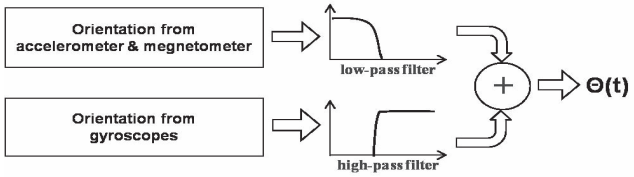


Fig. 2. The framework of sensor fusion.

It is worth mentioning that there are many methods [19] developed to fuse multiple sensors for estimating full 6D camera poses. For example, sensors measuring velocity or acceleration can be blended through Kalman filter with sensors measuring direct position. Visual trackers have also been combined with inertial sensors to help recover camera poses [20]. Good performance has been achieved for outdoor navigation and Augmented Reality. However, such methods cannot be easily used in our case. First, external position references such as GPS may be not available, *e.g.*, when taking photos indoor. Second, different from scenarios of navigation or modeling from videos, where there are consecutive frame inputs and constant time stamps for both frame sampling and sensor reading, we have inconsistent time delay between the sensor output and the captured photo. Finally, it is non-trivial to conduct online visual tracking on cell phones.

2.2. PSF estimation

The camera motion during photo capture is assumed to be constant. This is reasonable because the shutter speed of modern smartphones is pretty fast (faster than 1/200 sec outdoor and about 1/30 sec indoor) under most conditions. When the camera captures a photo during the exposure time, the intensity of light from a scene point (X, Y, Z) at an instantaneous time t is captured on the image plane at a location $(u(t), v(t))$, which is a function of the camera projection matrix $P(t)$. In homogenous coordinates:

$$(u(t), v(t), 1)^T = P(t)(X, Y, Z, 1)^T.$$

$P(t)$ varies with time as a function of camera rotation and causes fixed points in the scene to project to different locations at each time. The integration of these projected observations creates a blurred image. The projected trajectory of each point on the image plane is the PSF $\mathbf{k}$ of that point. For space-invariant blur, $\mathbf{k}$ is the same over the entire image. For image deblurring, we only need the relative motion during the exposure time instead of the absolute motion of the camera. This effect can be modeled by the planar homography that maps the initial projection of points at t_1 to another time t_2:

$$\begin{aligned}(u(t_2), v(t_2), 1)^T &\simeq H(t_1, t_2)(u(t_1), v(t_1), 1)^T, \quad (2)\\ H(t_1, t_2) &= CR(t_2)R(t_1)^{-1}C^{-1},\end{aligned}$$

where $\simeq$ means equal up to a scale. We record two sets of orientation for each captured photo, *i.e.*, $\mathbf{\Theta}(t_1)$ right before the device captures the photo, and $\mathbf{\Theta}(t_2)$ right after the device captures the photo. The corresponding camera rotation matrices $R(t_1)$ and $R(t_2)$ can be calculated based on $\mathbf{\Theta}(t_1)$ and $\mathbf{\Theta}(t_2)$, respectively. Using Equation (2), we can get a line between point $(u(t_2), v(t_2), 1)^T$ and point $(u(t_1), v(t_1), 1)^T$ with length $D(t_1, t_2)$. The spatially invariant $\mathbf{k}_r$ caused by camera rotation is then modeled as a 2D Gaussian, whose center is $(u_k + \beta(u(t_2) - u(t_1)), v_k + \beta(v(t_2) - v(t_1)))$. The variances of the two axes are $\gamma D(t_1, t_2)$ and $\gamma D(t_1, t_2)/3$ respectively. (u_k, v_k) gives the center of the kernel $\mathbf{k}$. β and γ are scaling factors. In our system, we set their values according to the blur scale, *i.e.*, $\beta = 0.01$, $\gamma = 0.5$ for small blur and double these numbers for medium to large blur.

Note that there exists an inconsistent time delay $\delta(t)$ between the moment when a user presses the photo capture button and the actual time when the camera captures the photo. $\delta(t)$ is usually quite small, and we can assume constant camera motion during this time. For smartphone users, this camera motion during $\delta(t)$ mostly comes from the finger movement on pressing the screen. As illustrated in Fig. 3, such a camera motion can be simplified as perpendicular to the image plane, and the effect of this motion during $\delta(t)$ is similar to the out-of-focus blur. We use a Gaussian blur kernel $\mathbf{k}_c$ to model this blur effect. $\mathbf{k}_c$ centers at the kernel center (u_k, v_k) with variance $\sigma_c = 2.0$ or 3.0 along both axes, respectively, for small and medium to large blur. By combining $\mathbf{k}_c$ and $\mathbf{k}_r$ together, we obtain the final PSF $\mathbf{k}$:

$$\mathbf{k} = \alpha \mathbf{k}_r + (1 - \alpha)\mathbf{k}_c \quad (3)$$

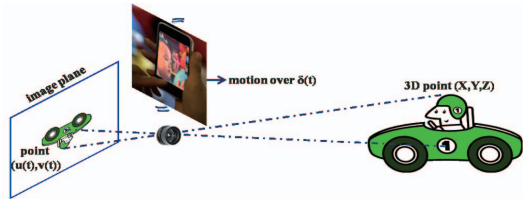


Fig. 3. An illustration of the camera motion during $\delta(t)$.

2.3. Camera and sensor calibration

The camera and sensor related parameters, *i.e.*, f_x and f_y as well as the correspondence of the sensor axes to the camera axes need to be determined. Precise calibration usually needs special equipments, such as the ones in [8] that can rotate the gyroscope at a known constant angular velocity and can

hold accelerometers stationary in various orientations with respect to gravity. Such experiments are too expensive and complicate to carry out by general consumers. Here we calibrate our system using a set of (*e.g.*, 20) consecutively captured photos. The user is asked to take a series of photos about a texture rich static scene, such as a building or a check board. The user should try to stand still when taking each photo, and there should be normal view change and camera shake among different photos. Under the assumption of small constant motion for each photo, the average orientation computed from the sensor data before and after the photo capture $((\Theta(t_1)+\Theta(t_2))/2)$ is used as the orientation of the photo. Matching points are detected from adjacent photos based on the SURF feature and RANSAC matching. The set of point matches $\mathcal{M}=\{(u_i,v_i,u_j,v_j)\}$ can be used as ground truth to formulate an optimization problem for calibration:

$$\min_{f_x,f_y} \sum_{i,j\in\mathcal{M}} ||(u_i,v_i)^T - H(i,j)(u_j,v_j)^T||^2. \quad (4)$$

Since we have relatively few parameters to recover and the calibration process is only performed once for each cell phone, we exhaustively search for the optimal parameters over a dense grid. To discover the orientation correspondences, similar to the approach in [9], we permute its 3 rotation axes and run exhaustive parameter search for each permutation. The permutation that minimizes the objective best corresponds to the cameras axis ordering. This approach works pretty well in practice since the re-projection error is usually significantly larger for incorrect permutations. The whole calibration process takes about 10 minutes.

3. FACE-ADAPTIVE DEBLURRING

Given the estimated PSF $\mathbf{k}$ from the previous section, non-blind deconvolution methods can be used to restore the latent image $\mathbf{x}$ from the blurred observation $\mathbf{y}$. Up to date, most deblurring approaches still rely on the classic Richardson-Lucy (RL) deconvolution algorithm [16, 17] due to its relatively robust and consistent overall performance.

When choosing the appropriate non-blind deconvolution algorithm, we have taken into account several factors. First, Wiener-filtering-based approaches generally cannot perform well, because they are known as sensitive to errors in estimated PSFs and to noise. Second, due to the limitations in mobile computation power and speed, methods involving intensive computation are not practical to use. For example, various image priors and regularization schemes have been extensively studied in the literature to preserve image structures and suppress ringing artifacts, such as the total variation regularization [18], sparse priors [22], Hyper-Laplacian priors [10, 12], and the local two-color prior [7]. Outliers such as pixel saturation and non-Gaussian noise are explicitly handled in [4], and denoising and deblurring are jointly optimized in [27]. These methods usually require expensive iterative nonlinear optimization. Finally, consumers are very sensitive to noise and artifacts over human faces. Special handling of the face area may be necessary to improve user experience. For example, photos captured under low-light, backlit, and normal lighting conditions may show different noises and artifacts. For photos that are originally sharp we probably should not perform aggressive deblurring.

In our system, the classic RL method is adopted to conduct deconvolution over the entire image using the estimated PSF $\mathbf{k}$. In addition, automatic face detection is performed over the captured photo. When there exist detected faces, several quality measurements are further computed over each face, including the sharpness and contrast [15], and the brightness and its variance. Based on these measurements as well as the relative size of the face in the image, an SVM classifier is used to choose the appropriate method to use for the face region, *i.e.*, deblurring using RL method followed by a denoising post process or a less aggressive edge-based enhancement such as sharpening [21]. Fig. 4 gives some examples where choosing the right method to use is important.

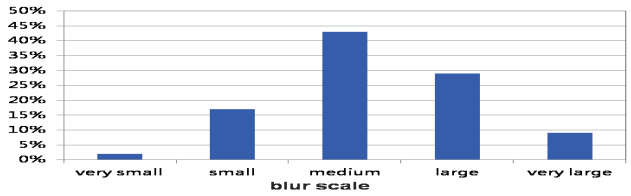
Note that measuring content-invariant sharpness is difficult in general. However, over the face area where the image content is relatively constrained, the sharpness feature can capture useful information about image quality. To compute the sharpness and contrast features, we follow the work of [15]. Edges are detected from gray-scale image after running an averaging filter to reduce noise. Regions containing strong edges are identified and are refined by applying a median filter. The average and standard deviation of the strongest edges give the sharpness and contrast measurements, respectively.

Fig. 4. Face examples. The green border gives the user-preferable method for each case.

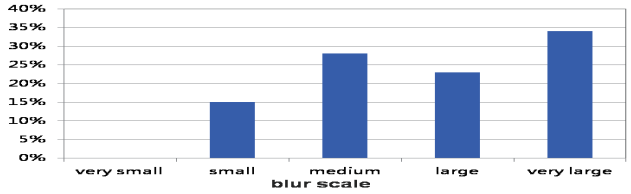
4. EXPERIMENTS

We collected 500 photos from 7 different users using two Android devices, a Samsung Nexus 7 Tablet and a Huawei Ascent D2 smartphone. These are amateur cell phone users aged between 30 to 60 with professions varying from accountants to engineers. 450 photos were collected under normal indoor/outdoor lighting while 50 were under low-light condi-

tions. The users were asked to stay still but with normal hand shakes when taking the photos, as most users would normally do. Fig. 5 gives some statistics about the blur information of these photos. We define 5 blur categories based on the size of the blur: very small as PSF $\leq$ 11 pixels; small as 11 $<$ PSF $\leq$ 21 pixels; medium as 21 $<$ PSF $\leq$ 41 pixels; large as 41 $<$ PSF $\leq$ 61 pixels; and very large as PSF $>$ 61 pixels. A number of 100 photos are randomly selected to train the SVM classifier for face deblurring algorithm selection. Note that there are no ground-truth PSFs for these photos, and we use our sensor-assisted PSF estimates to obtain the blur information. Fig. 6 shows some examples of the estimated PSFs over these photos using the state-of-the-art blind deconvolution algorithm like [25]. They are quite different from the popular synthetic PSFs used in the literature.



(a) data histogram for photos under normal-light condition.



(b) data histogram for photos under low-light condition.

Fig. 5. Blur information of consumer photos. Each bar gives the percentage of photos with the corresponding blur type.

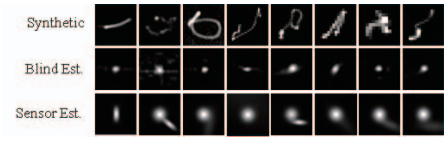


Fig. 6. PSF comparison. The first row shows some widely used synthetic PSFs in the literature. The middle row gives examples of PSFs estimated using blind-deconvolution methods like [25] over real consumer photos. The bottom row gives examples of sensor-assisted PSFs estimates.

To evaluate our approach, we compare our method with two most recent state-of-the-art blind deconvolution algorithms, *i.e.*, the L_0 sparse regularized method ("L_0Reg") [25] and the directional-filter-based method ("DirFil") [27]. In addition, using our estimated PSF, various non-blind deconvolution methods can be used to recover the latent image. We also compare the classic RL method that is used in our system with two other state-of-the-art non-blind deconvolution approaches, *i.e.*, the approach using the hyper-Laplacian prior ("HyLap") [10] and the approach with special outlier handling ("Outlier") [4]. Fig. 7 gives some example comparison results. As shown in Table 1, to process a 2448$\times$3264 consumer photo, the blind method "DirFil" and the non-blind method "Outlier" usually take a few hours on a regular PC with Intel i5 2.5GHz CPU and 4G RAM. Even with possible code optimizations and accelerations, they are still too expensive to be practical. Therefore, they are only applied to 15 photos. The "L_0Reg" method takes about 10 minutes to process each photo, and is applied to 50 photos only. As shown in Fig. 7 (and as generally observed in our experiments) it is usually hard for blind deconvolution methods to recover good PSFs in real consumer photos. In most cases, they either prefer a non-blurred solution or generate severe artifacts. The "DirFil" approach can produce good results sometimes, but is too slow to use in practice. In comparison, our approach can get good PSF estimates and therefore can achieve good performance with much less computation. The non-blind "HyLap" method, although with similar speed to ours, tends to generate overly smooth results. Finally, in our tests, over the evaluated 15 photos, users always prefer our approach rather than "Outlier" or "HyLap", and prefer "DirFil" for only 3 photos. Over the evaluated 50 photos, users also prefer our approach to "L_0Reg" in most cases except for severely blurred cases where they cannot tell any difference between different methods.

Tab. 1. Average time costs of different methods.

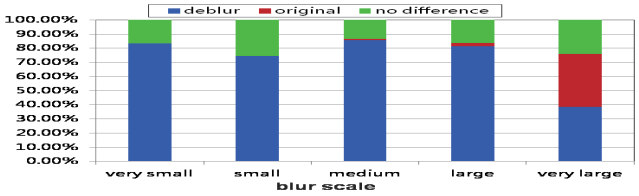
L_0Reg	DirFil	HyLap	Outlier	our method
10 minutes	3 hours	35 seconds	2 hours	30 seconds

As discussed before, a practical system should gracefully fail. To evaluate this, over the entire dataset, the deblurring results using our approach are presented side-by-side with the original photos to 3 users for subjective manual evaluation. The users label whether they prefer the original or the deblurred photo, or cannot tell any difference. Fig. 8 shows the statistics of users' evaluation. As we can see, when there is only a very small or small-size blur, under normal-light conditions, we do not degenerate the image quality at all. When there is a medium-size blur, we fail only for 1% cases. When there is a relatively large-size blur, we can still slightly improve the image quality in most photos. On the other hand, for low-light photos, the algorithm generates more artifacts in general, *e.g.,* by amplifying noticeable noises. Such results also help us to develop some guidance rules in real systems. For example, we can choose not to process low-light photos. Also, we can quickly measure the PSF size based on the sensor data and choose not to process severely blurred photos.

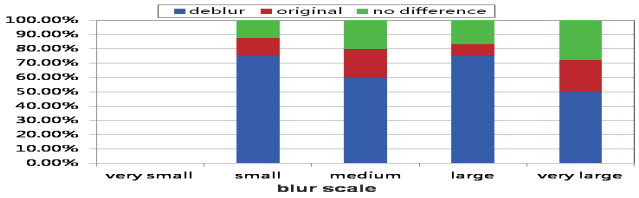
5. CONCLUSION

We have presented a sensor-assisted image deblurring system targeting at consumer photos captured by hand-held smartphones. The camera motion information is obtained by fusing outputs of the smartphone's built-in sensors, based on which the PSF is estimated to remove the motion blur. By assessing the image quality over face regions, the appropriate deblurring algorithm is selected. Experiments have shown that our approach is more efficient and robust compared with state-of-

Fig. 7. Examples of deblurring results.



(a) photos under normal-light condition.



(b) photos under low-light condition.

Fig. 8. Users' evaluation. Each bar shows the photo distribution where users prefer our deblurred results, the original photo, or cannot tell their difference, in each blur category.

the-art conventional algorithms. Future work includes exploring space-varying deblurring of smartphones photos. More studies will also be conducted to address low-light photos.

6. REFERENCES

[1] M. Ben-Ezra and S.K. Nayar, "Motion-based motion deblurring," *IEEE Trans. PAMI*, 26(6):689–698, 2004.

[2] J. Cai and *et al.*, "Blind motion deblurring from a single image using sparse approximation," *IEEE CVPR*, pp. 104–111, 2009.

[3] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graphics*, 28(5), 2009.

[4] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," *IEEE ICCV*, 2011.

[5] R. Fergus and *et al.*, "Removing camera shake from a single photograph," *ACM Trans. Graphics*, vol. 25, 2006.

[6] N. Joshi, R. Szeliski, and D.J. Kriegman, "Psf estimation using sharp edge prediction," *IEEE CVPR*, 2008.

[7] N. Joshi and *et al.*, "Image deblurring and denoising using color priors," *IEEE CVPR*, 2009.

[8] N. Joshi and *et al.*, "Image deblurring using inertial measurement sensors," *ACM Trans. Graphics*, 29(3), 2010.

[9] A. Karpenko and *et al.*, "Digital video stabilization and rolling shutter correction using gyroscopes," Stanford University Computer Science Tech Report CSTR 2011-03, 2011.

[10] D. Krishnan and R. Fergus, "Fast Image Deconvolution using Hyper-Laplacian Priors," *NIPS*, 2009.

[11] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," *IEEE CVPR*, 2011.

[12] A. Levin, *et al.*, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graphics*, 26(3), 2007.

[13] A. Levin and *et al.*, "Understanding and evaluating blind deconvolution algorithms," *IEEE CVPR*, pp. 1964–1971, 2009.

[14] A. Levin and *et al.*, "Efficient Marginal Likelihood Optimization in Blind Deconvolution," *IEEE CVPR*, 2011.

[15] A. Loui and A. Savakis, "Automated event clustering and quality screening of consumer pictures for digital albuming," *IEEE Trans. Multimedia*, 5(3):390–402, 2003.

[16] L. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical Journal*, 79(6), 1974.

[17] W. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Am.*, 62(1), 1972.

[18] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D: Nonlinear Phenomena, 60(1-4):259268, 1992.

[19] J. Ryu and J.C. Gerdes, "Integrating inertial sensors with GPS for vehicle dynamics control," *Journal of Dynamic Systems, Measurement, and Control*, 126(2):243–254, 2004.

[20] G. Schall and *et al.*, "Global pose estimation using multi-sensor fusion for outdoor augmented reality," *IEEE ISMAR*, pp.153–162, 2009.

[21] J. Schewe and B. Fraser, Real World Image Sharpening with Adobe Photoshop, Camera Raw, and Lightroom (2nd Edition), Peachpit Press, September, 2005.

[22] Q. Shan and *et al.*, "High-quality motion deblurring from a single image," *ACM Trans. Graphics*, 27(3), 2008.

[23] Y.W. Tai and S. Lin, "Motion-aware noise filtering for deblurring of noisy and blurry images," *IEEE CVPR*, 2012.

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

[25] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 Sparse Representation for Natural Image Deblurring," *IEEE CVPR*, 2013.

[26] L. Yuan and *et al.*, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graphics*, 26(3):110, 2007.

[27] L. Zhong and *et al.*, "Handling noise in single image deblurring using directional filters," *IEEE CVPR*, 2013.

[28] http://www.thousand-thoughts.com/2012/03/android-sensor-fusion-tutorial/