SmartDCap: Semi-Automatic Capture of Higher Quality Document Images from a Smartphone

Francine Chen Scott Carter Laurent Denoue
FX Palo Alto Laboratory
Palo Alto, CA USA
{chen, carter, denoue}@fxpal.com

Jayant Kumar
University of Maryland
College Park, MD USA
jayant@umiacs.umd.edu

ABSTRACT
People frequently capture photos with their smartphones, and some are starting to capture images of documents. However, the quality of captured document images is often lower than expected, even when an application that performs post-processing to improve the image is used. To improve the quality of captured images before post-processing, we developed the Smart Document Capture (SmartDCap) application that provides real-time feedback to users about the likely quality of a captured image. The quality measures capture the sharpness and framing of a page or regions on a page, such as a set of one or more columns, a part of a column, a figure, or a table. Using our approach, while users adjust the camera position, the application automatically determines when to take a picture of a document to produce a good quality result. We performed a subjective evaluation comparing SmartDCap and the Android Ice Cream Sandwich (ICS) camera application; we also used raters to evaluate the quality of the captured images. Our results indicate that users find SmartDCap to be as easy to use as the standard ICS camera application. Also, images captured using SmartDCap are sharper and better framed on average than images using the ICS camera application.

Author Keywords
Mobile computing; mobile capture; image analysis; documents

ACM Classification Keywords
H.5.2 User Interfaces; I.7.5 Document Capture: Document analysis, Scanning

INTRODUCTION

Mobile devices with cameras are rapidly proliferating and are being used to capture media at an ever-increasing rate. Furthermore, document scanning applications in particular generally rank among the most downloaded enterprise applications on the Android and iOS application stores, with several individual applications having more than a million downloads. Unfortunately, the quality of document photos, even using specialized applications, is often poor. Although cameras on mobile devices have increasingly higher resolutions, the small camera size, uneven lighting conditions, and other contextual issues often conspire to make using a camera phone to capture a high quality image of a document difficult. Even if a user tries to carefully position the document in the camera's preview screen, the quality of the final image may still not be as good as expected. Some possible causes for poor image quality include:

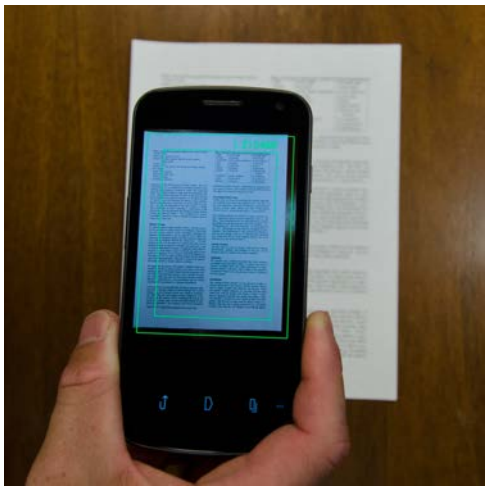


Figure 1. Using SmartDCap to take a high quality document image.

Focus: With small viewers, it may not be obvious when the focus is poor. At the close ranges used for capturing page images, small hand movements can cause blur, and autofocus does not always correctly determine focus. In addition, the camera may occasionally refocus without warning.

Framing: The captured photo may crop part of the desired content or include too much extraneous context (and therefore the resolution is lower than it could be) because it is hard to simultaneously check that multiple edges are correctly framed. A framing aid based on structured light was developed for capturing documents using a regular camera [14]. Although this approach improved framing, it is not currently feasible for use with smartphones.

Rotation: The frame of the camera image may be rotated relative to the page, resulting in lower resolution characters.

Shadows and poor lighting: Poor lighting results in lower image quality. A slower shutter speed is often used to com-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IUI'13, March 19–22, 2013, Santa Monica, CA, USA.
Copyright 2013 ACM 978-1-4503-1965-2/13/03...$15.00.

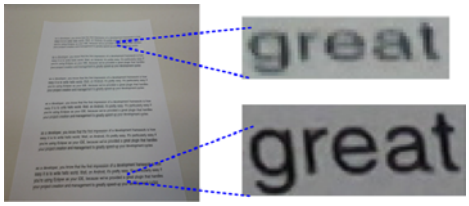


Figure 2. Sharpness can vary due to limited depth of field.

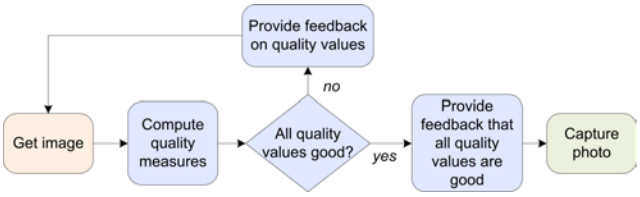


Figure 3. SmartDCap analyzes video preview frames in realtime, communicates their quality to the user, and automatically snaps a photo when the image quality is high.

pensate under low light conditions, with the consequence that small hand movements result in blur. The act of taking a picture may itself lower the quality of the image, especially if the user casts a shadow by leaning over the document. Sometimes the shadow is not apparent when looking at the paper, but appears exaggerated in the captured photo.

Depth variation: Due to limited depth of field in most smartphone cameras, a photo of a page or other items that fill the camera screen may have blurry regions if the camera plane is not relatively parallel to the plane of the page. For example, if the user holds their camera to one side, as someone sitting might do when photographing a page on their desk, the camera may not be able to uniformly bring into focus the different depths at close range. This is illustrated in Figure 2, where the text at the top of the page is blurrier than the text at the bottom.

Although post-image processing can address some of these issues, some information may be lost and artifacts can be introduced. The resulting quality is not as good as a photo of a flat page captured straight-on so that it is aligned with the plane of the camera sensor, closely framed for high resolution, and with good lighting.

In this paper we present our *Smart Document Capture* (SmartDCap) system for semi-automatic, higher quality capture of the content of a document page. SmartDCap provides realtime feedback to users about the likely quality of an image of a flat document page and automatically snaps a photo when the quality is acceptable, allowing the user to lean away from the document so as not to interfere with the captured image. We describe the two measures used to estimate image quality, sharpness and framing, and show that the sharpness measure correlates well with perceived sharpness of document and scene images. We also describe a subjective study evaluating the ease of using our SmartDCap application and the Android Ice Cream Sandwich (ICS) camera application. We also present an evaluation of the quality of captured images.

RELATED WORK

Many applications for the iPhone and Android perform post-processing to improve pictures people have taken of business cards and document pages. For example, Genius Scan[1], CamScanner+[2], TurboScan[3] and CamCard[4] offer features such as automatic and semi-automatic detection of the edges of a business card or page and perspective correction, color correction, exporting the document in formats including JPG and PDF, and supporting email and sharing of the document. ClearCam[5] merges several rapid-fire photos into a single higher-resolution image. With these document-capture applications, the offered features are applied to an image that has been captured in the usual way: look at the phone's screen, decide when a shot of a document page is good, and press a button. The applications do not help a user to capture a better image, but perform post-processing to try and improve whatever image was captured. However, there is a limit on how much the post-processing methods can improve a poorly captured image. For example, when post-processing to sharpen an image is applied to an image with varying sharpness, as in Figure 2, the blurrier regions will often still appear blurrier.

In contrast, our focus is on improving the quality of captured document images *at the time of capture*. With our method, the user need not try and determine when the capture conditions are good. Instead, the user only has to adjust the phone until SmartDCap detects the image quality is good. The resulting higher quality images captured using SmartDCap can then be improved even further using any of post-processing methods employed by other capture applications. Because SmartDCap provides higher quality images to begin with, the post-processed images should be of at least as high quality as the regularly captured images, if not higher. For example, SmartDCap encourages pages to be captured straight-on, so there is minimal blurriness from limited depth of field.

Many cameras now use face detection or smile detection to provide feedback when one or more people are visible in a camera viewer. Some Casio cameras have an Auto Shutter mode[6]. The Anti Blur Auto Shutter will automatically capture a photo when the camera is stationary. This is especially useful for capturing sharper images when subjects are moving. However, a stationary camera alone may not be adequate for capturing sharp document images; in the close range used to take photos of pages, the camera autofocus does not always work correctly. Thus, an image may be blurry even if the camera is stationary. Casio also offers Auto Shutter for Panning, Smile Detection, and Self Portrait modes. However, none of these are applicable to capturing document pages.

[1]https://play.google.com/store/apps/details?id=com.thegrizzlylabs.geniusscan

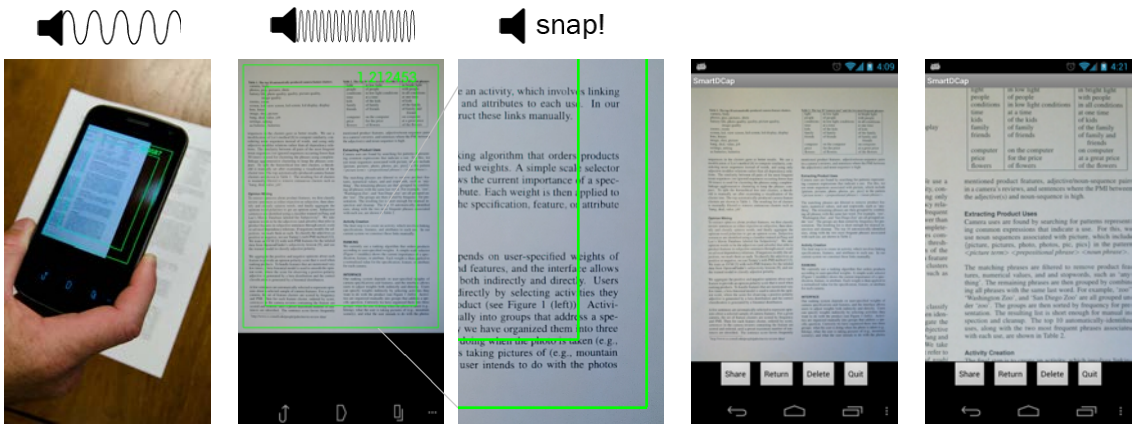
[2]http://itunes.apple.com/us/app/camscanner/id388624839

[3]http://itunes.apple.com/us/app/turboscan-quickly-scan-multipage/id342548956

[4]http://creativeoverflow.net/camcard-application-review/

[5]http://itunes.apple.com/us/app/clearcam/id364930963

[6]http://www.exilim.eu/euro/exilimcard/exs10/editing/#editing/autoshutter



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

Figure 4. Capturing a photo with SmartDCap. (a) At first, as the user positions the device, the auditory feedback tone frequency and sharpness score are low. (b) As the document sharpness score increases, the feedback tone frequency increases. (c) Once the photo is sharp and the edges of the text or image regions to be captured are between the two green overlaid rectangles, the application snaps a photo. (d) The user can then review the shot before sharing it, deleting it, or taking another photo. (e) The user can also pan and zoom the captured image to verify its quality.

A few systems have been developed which provide feedback on framing an image. NudgeCam [2], which provides hints at point-of-capture to help users take a high quality video. [17] and EasySnap [19, 7] couple real-time image processing with audio feedback to help the visually impaired center a text region of interest for taking a photo. In [17], tones at different pitch and tempo "based on the distance on the screen from the current position to the target" were used as feedback, where the target is text in street scenes. EasySnap [19, 7] gives verbal feedback, such as "Go Left" and "Turn Right". The version in [19] has a "document" mode that provides audio feedback for centering and aligning text but does not handle pictures on a page. In contrast, the framing in SmartDCap is more precise and focused on how well text and/or pictures in a document fill the photo. Additionally, shots of any region sufficiently separate from the rest of the page content, such as one or more columns, a part of a column, a figure, or a table, can be taken. Another difference is that SmartDCap automatically determines when to take a photo.

Our contributions in this paper include a method for capturing higher quality images of documents *at the time of capture*, which enables even higher quality final images after post-processing, and a simple-to-use application, SmartDCap, that determines when to capture an image.

SMART DOCUMENT CAPTURE

The SmartDCap application helps a user to capture higher quality images of a document from a smartphone. The application can run on a phone with no external support – it only needs access to the phone's camera.

The application has three distinct stages: *analysis* of video preview frames; *capture* of a high resolution photo; and *user review* of the captured photo.

Analysis As the camera views the document, the system analyzes the preview frame using two distinct quality scores that characterize image sharpness and framing (Figure 3). The two scores are used to determine whether the image quality is high enough for capturing a good photo. The phone translates these scores into feedback to the user. Past work has found that real-time feedback can improve task performance in mobile-video applications [11], that such applications should support multiple different types of feedback to match the demands of different environments [6], and that non-speech, auditory feedback in particular can enhance visual interactions on a mobile device [20]. In SmartDCap, feedback is presented as a visual score displayed on the screen as well as audio trills and tones, where lower scores are mapped to low frequency tones and higher scores are mapped to high frequency tones (Figure 4a,b). In designing the auditory feedback, we employed a common method for auditory graphing: the data values, i.e., sharpness estimates, were mapped to the frequency of a short tone, so that increasing pitch corresponds to increasing sharpness [13, 18]. The sharpness values were clipped to a minimum and maximum value to limit the frequency range. The tones were relatively short to better accommodate quick changes in sharpness, but long enough to allow users to perceive pitch. Although sound frequency is a relative perception for most people, in SmartDCap the tones are played up to five times a second (after each sharpness computation a tone corresponding to the current sharpness value is played if a sound is not currently being played to allow for a short pause between tones) so that a user only need be concerned with whether the pitch is rising. Also during the analysis stage, the screen of the camera has overlaid two rectangles, which we refer to as *framing rectangles*, that indicate where the text or page content should extend as in Figure 4a-c. With the camera in autofocus mode, the user holds the phone approximately over the region of interest, and then waits until the feedback tones are high and constant in pitch to indicate that the image is sharp and in focus. As the user adjusts the position of the phone so that edges of the content of interest fall between the two rectangles, the camera continues to provide sharpness feedback.

Capture When both sharpness and framing quality are good, a short trill, an earcon [1], is played to alert the user that a photo is about to be automatically captured (Figure 4c). The user should try to hold the camera still while the picture is captured to minimize motion blur, especially in dim light. When both sharpness and framing quality are good in a second video frame, a second short trill at a higher pitch is played and the system automatically takes a photo of the document.

Review Since the application analyses video preview frames but snaps a high resolution photo, there can be a brief delay in the handoff between the video preview analysis and the photo capture. Though unlikely, it is possible that the device moves between the time the last video frame was analyzed and the time the phone snapped the high resolution image. For this reason, it is important to provide a pannable, zoomable view to allow the user to review the captured image (Figure 4d,e). From this view the user can share the photo with an external application, take a photo of a new page, or delete the current image and reshoot the current page.

We implemented our quality measures and the SmartDCap interface on the Android platform. We have tested the application on a variety of devices running Android 4.0.x (ICS) as well as Android 4.1.x (Jelly Bean). For the purposes of this paper and the evaluations described later, we deployed the application to a Galaxy Nexus running ICS. Computation of the sharpness estimates and the character-based framing measures in Android Java was too slow for real-time use. So to compute the quality measures quickly enough for real-time feedback, the measures were implemented in native C++ and called methods from the Android port of the OpenCV library[7]. We found that in the native implementation, the time to estimate framing quality was negligible compared to the time to estimate sharpness. For common technical articles, our native implementation processed approximately 2-5 frames/sec, depending in part on the complexity of the image. Note that this rate applies only to the framing and sharpness feedback – since processing was run on a thread separate from the camera the video content was shown to the user at the default frame rate. We next describe the computation of the image quality measures used by SmartDCap.

Quality Measures

Two different measures are used to estimate when the camera is in a good position for taking a photo: 1) image sharpness and 2) framing quality. We do not explicitly compute the orientation of text lines to estimate how well they are aligned with a camera edge. While this may improve results, this is computationally expensive, which we try to minimize for real-time mobile capture. Instead, as described below, the simple-to-compute image sharpness measure and framing quality measure indirectly give an indication of text line alignment.

Image Sharpness

We developed an efficient method for estimating the sharpness of the preview image that is computed on-the-fly on a smartphone. Several top-performing, perceptually-motivated, sharpness measures for scene images are based on the width of edges (sharper edges have smaller edge widths) measured in pixels e.g., [3, 12]. However these measures are slow to compute and do not perform well on text, which has sharp edges that often only three or fewer pixels wide. Instead of measuring edge width in pixels, a coarsely quantized value, we estimate the sharpness of an edge based on the "slope" of gray-scale intensities. We compute the slope by integrating the second derivative of gray-scale intensities over a small window around an edge. Since the second derivative will be close to zero away from an edge, summing outside the edge does not affect the computed slope. Thus our method is less sensitive to pixel width quantization than computing slope directly. Taking a digital derivative by computing differences, the second derivative at pixel $I_{k,j}$ is computed digitally as a difference of differences, $(I_{k+2j} - I_{k,j}) - (I_{k,j} - I_{k-2,j})$, assuming pixel width is a constant that can be ignored. The sharpness, $S_x(i,j)$, in the x-direction at median-filtered edge pixel $I_{i,j}$ located at (i,j) in an image, is computed as the digital integral (i.e., sum) of the magnitude of the second derivative of median-filtered pixel values, normalized by the magnitude of the change in contrast over a window:

$$S_x(i,j) = \frac{\sum_{i-w \le k \le i+w} |(I_{k+2,j} - I_{k,j}) - (I_{k,j} - I_{k-2,j})|}{\sum_{i-w \le k \le i+w} |g_{k,j} - g_{k-1,j}|}$$

where $g(k,j)$ is the grayscale intensity of a pixel at (k,j) in the image and w is a parameter defining window size. In practice, we set $w = 3$. Because we do not care whether slope is positive or negative, slope magnitudes are used. The denominator provides normalization for different contrast levels; it was noted in [3] that as contrast increases, the width of just noticeable blur decreases. The pixel at i,j is defined to be sharp in the horizontal direction if $S_x(i,j) > T$, where the threshold T is learned from a set of images labeled for sharpness.

Our measure for estimating the overall sharpness of an image, ΔDoM, combines the proportion of edge pixels that are sharp in the horizontal and vertical directions using the Frobenius norm:

$$\Delta DoM = \sqrt{\left(\frac{\#sharpPix_x}{\#edgePix_x}\right)^2 + \left(\frac{\#sharpPix_y}{\#edgePix_y}\right)^2}.$$

Dataset	*Method*		
	JNB	CPBD	ΔDoM
Doc	0.43	0.31	0.63
LIVE	0.84	0.94	0.89
CSIQ	0.77	0.89	0.84

Table 1. Spearman rank correlation of perceived image sharpness with three sharpness estimation methods on a document image dataset and two scene image datasets.

Image Sharpness Evaluation We evaluated how well our ΔDoM sharpness measure corresponds with perceived sharpness on three datasets:

Doc 135 document page images contributed by 27 smartphone users. Each page was labeled for sharpness by 21-25

[7] http://opencv.willowgarage.com/wiki/

Dataset	*Method*		
	JNB	CPBD	ΔDoM
Doc	33.63	55.46	3.91
LIVE	2.25	1.05	0.27
CSIQ	1.71	0.68	0.26

Table 2. Computation time per image (in sec) for three sharpness estimation methods on a document image dataset and two scene image datasets.

(a) (b) (c)

Figure 5. Images that do not meet framing quality constraints: (a) page is rotated (b) text area is too small (c) all four margins are bad.

Mechanical Turk workers from a pool of 76 workers (the judgments of seven workers who had poor agreement with the other 76 workers were removed).

LIVE A freely available image sharpness dataset[8] [16]. We used the Gaussian-blurred subset composed of 174 images labeled for sharpness by 24 subjects.

CSIQ A freely available image sharpness datase[9] [8]. We used the Gaussian-blurred subset composed of 150 images labeled for sharpness by 35 subjects.

We compared how ΔDoM performs against two leading perceptually-based sharpness measures, JNB [3] and CPBD [12], for estimating the sharpness of photos of natural scenes. Table 1 shows the Spearman rank correlation between manually labeled sharpness and sharpness estimates by the JNB, CPBD and ΔDoM measures. We observed that on the Doc dataset, ΔDoM performed best among the measures. We observed that ΔDoM also performed competitively with JNB and CPBD for predicting the sharpness of the blurred scene images in the LIVE and CSIQ datasets.

We also compared the computational speed of the three methods by measuring the average time it took to process an image on each of the three datasets. A 64-bit 2.83 GHz Intel Core2 Quad Windows 7 machine with 4 GB of memory was used for all computations. A MatLab implementation of ΔDoM was used for comparison against the freely available MatLab implementations of JNB and CPBD. Each method and dataset combination was run three times and the median of the average times is shown in Table 2. Note that ΔDoM is significantly faster than JNB and CPBD. Unlike the JNB and CPBD methods, ΔDoM does not require computing multiple time-consuming exponentials, Canny edge-detection, or counting pixels for edge-width computation.

[8]http://live.ece.utexas.edu/research/quality/subjective.htm
[9]http://vision.okstate.edu/?loc=csiq

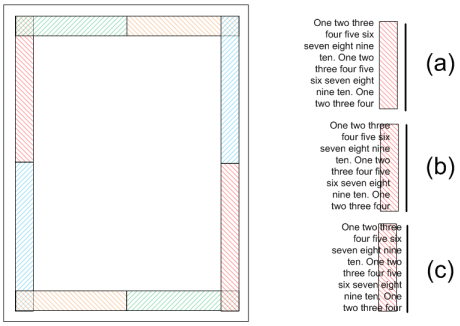


Figure 6. Quiet Margins in an image are required to extend from each edge to inside the nearest hashed/colored area. There are two zones along each edge in the left figure. The quiet margins meet framing quality constraints for a zone on the right side in (b) but not in (a) or (c).

Framing Quality

A document should be well aligned to the preview image. Unlike scanned images, capture conditions when using a mobile device are not controlled and the camera may be so close that no margins or full columns are in view, or so far that other irrelevant objects are in view. Three examples of poorly framed text are shown in Figure 5. When taking a photo of a document page, a user may wish to photograph the whole page or only part of the columns in a page. Identifying the edges of the page can be used to frame a whole page, but not text columns or regions such as a figure or table.

To identify whether the content is well-framed, we offer a simple method for estimating framing quality that can be computed in real-time on a smartphone. While many page segmentation techniques have been proposed, e.g. a comparison of six methods is given in [15], these techniques are not directly applicable to camera-captured images of document pages due to many factors [10], including lack of consideration for real-time computation on a mobile device and the assumptions of flatbed scanning without extraneous objects and of pages being printed on white paper. Rather than performing full page segmentation, we draw from a subset of page segmentation methods to identify "gutters" of white space between columns of content to estimate whether framing along an edge is good.

We focus on the width of the "gutter" between a text column and another column or edge of the page, which allows a user to frame regions of a page in addition to whole pages. We refer to these gutters as **Quiet Margins** and the width of a gutter as a 'quiet margin size'. The permissible quiet margin size is constrained between a minimum (0.02 * imageWidth) and maximum (0.08 * imageWidth), represented in Figure 4a-c by the outer and inner green framing rectangles, respectively. A minimum quiet margin size helps ensure that a column is not cropped; a maximum quiet margin size helps ensure that most of the image is filled with the region of interest. Although the background could be cropped as done by many applications such as those mentioned in Related Work, the resulting resolution of the text content decreases as the background area increases. To enforce rotation restrictions, two zones along each edge are checked for whether the quiet margin values are acceptable. The allowable range of the inside

(a) (b) (c)

Figure 7. Captured images that meet framing quality constraints: (a,b) left, right and top margins meet constraints (c) top, bottom and left margins meet constraints.

edge of quiet margins with two zones per side is shown by the hashed, colored areas in Figure 6.

A number of features can be used to estimate quiet margin size, including pixel values, text character locations, and feature point locations. Independent pixel-based and connected component-based methods have been developed for page segmentation [15]. We chose to use a combination of these two complementary approaches: (1) pixel-based works well on uniform backgrounds and handles photos, and 2) character-based works well on text, handles non-uniform backgrounds better than pixel-based, but does not handle photos with gradual color changes well.

Pixel-based Our pixel-based method for estimating quiet margin size is similar to computing pixel-based projection profiles [9], but is computed over only part of a page for efficiency. Specifically, the image is first binarized, and then a row/column of pixels in a zone starting at the edge is processed. A row/column is classified as 'quiet' if it contains fewer than a small, pre-defined number of foreground pixels, thus allowing for some noise in a margin. To handle different colored backgrounds, the background color of a zone is identified as the predominant color of the row/column along the edge of that zone. A zone is labeled as 'good' if the first non-quiet row/column from the edge falls within the hashed areas in Figure 6.

Character-based To estimate quiet margin size based on character locations, our approach is similar in spirit to identifying gutters by computing a projection profile of all connected components, as in [5], but more efficient. To identify character-size components, contours of edges are identified by first converting the image to grayscale, applying Canny edge detection, and then applying connected component analysis. Next, connected components of a size and aspect ratio that are not in a range appropriate for text characters are filtered out. The location of the bounding box of each remaining connected component is computed and assigned for consideration in one or more zones. For example, the bottom zone along the right margin in Figure 6 (red) considers all bounding boxes which have a centroid in the bottom half of the image. Next, rather than creating a profile of connected component bounding boxes with values for each pixel along an edge, we examine only the locations of one side of the bounding boxes in the zone for efficiency. For example, again using the bottom zone in the right margin, only the locations of the right edge of bounding boxes are examined. To remove noisy characters from consideration while preserving characters at the right edge of a paragraph, a small number of the rightmost characters (up to 10) are ignored unless they are aligned with at least three other bounding boxes in a direction roughly parallel to the edge. The difference between the rightmost remaining value and the right edge of the image serves as an estimate of the right margin; the bottom right zone is 'good' if the margin falls in the (red) hashed area (see Figure 6b). To handle non-text regions, the number of characters is counted, and if there are too few, it is noted. The margin size is computed similarly for the other zones. A zone that was noted as having too few characters is labeled 'good' if the other zone associated with the same edge has been labeled 'good'.

An edge is labeled as a good quiet margin if each zone of an edge is determined to be good by either the pixel- or character-based methods. At least three good quiet margins are required in order for the framing of the page image to be considered acceptable, as illustrated by the examples in Figure 7. Note that quiet margins can handle different colored backgrounds and photos on a page, as well as text. Also note in (a) and (c) that by requiring only three, rather than four, edges to have good margins, we can capture parts of pages that do not completely fill the image and yet are well-framed.

Combined Quality Measures

The Image Sharpness and the Quiet Margins measures are used together by the SmartDCap system for providing feedback and determining when the image quality is high enough for taking a photo. Image Sharpness is used first to indicate the sharpness of the preview image. If the preview is sharp enough for capturing a photo, then the Quiet Margin measure is checked simultaneously. When both the framing is good and the preview image is in focus the camera can automatically capture a photo. When both measures jointly indicate good image quality, many of the issues outlined in the Introduction are ameliorated:

Focus: Image Sharpness indicates when the image is sharp.

Framing: Quiet Margins constrain the framing to be relatively straight.

Rotation: Quiet Margins constrain the amount of rotation allowed.

Shadows and poor lighting: When there are shadows over the image or poor lighting, the edges are not as sharp, and so the Image Sharpness measure will be lower.

Depth variation: When taking a photo from the side, both the Image Sharpness and the Framing Quality measures will be poor.

EVALUATION

We conducted a subjective evaluation to compare the *ease of using* SmartDCap versus a standard camera application. We also evaluated the *quality* of the captured photos.

In our evaluations, we compared SmartDCap and the Android ICS camera application (shown in Figure 9). We chose to

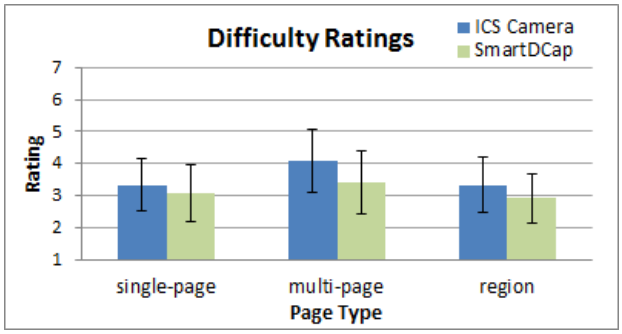


Figure 8. Mean difficulty ratings with 95% confidence intervals for images captured by the ICS camera application and SmartDCap by page type. Rating of 1 is easy, 7 is difficult.

Figure 9. The default Android ICS camera application used in the evaluation. The application automatically focuses when taking a photo. Users could view the previously taken image by clicking on its thumbnail (lower right).

compare against the ICS camera application since it comes pre-installed on the device and is therefore the default camera application for most users. Although we could compare against other capture applications, the improvements that they offer are done *after* capture; those same post-processing improvements can also be applied to the images captured by SmartDCap. Comparing to the default application allows us to test whether SmartDCap improves the quality of photos captured *before* post-processing.

Subjective Evaluation

For our subjective experiment, we used a Galaxy Nexus with both the ICS camera application and SmartDCap installed. The SmartDCap display was as shown in Figure 1, except that the sharpness score was rotated 90 degrees clockwise and located inside the upper right corner of the inner framing rectangle. All of our 12 subjects used both SmartDCap and the ICS camera application; half the subjects used SmartDCap first. Before a subject used an application, instructions and a demo were given on its use. Both for SmartDCap and the ICS camera application, users were instructed to take photos in focus and to fill the camera view with the content of interest and parallel to the viewer edges. Users were also told

	Factor	*F-value*	*Significance*
sharpness	lighting	666.19	$p < 0.01$
	capture method	25.32	$p < 0.05$
framing	lighting	51.51	$p < 0.05$
	capture method	142.23	$p < 0.01$

Table 3. Significant factors for sharpness and framing

to minimize shadows as well as possible. For SmartDCap in particular, subjects were instructed to adjust the position of the camera to align the content to be captured so that at least three edges of the desired content were between the framing rectangles, and then to hold the position until the camera took a photo. For each application, the subjects first practiced using the application and then were asked to take good photos of three types of documents: a single brochure-type page, pages from a 4-page paper, and a close-up of a picture on part of a page. After a photo was taken using either application, the subjects could check the image quality by panning and zooming the captured image. For both applications, the subjects were allowed to take photos of a page until they were satisfied with the quality of the photo. We decided on this approach rather than asking subjects to take only one photo of a page since it better reflects observed practices.

After taking one or more photos of each of the three page types using one application, subjects were asked to rate on a scale of 1 (easy) to 7 (difficult) how hard it was to capture an acceptable image for each type of document. We also questioned participants about their experience with SmartDCap features, asking them to rate (using the same scale) the difficulty of aligning the document content boundary between the green framing rectangles, and, from 1 (agree) to 7 (disagree), whether the auditory feedback helped them understand when they needed to adjust the camera, whether they understood how to adjust the camera, and whether they preferred to have the application capture images automatically. We also asked participants to provide open-ended feedback about the strengths and weaknesses of each application.

Half the subjects were run in a poorly lit ("dim") room and half in a well-lit ("bright") room to examine how the capture applications perform under both conditions. Images may be less sharp under dimmer lighting due to a slower shutter speed combined with small hand movements. Because documents are photographed at close range, small hand movements are relatively noticeable.

Thus the factors that may affect the difficulty scores are application (ICS camera application vs. SmartDCap), task type, and lighting. We performed a 3-way ANOVA on these factors with difficulty score as the dependent variable; Figure 8 shows difficulty ratings by application type and task type. These results indicate that SmartDCap is not any harder to use than the ICS camera application. In addition, some users reported that SmartDCap makes photo review easier.

Feedback from the subjective questions specific to the SmartDCap application revealed that users were conflicted about aligning a document with the green rectangles. While users overall rated the task as moderately difficult, others found that

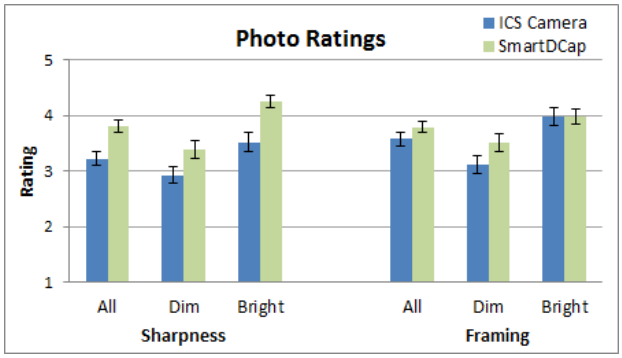


Figure 10. Mean ratings with 95% confidence intervals for sharpness and framing of captured images using the ICS camera application and SmartDCap under dim, bright, and combined (all) lighting conditions. Rating of 1 is poor, 5 is excellent.

the "green borders [helped] line up edges." They also relied more on the visual feedback than the audio feedback, and suggested that the application show a "visual indication of which edge is an issue" when framing a document. Users were particularly fond of the automated capture feature, which made it "much easier to concentrate on framing region of interest ...." They also appreciated not "having to press the button which messes up focus."

Image Quality Evaluation

To evaluate the quality of the captured photos, three people not in the subjective study were asked to rate the sharpness and framing of photos taken during the study on a scale of 1 (poor) to 5 (excellent). Since some of the subjective study participants experimented with the applications to try out different options, we presented (in random order) only the last photo of each page captured by each participant, for a total of 6 photos/(subject&app) * 12 subjects * 2 apps = 144 photos.

We ran a 2-way ANOVA with repeated measures on the factors of lighting (dim and bright) and capture method (SmartDCap or ICS camera application) with *sharpness* as the dependent variable. Both lighting and capture method were main effects, with lighting more significant. We also ran a 2-way ANOVA with repeated measures on the factors of lighting and capture method with *framing* as the dependent variable. Both lighting and capture method were again main effects, but with capture method more significant (Table 3). There were no significant interactions between factors.

More detailed results are shown in Figure 10. Note that both measures of image quality are better under bright light, as has been noted by others (e.g., JotNot[10], TurboScan). One-sided t-tests at the 0.05 level of significance indicate that SmartDCap captured sharper images than the ICS camera application under both dim ($p = 0.002$) and bright ($p = 1.588e^{-6}$) lighting conditions. And under dim lighting, framing using SmartDCap was significantly better ($p = 0.006$).

DISCUSSION AND PROPOSED REFINEMENTS

Our evaluations indicate that SmartDCap captures sharper, better-framed images semi-automatically and is as easy to use as the standard ICS camera application. SmartDCap's framing guides for closely-framed images result in higher resolution of the captured context while reducing the number of extraneous pixels. Although it is difficult for a user to simultaneously check that all four edges are well-framed when capturing closely-framed photos of document content, SmartDCap reduces the difficulty by offering guides for framing and automatically deciding when to capture, allowing users to focus on framing. The result is that the content is captured well-framed, at a higher image resolution, and sharper, both because the user does not press any buttons (which can cause camera movement) and because the application requires a minimum image sharpness before snapping a photo.

Photos captured by SmartDCap can be processed using the types of post-processing techniques that other capture applications offer, such as perspective and tone correction, to further improve the captured image. Because the SmartDCap images are of higher quality to begin with, the processed images should be of at least as high quality as those captured using other apps.

Capture applications that offer perspective correction may implicitly encourage a user to capture pages from one side, often resulting in some regions of the photo being blurrier, especially when capturing larger areas, such a page, than when capturing smaller areas such as business cards. Even with simple post-processing of the image offered by capture apps, parts of the photo will often still be blurrier. In contrast, SmartDCap checks sharpness over the entire image so that the whole page is more uniformly sharp.

Based on user comments, we propose several refinements to SmartDCap. Two areas that our study suggests changes would be beneficial are the framing rectangles and capture speed under dim lighting. We further propose an extension of SmartDCap usage.

In our experiments, the distance between the framing rectangles was set to be relatively small and to be close to the edge of the mobile display. This was to force a page to be captured straight on and to force the page content to fill a large proportion of the image for best quality. This could be relaxed a bit by decreasing the size of the inner rectangle so that images could be captured slightly off to the side, but the edges of the page would still be in focus. Another refinement to help a user during alignment is for the framing lines to change color or disappear to indicate which edges are well-aligned.

Another issue raised in our experiments is that capture is slower in a dimly lit room. Originally, a single sharpness threshold was hardwired in the program. However, attaining a sharp image is much easier in bright light, where the shutter speed is faster, than in dim light. At the possible expense of capturing somewhat blurrier images, a slider could be added to the interface to allow users to set the sharpness threshold. Alternatively, the system could automatically adjust the threshold based on the best recent sharpness values. A minimum allowed sharpness could be used to prevent blurry images from being considered when computing a threshold. An example of the computation of minimal required sharpness is

[10]http://itunes.apple.com/us/app/jotnot-scanner-pro/id307868751

to compute a running average of the most recent sharpness values greater than the minimum sharpness:

$$thresh_{sharpness} = \frac{1}{N} \sum_{t=i-N}^{i} s(t) - \alpha$$

where $s(t)$ is the sharpness value of the t^{th} frame with a value greater than the minimum, N is the number of frames used in the estimate and set to a relatively small value, e.g., 3, and α lowers the $thresh_{sharpness}$ so that images can be captured more quickly, but may be blurrier.

We can also expand our work on automatic, high quality capture of the content of a document page to provide for efficient capture of multi-page documents. We note based on observation that as a user moves a camera and/or page into position while SmartDCap is working, the page content is not well framed and the image is not sharp enough for capture since movement results in blur and the varying distance between the camera and the page results in the image being mostly out of focus. We can extrapolate that similarly, as a page is being removed from view or turned over, the image will not be well-framed or sharp enough for capture. Based on this and evidence from our experiments that SmartDCap can take high quality shots automatically, we can optionally allow users to advance to the next page in the document immediately and wait to review all captured images as a group after scanning the whole document, which could increase scanning speed.

Finally, we may be able to improve the audio feedback in SmartDCap using earcons [1] or auditory icons [4] to communicate more specific directions to users during capture, such as "move left/right", "move closer", or "light too low".

CONCLUSIONS

People continue to take an increasing number of photos with mobile devices. Even so, capturing a sharp, well-framed page or region of a page is more challenging than capturing a scene. In this paper we presented the SmartDCap system for helping people capture better document images. SmartDCap provides real-time feedback on image quality and automatically captures an image when the quality measures are good. The measures are based on image sharpness and framing of document content, where the content can be either full pages or portions of page columns. SmartDCap makes it easier to capture good photos of document content by allowing a user to focus on framing while freeing them from having to simultaneously decide when sharpness and framing are good and then indicating that a photo should be captured. Our users rated SmartDCap at least as easy to use as the ICS camera application. Furthermore, rating of the captured images showed that the sharpness of the images captured by SmartDCap is significantly better under different lighting conditions and the framing is significantly better overall.

With SmartDCap, higher quality photos are captured *before* any post-processing is applied. Post-processing techniques that other capture applications offer, such as perspective and tone correction, can be applied to photos captured by SmartDCap to further improve the captured images. Results from our evaluations of SmartDCap led us to propose refinements to simplify its use and to propose how it can be used as the basis of a mobile document capture system that more efficiently captures multi-page documents.

REFERENCES

1. Brewster, S., Wright, P., and Edwards, A. An evaluation of earcons for use in auditory human-computer interfaces. In *Proc. of the INTERACT '93 and CHI '93 Conf. on Human Factors in Computing Systems*, ACM (1993), 222–227.
2. Carter, S., Adcock, J., Doherty, J., and Branham, S. Nudgecam: toward targeted, higher quality media capture. In *Proc. of the ACM Intl. Conf. on Multimedia* (2010), 615–618.
3. Ferzli, R., and Karam, L. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE Transactions on Image Processing 18*, 4 (2009), 717–728.
4. Garzonis, S., Jones, S., Jay, T., and ONeill, E. Auditory icon and earcon mobile service notifications: intuitiveness, learnability, memorability and preference. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM (2009), 1513–1522.
5. Ha, J., Haralick, R., and Phillips, I. Recursive x–y cut using bounding boxes of connected components. In *Proc. of the Intl. Conf. on Document Analysis and Recognition*, vol. 2, IEEE (1995), 952–955.
6. Hoggan, E., Crossan, A., Brewster, S., and Kaaresoja, T. Audio or tactile feedback: which modality when? In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM (2009), 2253–2256.
7. Jayant, C., Ji, H., White, S., and Bigham, J. P. Supporting blind photography. In *Proc. of the ACM SIGACCESS Conf. on Computers and Accessibility* (2011), 203–210.
8. Larson, E., and Chandler, D. Most apparent distortion: Full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging 19*, 1 (2012).
9. Lee, S., and Ryu, D. Parameter-free geometric document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1240–1256.
10. Liang, J., Doermann, D., and Li, H. Camera-based analysis of text and documents: a survey. *Intl. Journal on Document Analysis and Recognition 7*, 2 (2005), 84–104.
11. Liu, C., Huot, S., Diehl, J., Mackay, W., and Beaudouin-Lafon, M. Evaluating the benefits of real-time feedback in mobile augmented reality with hand-held devices. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM (2012), 2973–2976.
12. Narvekar, N., and Karam, L. A no-reference image blur metric based on the cumulative probability of blur detection (CPBD). *IEEE Transactions on Image Processing 20*, 9 (2011), 2678 –2683.

13. Nees, M., and Walker, B. Data density and trend reversals in auditory graphs: Effects on point-estimation and trend-identification tasks. *ACM Transactions on Applied Perception (TAP) 5*, 3 (2008), 13.

14. Pollard, S., and Pilu, M. Building cameras for capturing documents. *Intl. Journal on Document Analysis and Recognition 7*, 2 (2005), 123–137.

15. Shafait, F., Keysers, D., and Breuel, T. Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 6 (2008), 941–954.

16. Sheikh, H., Sabir, M., and Bovik, A. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing 15*, 11 (Nov 2006), 3440–3451.

17. Vázquez, M., and Steinfeld, A. Facilitating photographic documentation of accessibility in street scenes. In *CHI Extended Abstracts on Human Factors in Computing Systems*, ACM (2011), 1711–1716.

18. Walker, B., and Kramer, G. Mappings and metaphors in auditory displays: An experimental assessment. *ACM Transactions on Applied Perception (TAP) 2*, 4 (2005), 407–412.

19. White, S., Ji, H., and Bigham, J. Easysnap: Real-time audio feedback for blind photography. In *Adj. Proc. of the ACM Symposium on User Interface Software and Technology* (2010), 409–410.

20. Yu, Y., and Liu, Z. A user study of visual versus sonically-enhanced interfaces for use while walking. In *Proc. of the ACM Intl. Conf. on Multimedia* (2010), 687–680.