Cell Phone-based Wayfinding for the Visually Impaired

James Coughlan[1], Roberto Manduchi[2], and Huiying Shen[1]

[1] Smith-Kettlewell Eye Research Institute 2318 Fillmore Street
San Francisco, CA 94115
{coughlan,hshen}@ski.org
[2] University of California, Santa Cruz
1156 High Street
Santa Cruz, CA 95064
manduchi@soe.ucsc.edu

Abstract. A major challenge faced by the blind and visually impaired population is that of *wayfinding* – the ability of a person to find his or her way to a given destination. We propose a new wayfinding aid based on a camera cell phone, which is held by the user to find and read aloud specially designed machine-readable signs in the environment (labeling locations such as offices and restrooms). Our main technical innovation is that we have designed these machine-readable signs to be detected and located in fractions of a second on the cell phone CPU, even at a distance of several meters. A linear barcode printed on the sign is read using novel decoding algorithms that are robust to noisy images. The information read from the barcode is then read aloud using pre-recorded or synthetic speech. We have implemented a prototype system on the Nokia 7610 cell phone, and preliminary experiments with blind subjects demonstrate the feasibility of using the system as a real-time wayfinding aid.

1 Introduction

There are nearly 1 million legally blind persons in the United States, and up to 10 million with significant visual impairments. A major challenge faced by this population is that of *wayfinding* – the ability of a person to find his or her way to a given destination. Well-established orientation and mobility techniques using a cane or guide dog are effective for following paths and avoiding obstacles, but are less helpful for finding specific locations or objects.

We propose a new assistive technology system to aid in wayfinding based on a camera cell phone, which is held by the user to find and read aloud specially designed signs in the environment. These signs consist of barcodes placed adjacent to special landmark symbols (see Fig. 1). The symbols are designed to be easily detected and located by a computer vision algorithm running on the cell phone; their function is to point to the barcode to make it easy to find without having to segment it from the entire image. Our proposed system, which we have already prototyped, has the advantage of using standard off-the-shelf cellphone technology – which is inexpensive, portable, multi-purpose and becoming nearly ubiquitous – and simple color signs that can be easily produced on a standard color printer. Another advantage of the cell phone is that it is a mainstream consumer product which raises none of the cosmetic concerns that might arise with other assistive technology requiring custom hardware.

Fig. 1. (a) Camera cell phone held by blind user. (b) Color target next to barcode mounted on wall. The distinctive pattern of the color target allows the barcode to be rapidly localized, even against background clutter.

Our system is designed to operate efficiently with *current* cell phone technology using machine-readable signs. Our main technological innovation is the design of special landmark symbols, which we call *color targets*, that can be robustly detected and located in fractions of a second on the cell phone CPU, which is considerably slower than a typical desktop CPU. The color targets allow the system to quickly detect and read a linear barcode placed adjacent to the symbol. It is important that these symbols be detectable at distances up to several meters in cluttered environments, since a blind or visually impaired person cannot easily find a barcode in order to get close enough to it to be read.

Once the system detects a color target it guides the user towards the sign by providing appropriate audio feedback.

It is also important to be able to read the barcode from as far away as possible (without it being unacceptably large), so that the user does not have to get too close to it. We have devised novel algorithms for reading linear barcodes photographed by the camera cell phone under adverse – but not uncommon – conditions such as poor resolution, low light and image noise. These conditions are all the more serious because the cell phone camera is of considerably lower quality than a typical digital camera; low resolution, saturation and poor color fidelity are particularly problematic.

We have implemented a prototype system that works with any camera cell phone running the Symbian OS. The system is set up to guide the user towards signs using audio beeps, and reads aloud the sign information using pre-recorded speech (which will eventually be replaced by text-to-speech). Sign information can either be encoded directly as ASCII text in the barcode, or can encode a link to an information database (which is what our prototype does on a small scale). The signs are affixed to the walls of office building corridors to label such locations as particular office numbers and restrooms. Preliminary experiments with blind subjects demonstrate the feasibility of using the system as a real-time wayfinding aid (see Sec. 5).

2 Related Work

A number of approaches have been explored to help blind travelers with orientation, navigation and wayfinding, most using modalities other than computer vision. Among the most promising include infrared signage that broadcasts information received by a hand-held receiver [6], GPS-based localization (e.g. http://www.senderogroup.com), RFID labeling, and indoor Wi-Fi based localization (based on signal strength) and database access [11]. However, each of these approaches has significant limitations that limit their attractiveness as stand-alone solutions. Infrared signs require costly installation and maintenance; GPS has poor resolution in urban settings and is unavailable indoors; RFIDs can only be read at close range and would therefore be difficult to locate by blind travelers; and Wi-Fi localization requires extensive deployment to ensure complete coverage, as well as a time-consuming calibration process.

Research has been undertaken on computer vision algorithms to aid in wayfinding for such applications as navigation in traffic intersections [19] and sign reading [17]. The obvious advantage of computer vision is that it is designed to work with little or no infrastructure or modification to the environment. However, none of it is yet practical for commercial use because of issues such as insufficient reliability and prohibitive computational complexity (which is especially problematic when using the kind of portable hardware that these applications require).

Our approach, image-based labeling, is motivated by the need for computer vision algorithms that can run quickly and reliably on portable camera cell

phones, requiring only minor modifications to the environment (i.e. posting special signs). Image-based labeling has been used extensively for product tagging (barcodes) and for robotic positioning and navigation (fiducials) [4, 18, 16, 3, 14]. It is important to recognize that a tag reading system must support two complementary functionalities: detection and data embedding. These two functionalities pose different challenges to the designer. Reliable detection requires unambiguous target appearance, whereas data embedding calls for robust spatial data encoding mechanisms. Distinctive visual features can be used to maximize the likelihood of successful detection. Computational speed is a critical issue for our application. We argue that color targets have a clear advantage in this sense with respect to black and white textured patterns.

Variations on the theme of barcodes have become popular for spatial information encoding. Besides the typical applications of merchandise or postal parcel tagging, these systems have been demonstrated in conjunction with camera phones in a number of focused applications, such as linking a product or a flyer to a URL. Commercial systems of this type include the Semacode, QR code, Shotcode and Nextcode. An important limitation of these tags is that they need to be seen from a close distance in order to decode their dense spatial patterns. Our approach addresses both requirements mentioned above by combining a highly distinctive fiducial with a barcode (see Sec. 3 for details).

Direct text reading would be highly desirable, since it requires no additional environment labeling. Standard OCR (optical character recognition) techniques are effective for reading text against a blank background and at a close distance [21], but they fail in the presence of clutter [13]. Recently developed algorithms address text localization in cluttered scenes [2, 9, 12, 10], but they currently require more CPU power than is available in an inexpensive, portable unit: our preliminary tests show cell phone processing speed to be 10-20 times slower than that of a portable notebook computer for integer calculations (and slower still if floating point calculations are performed). Barcodes suffer from a similar limitation in that they must be localized, typically by a hand-held scanner, before they can be read. We note that our color target approach solves both the problems of quickly localizing barcodes or text and of extracting the specific information that is relevant to wayfinding.

3 Color Targets

We have designed the color targets to solve the problem of localizing information on signs. The targets are designed to be distinctive and difficult to confuse with typical background clutter, and are detectable by a robust algorithm that can run very quickly on a cell phone (i.e. up to 2 or more frames/sec. depending on resolution). Once the targets are detected, barcodes or text adjacent to them are easily localized (see Sec. 4). A variety of work on the design and use of specially designed, easily localized landmarks (i.e. fiducials) has been undertaken [3, 4], but to the best of our knowledge this is the first cell phone-based application of landmark symbols to the problem of environmental labeling.

We use a cascade filter design (such as that used in [20]) to rapidly detect the color target in clutter. The first filter in the cascade is designed to quickly rule out regions of the image that do not contain the target, such as homogeneous regions (e.g. blue sky or white wall without markings). Subsequent filters rule out more and more non-target locations in the image, so that only the locations containing a target pass all the filter tests in the cascade (with very few false positives).

3.1 Color for Rapid Search

Rather than rely on generic edge-like patterns – which are numerous in almost every image of a real scene – we select for a smaller set of edges: those at the boundaries of particular color combinations, identified by certain color gradients. Empirically we have found that red-green boundaries are comparatively rare in most scenes compared to the incidence of general edges of all colors. A small but significant number of red-green boundaries are still present in non-target regions. However, we have found that the combination of three colors in a particular spatial configuration (see Fig. 2) is characterized by multiple color gradients which together suffice to rule out almost all false positives.

Some form of color constancy is required if color is to be a defining feature of the target under varied illumination. One solution would be to pre-process the entire image with a generic color constancy algorithm [1], but such processing generally makes restrictive assumptions about illumination conditions and/or requires significant computational resources. Fortunately, while the appearance of individual colors varies markedly depending on illumination, color gradients tend to vary significantly less [8]. We exploit this fact to design a cascade of filters that threshold certain color gradient components. The gradients are estimated by computing differences in RGB channels among three pixels in a triangular configuration (Fig. 2(a)). The centroid of the three pixels, (x, y), is swept across the entire pixel lattice. Empirically, five sequential color gradient tests suffice to rule out all but a small fraction of the image pixels that do not lie on a color target.

The choice of gradient thresholds, and the sequence these tests appear in the cascade, were chosen empirically by taking pictures of the color target under a variety of indoor and outdoor lighting conditions: sunlight, shade, fluorescent light and tungsten (incandescent) lighting (Fig. 2(b)). Then for each color region in each picture the mean and variance of red, green and blue channels were calculated. Under the assumption that, for a given image, the colors in different patches ((R_i, G_i, B_i) denotes the color at point p_i) are uncorrelated, the variance of the difference (say, of $R_1 - R_2$) is equal to the sum of the variances in each patch (the mean of $R_1 - R_2$ is always equal to the difference of the means in the two patches in the same image). The $\pm 1\sigma$ confidence interval of each difference for each image was plotted. In practice we set $T = 20$ as a conservative threshold for each filter in the cascade to handle low lighting conditions. Separate thresholds for each test will be investigated in future research to optimize the detection vs. false positive rate.

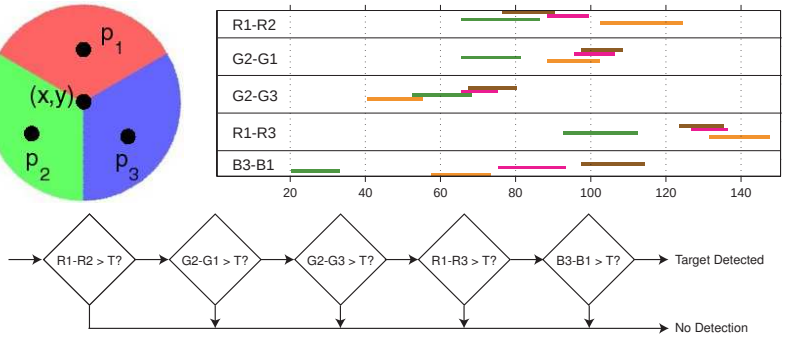

Fig. 2. Top left: Sample point locations p_1, p_2 and p_3 in red, green and blue regions are defined by fixed offsets relative to the center (x,y). Top right: Empirical confidence intervals of RGB channel differences between pairs of regions under different lighting conditions. Legend: brown = direct sunlight; magenta = shade; green = fluorescent light; and orange = tungsten (incandescent) lighting. Bottom: filter cascade, a sequence of five inequality tests based on RGB values at p_1, p_2 and p_3.

Additional post-processing is needed to rule out the few false positives that survive the filter cascade. In our preliminary experiments we have implemented a grouping procedure that classifies sufficiently large clusters of on-target pixels as belonging to a single target (most false positive pixels are scattered sparsely throughout the image). Since relatively few pixels are classified as on-target in most images, the grouping procedure adds negligible cost to the overall target detection algorithm. If necessary in the future we will implement additional tests to use a higher-level model to verify each candidate target (e.g. based on fitting a circle or ellipse to the edges of the hypothesized three-color region).

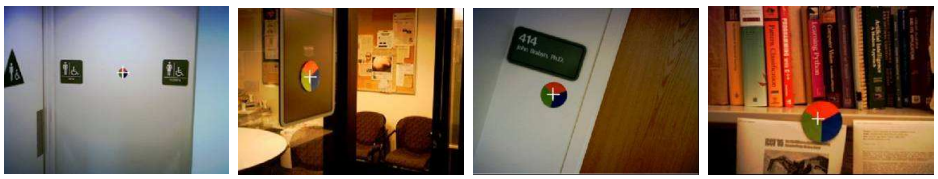

Fig. 3. Images with detection results drawn as white crosses superimposed on targets.

We implemented this simple prototype algorithm in C++ on a Nokia 7610 cell phone running the Symbian 7.0 OS. The camera in the phone has a maximum resolution of 1152 by 864 pixels. The algorithm detects multiple targets in a fraction of a second to about half a second (depending on camera resolution). The detection is invariant to a range of scales (from about 0.5 m to as far as 10 m), and accommodates significant rotations (up to about 30 degrees in the

camera plane), slant and background clutter. See Fig. 3 for examples of detection results.

Note that the color targets need to be well illuminated to be detected, or else image noise will obscure the target colors. One way to overcome this limitation might be to operate a flash with the camera, but this approach would use significant amounts of battery power, would fail at medium to long range, and would be annoying to other people in the environment. Another possibility might be to increase the exposure time of the camera, but this would make the images more susceptible to motion blur; similarly, increasing the camera gain would increase pixel noise as well as the brightness. Overall, it seems most practical to site the targets at locations that are already well lit.

3.2 Theoretical and Empirical Bounds

Maximum Detection Distance (Stationary) The width of the color target, together with the resolution and the field of view (FOV) of the camera, determine the maximum distance at which the target can be detected. For the Nokia 7610 cell phone, the instantaneous horizontal FOV (IFOV) of a single pixel is approximately 1.5 mrads for the 640×480 resolution, and 0.82 mrads for the 1152×864 resolution. The pixels can be considered square to a good approximation. In order to detect a target at a distance d, it is necessary that all three color patches be correctly resolved. The color at a pixel location, however, is computed by interpolation from the underlying Bayer mosaic, which typically involves looking at color values within a 3×3 window centered at the pixel. This means that, in order to correctly measure the color of a patch, the patch must project onto a square of at least 3×3 pixels, so that at least one pixel represents the actual patch color. In fact, we found out that as long as at least half of the pixels within the 3×3 window receive light from the same color patch, detection is performed correctly.

Now suppose that two measurement pixels are separated by a buffer zone of M pixels as in Fig. 4. In our implementation, we chose $M = 7$. The importance of these buffer pixels in the context of motion blur will be discussed in Sec. 3.2. It is clear from Fig. 4 that the diameter D of the color target should project onto at least $M + 4$ pixels for color separation. This is obviously an optimistic scenario, with no blurring or other forms of color bleeding and no radial distortion. In formulas, and remembering that the tangent of a small angle is approximately equal to the angle itself:

$$d \leq \frac{D}{\mathrm{IFOV} \cdot (M + 4)} \tag{1}$$

We have considered two target diameters in our experiments, $D = 6$ cm and $D = 12$ cm. Tab. 1 shows the theoretical bounds, computed using (1), as well as empirical values, obtained via experiments with a color target under two different incident light intensities (175 lux and 10 lux respectively). A lower detection distance may be expected with low light due to increased image noise. The maximum distances reported in the table include the case when no post-processing is performed (see Sec. 3.1). This provides a fairer comparison with

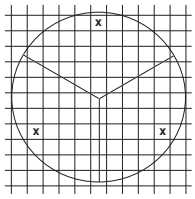

Fig. 4. The layout of a 3-patch color target, with the location of the "probing" pixels. The lower two pixels are separated by a buffer of $M = 7$ pixels.

the model of Fig. 4, which only requires one point triplet detection. Of course, postprocessing (which is necessary to reject false positives) reduces the maximum detection distance, since it requires that a certain number of triplets is found. The experiments were conducted while holding the cell phone still in the user's hand. Note that the experimental values, at least for the case of well lit target and without postprocessing, do not differ too much from the theoretical bounds, which were obtained using a rather simplistic model.

	$D = 6$ cm			$D = 12$ cm		
	Theor.	Exp. - No PP.	Exp. - PP.	Theor.	Exp. - No PP.	Exp. - PP.
640×480	3.6	3.5 (3)	2.7 (2.4)	7.2	6.7 (5.7)	5.6 (4.7)
1152×864	6.6	5.5 (4.5)	4.3 (3)	13.2	11.1 (8.5)	9.3 (6.4)

Table 1. Maximum distances (in meters) for color target detection. Theoretical bounds are reported together with experimental values with and without the postprocessing (PP.) module. Values in the case of poor illumination are shown within parentheses.

Maximum Detection Distance (Panning) Searching for a color target is typically performed by pivoting the cell phone around a vertical axis (panning) while in low-res (640×480) mode. Due to motion, blur may and will arise, especially when the exposure time is large (low light conditions). Motion blur affects the maximum distance at which the target can be detected. A simple theoretical model is presented below, providing some theoretical bounds.

Motion blur occurs because, during exposure time, a pixel receives light from a larger surface patch than when the camera is stationary. We will assume for simplicity's sake that motion is rotational around an axis through the focal point of the camera (this approximates the effect of a user pivoting the cell phone around his or her wrist). If ω is the angular velocity and T is the exposure time, a pixel effectively receives light from a horizontal angle equal to IFOV $+ \omega T$. This affects color separation in two ways. Firstly, consider the vertical separation between the two lower patches in the color target. For the two lower probing pixels in Fig. 4 to receive light from different color patches , it is necessary that

the apparent image motion be less than $\lfloor M/2 \rfloor - 1$[3] (this formula takes the Bayer color pattern interpolation into account). The apparent motion (in pixels) due to panning is equal to $\omega T/\text{IFOV}$, and therefore the largest acceptable angular velocity is $(\lfloor M/2 \rfloor - 1) \cdot \text{IFOV}/T$. For example, for $M = 7$ and $T = 1/125$ s, this corresponds to 21.5°/s. The second way in which motion blur can affect the measured color is by edge effects. This can be avoided by adding a "buffer zone" of $\lceil \omega T/\text{IFOV} \rceil$ pixels to the probing pixels of Fig. 4. This means that the diameter of the target should project onto $M + 2 \cdot (2 + \lceil \omega T/\text{IFOV} \rceil)$ pixels. Hence, the maximum distance for detection decreases with respect to the case of Sec. 3.2.

In fact, these theoretical bounds are somewhat pessimistic, since a certain amount of motion blur does not necessarily mean that the target cannot be recognized. In order to get some more realistic figures, we ran a number of experiments, by pivoting the cell phone at different angular velocities in front of a 12 cm target from a distance of 2 meters. Since we could neither control nor measure exposure time, comparison with the theoretical bounds is difficult. When the color target was lit with average light intensity (88 lux), detection was obtained with probability larger than 0.5 at angular speeds of up to 60°/s. With lower incident light (10 lux), this value was reduced to 30°/s, presumably due to larger exposure time.

Detection Speed The rate at which target detection is performed depends on two factors: the image acquisition rate, and the processing time to implement the detection algorithm. Tab. 2 shows the rates attained with and without processing and display (in the viewfinder). Image display is obviously not necessary when the system is used by a blind person, but in our case it was useful for debugging purposes. Note that image display takes 44% of the time in the VGA detection loop. If the images are not displayed, the frame rate in the VGA resolution mode is of more than 2.5 frame per second. However, for the high resolution case, image acquisition represents a serious bottleneck. Even without any processing, the acquisition/display rate is of 21 frames per minute. When processing is implemented (without display), the rate is of 20 frames per minute.

	no proc./displ.	proc./displ.	proc./no displ.
640×480	114	110	154
1152×864	21	19	20

Table 2. Rates (in frames per minute) attained for different image resolutions with and without target detection module(proc./no proc.) and with and without display in the viewfinder (disp./no displ.).

Given the extremely low acquisition rate for high resolution images provided by this cell phone, we us the following duty cycle strategy. The scene is searched

[3] The symbol $\lfloor \cdot \rfloor$ represents the largest integer smaller than or equal to the argument.

using VGA resolution. When a target is detected over a certain number M (e.g., $M = 5$) of consecutive frames, a high resolution snapshot is taken. Bar code analysis is then implemented over the high resolution data. The number M of frames should be large enough to allow the user to stop the panning motion, thereby stabilizing the image and reducing the risk of motion blur when reading the bar code.

4 Barcodes

The barcode is a time-tested visual means for conveying information through spatial patterns. The Universal Product Code (UPC) barcode represents digits as sequences of four consecutive variable-width bars with alternating color. More recently, a wide variety of 2-D barcodes have been proposed (e.g., SemaCode, www.semacode.org; Datamatrix, www.tec-it.com). Some of these codes have been designed expressly for a camera phone (e.g., ShotCode, www.shotcode.com; ColorCode, www.colorzip.co.jp/en). The vast majority of proposed tags are intended for either inventory/labeling or for product promotion. In the first case, dedicated hardware is normally used for tag reading. In the case of product promotion, a user is supposed to take a picture of the tag from a relatively short distance.

A barcode for our application should convey necessary information while being easily readable from a camera phone. Detectability of the barcode, a foremost problem for automated systems [7], is solved by means of our color target discussed earlier. Once the color target has been detected, the barcode is found within a "reading window" near it. The number of bits that need to be encoded in the barcode is highly variable. Only a few bits (4-5) are needed to specify a category of locations, for example whether the tag signals an elevator, an emergency exit, or a restroom. More bits are required to encode a descriptive string (e.g., the name of an office's occupant) or a link such as a URL.

For our initial experiments we used a simplified 1-D barcode. These codes have low efficiency (in terms of bits per surface area), but this redundancy can be exploited for increased robustness and of speed of reading. In particular, 1-D barcodes can be read by analyzing the brightness pattern along a line within a certain range of orientations, and multiple line readings can be used for validation. The next subsection describes a simple, fast and robust algorithm for 1-D barcode reading that only uses fixed point computation (i.e. C++ integer math without any floating point computations).

4.1 Fast and Robust 1-D Barcode Reading

The barcodes used in our experiments are a simplified version of the standard UPC codes. We use only 2 bar widths, with each bar encoding 1 bit. In addition, we don't constrain the first bar to be of a fixed (narrow) width, thereby gaining 1 bit. Fixing the width of the first bar allows UPC codes to be read by a serial scanning machine, which uses the first bar for calibration. Instead, our system

learns the apparent bar widths in the image directly by the statistics of bar widths, under the assumption that there is at least one narrow and one wide bar.

The two main steps of our 1-D barcode reading are: (1) Determination of each bar's width; (2) Determination of the correct bar width threshold for classifying each bar as "narrow" or "wide". For the first step, we have tried with two different algorithms, obtaining similar results. Note that we can safely assume that, as long as the camera is roughly horizontal, there is no need for re–orienting the barcode, and reading can be performed along image rows. The first algorithm identifies the maxima of the brightness derivatives along each row, and retains only the maxima above a certain threshold. The sign of the derivative (polarity) gives indication of whether a white or black is to the right of the maximum location. The second algorithm uses a version of Niblack's algorithm for image binarization [15]. At each row within the reading window, we compute local brightness mean and covariance on a 1-D moving window with size of 32 pixels. When the variance is larger than a fixed threshold (in our case, 800), the mean is used as a brightness threshold for binarization. Otherwise, the pixel is simply binarized to 0. An example of binarization is shown in Fig. 5 (b).

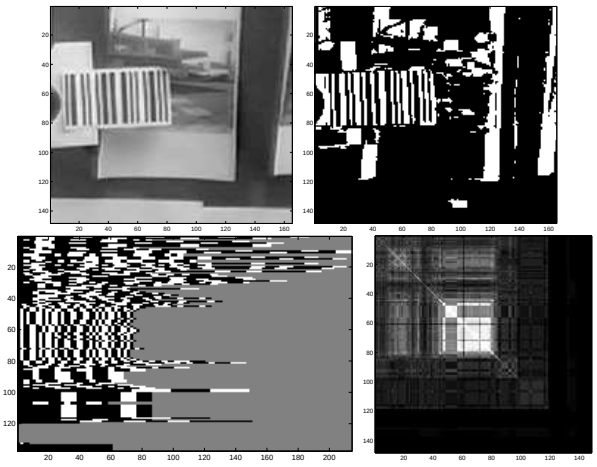

Fig. 5. Illustration of stages of barcode reading algorithm. Top left (a): Original image. (Note that, in practice, smaller reading windows with size of 40 by 200 pixels are used.) Top right (b): Binarized image. Bottom left (c): detected bars, with widths quantized to "narrow" and "wide" according to chosen threshold (which is estimated independently for each row). Only the first $N = 27$ bars are considered. Bottom right (d): Matrix $M_{i,j}$ as described in the text. (In this example the barcode was decoded correctly.)

For each row within the reading window, we then compute the length of each run of consecutive 0 (black) and 1 (white) pixels. The main assumption here is that the first bar is separated from the color target by a small patch, and

that the colors in the target are binarized to 0. Hence, the first (black) bar in a left–to–right scanning must be preceded by at least one white pixel. Only the width for at most N consecutive runs of pixels need to be computed, where N, the number of bars in the pattern, is known in advance.

The next step consists of the estimation of the widths of the narrow bars (W_n) and of the wide bars (W_w), whose ratio is known ($W_w = 3W_n$ for our patterns). We use a simple maximum likelihood procedure for this. We define the un–normalized negative log–likelihood of the bar ensemble given a candidate bar width W_n as follows:

$$L(W_n) = \sum_{i=1}^{N} \min\left(|W(i) - W_n|, |W(i) - 3W_n|\right) \tag{2}$$

where $W(i)$ is the width of the i–th bar in the current row. The value W_n that minimizes $L(W_n)$ represents the most likely width for the narrow bars. Accordingly, the bar width threshold is set to $0.5(W_n + W_w) = 2W_n$. The search for the optimal width W_n is performed exhaustively over a fixed interval ([1,20] in our case). Since the bar widths can only take integer values, we use a step equal to 1/3 of a pixel (which translates to a step of 1 pixel for W_w). Fig. 5 (c) shows for each row the detected bars, with widths quantized to "narrow" and "wide" according to the chosen threshold.

At this point, each row in the reading window is characterized by a binary string (with 0 and 1 corresponding to narrow and wide bars respectively) with at most N bits. We should expect that some (possibly more than half) of the rows may contain incorrect values, either because the row was partially or entirely outside the barcode pattern, or because of binarization and/or bar thresholding errors. The next step consists of the determination of the "optimal" binary string based on a maximum voting procedure. More precisely, we consider each pair of strings (for rows i and j) and compute the number $M_{i,j}$ of corresponding bits with matching values. (This is related to the Hamming distance between the two strings.) The string of index i with the maximum cumulative value $\sum_{j \neq i} M_{i,j}$ is chosen as the best representative of the string ensemble. An example of the matrix $M_{i,j}$ is shown in Fig. 5 (d).

This algorithm is very fast, and is applied only on the reading window, which in our experiments had size of 40 by 200 pixels. The added computational cost is negligible. The main weakness of the algorithm is that it relies entirely on a correct brightness binarization step. Poor binarization may incorrectly join three consecutive bars into one (typically due to blurring) or split one bar into three (typically due to noise). As long as such phenomena are local, they are rejected by our inter–row validation procedure. Another concern is that bar widths are quantized to an integer number of pixels. This may determine an incorrect bar width threshold determination when the pattern is seen from a distance (and thus bars have a short apparent widths). We plan to implement a binarization procedure that allows for subpixel localization of bar edges.

5 Experiments with Blind Subjects

We have conducted three proof-of-concept experiments to demonstrate the feasibility of a blind person using a cell phone to obtain location information using a color target. These experiments were performed by three blind volunteers who were informed of the purpose of the experiments. To guide the subjects towards color targets using the system, we devised a simple three-pitch audio feedback strategy: low, medium or high tones signified the target appearing in the left, center or right part of the camera's field of view, respectively, and silence signified that no target was visible to the system. We used 640×480 camera resolution in the experiments, which allowed the system to process a few frames per second. In addition, we performed the experiments with a modified color target based on four color patches rather than three (white, green, black and red); this was done because we found that the low light levels which prevailed in the experiment made it difficult to capture the blue patch in the original target. (We expect that the results of Sec. 3.2 will be similar for the modified target, and we will investigate several color target pattern variants in future research.)

In the first experiment, a blind subject was seated near the wall of a conference room, and a color target was placed on another wall in one of four possible locations relative to the subject: left, far left, right or far right. For each trial, the color target location was chosen at random by the experimenter. The subject was asked to use the cell phone target detector to identify which location the target was in. After a practice session, he identified the location for ten consecutive trials without making any mistakes.

A second experiment featuring a more challenging task was conducted with another blind subject, testing his ability to locate, walk to and touch a color target in an unknown location. For each trial, the experimenter placed a color target at a random location (at shoulder height) on the walls of an obstacle-free conference room approximately 7 meters by 5 meters in size. Beginning in the center of the room, the second blind subject used the color target detector to find the color target, walk towards it and touch it. In 20 trials (after a training session), it took him anywhere from 13 to 33 seconds to touch the target - significantly faster than if he had searched the perimeter of the room for the target by feel alone (which would have taken on the order of minutes rather than tens of seconds). In 19 of the 20 trials he touched only the target, while in one trial he first touched the wall several inches away before reaching the target.

The third experiment was designed to test the ability of a blind subject to find locations of interest in an office corridor using the color target system. The goal was for him to find four locations along either wall of a straight corridor (about 30 meters long) using the cell phone system; he was advised that the labels did not correspond to the actual building layout, so that his familiarity with the building would not affect the outcome of the experiment. A color target with 5-bit bar code was affixed at waist height near each of the four locations in the corridor. The bar codes encoded four separate numbers which were associated with four pre-recorded sounds ("elevator," "restroom," "room 417" and "staircase"). The subject was instructed to walk along the corridor to scan for all four labels.

When the camera was close enough to the color target, the bar code next to it was read and the appropriate recording was played back. After a training session, the labels were moved to new locations and the subject was able to find all four of them in about two minutes. No false positives or false bar code readings were encountered during the experiment.

While these experiments are preliminary, they show that blind subjects are able to use a simple cell phone interface to locate signs – at a range of distances – and that they are capable of orienting the camera properly and moving it smoothly and slowly enough to prevent interference from motion blur. These results provide direct evidence of the feasibility of our proposed system.

6 Conclusion

We have demonstrated a camera cell phone-based wayfinding system that allows a visually impaired user to find and read signs marked with color targets and barcodes. A key challenge of the system is the limited computational power of the cell phone, which is about 10-20 times slower than a typical notebook computer. Our solution is to place a distinctive color target pattern on the sign, which may be rapidly detected (using integer arithmetic) even in cluttered scenes. This swiftly guides the system to an adjacent barcode, which we read using a novel algorithm that is robust to poor resolution and lighting. Preliminary experiments with blind subjects confirm the feasibility of the system.

A priority for future work is to detect and read signs at greater distances. This task will become easier as higher resolution cell phone cameras become available. In addition, we will investigate variations on the color target design and improvements that will allow color target detection algorithms to reliably detect color targets with smaller apparent size (i.e. at a greater distance). We will also explore ways of increasing the information density of barcodes, such as the use of 2-D barcodes. Finally, we will test the system with more blind and low vision subjects to determine the most effective user interface, which may combine the use of vibratory signals, audio tones and synthesized speech.

7 Acknowledgments

We would like to thank Alessandro Temil for valuable assistance with Symbian C++ programming. J.C. and H.S. were supported by the National Institute on Disability and Rehabilitation Research (grant no. H133G030080), the NSF (grant no. IIS0415310) and the National Eye Institute (grant no. EY015187-01A2).

References

1. D.H. Brainard and W.T. Freeman. “Bayesian Color Constancy.” J. Opt. Soc. Amer.-A, 14:1393–1411, July 1997.

2. X. Chen and A. L. Yuille. “Detecting and Reading Text in Natural Scenes.” CVPR 2004.
3. Y. Cho and U. Neumann. ”Multi-ring color fiducial systems for scalable fiducial tracking augmented reality”. In: Proc. of IEEE VRAIS. 1998.
4. D. Claus and A.W. Fitzgibbon. (2004) ”Reliable Fiducial Detection in Natural Scenes”, in Proc. ECCV, 2004.
5. J. Coughlan, R. Manduchi, M. Mutsuzaki and H. Shen. “Rapid and Robust Algorithms for Detecting Colour Targets.” 10th Congress of the International Colour Association, AIC Colour ’05, pp. 959 - 962. Granada, Spain. May 2005.
6. W. Crandall, B. Bentzen, L. Myers and J. Brabyn. New orientation and accessibility option for persons with visual impairment: transportation applications for remote infrared audible signage. Clinical and Experimental Optometry 2001 May, 84(3): 120-131.
7. M. Fiala. ”ARTag, A Fiducial Marker System using Digital Techniques.” IEEE Proc. CVPR. San Diego, June 2005.
8. Funt, B.V., and Finlayson, G., ’Color Constant Color Indexing,’ IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (5), 522-529, May 1995.
9. J. Gao and J. Yang. “An Adaptive Algorithm for Text Detection from Natural Scenes.” CVPR 2001.
10. A.K. Jain and B. Tu. “Automatic Text Localization in Images and Video Frames.” Pattern Recognition. 31(12), pp 2055-2076. 1998.
11. A.M. Ladd, K.E. Bekris, A.P. Rudys, D.S. Wallach and L.E. Kavraki. ”On the feasibility of using wireless ethernet for indoor localization”, IEEE Trans. On Robotics and Automation, 20(3):555-9, June 2004.
12. H. Li, D. Doermann and O. Kia. Automatic text detection and tracking in digital videos. IEEE Transactions on Image Processing, 9(1):147–156, January 2000.
13. J. Liang, D. Doermann and H. Li. ”Camera-based analysis of text and documents: a survey”, International Journal on Document Analysis and Recognition, 7:84-104, 2005.
14. L. Naimark and E. Foxlin. (2002) Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: ISMAR.
15. M. Rohs, “Real–World Interaction with a Camera–Phone”, Second International Symposium on Ubiquitous Computing Systems (UCS 2004), Tokyo, Japan.
16. D. Scharstein and A. Briggs. (2001) ”Real-time recognition of self-similar landmarks”. Image and Vision Computing 19, 763-772.
17. P. Silapachote, J. Weinman, A. Hanson, R. Weiss and M. A. Mattar. “Automatic Sign Detection and Recognition in Natural Scenes.” IEEE Workshop on Computer Vision Applications for the Visually Impaired (CVAVI ’05), in conjunction with CVPR ’05. San Diego, CA. June 20, 2005.
18. A. State et al. (1996) ”Superior augmented reality registration by integrating landmark tracking and magnetic tracking”. In: SIGGRAPH429-438.
19. M.S. Uddin and T. Shioyama. “Bipolarity- and Projective Invariant-Based Zebra-Crossing Detection for the Visually Impaired.” IEEE Workshop on Computer Vision Applications for the Visually Impaired (CVAVI ’05), in conjunction with CVPR ’05. San Diego, CA. June 20, 2005.
20. P. Viola and M. Jones. ”Rapid Object Detection using a Boosted Cascade of Simple Features.” CVPR 2001. Kauai, Hawaii. December 2001.
21. A. Zandifar, P.R. Duraiswami, A. Chahine and L.S. Davis. ”A Video Based Interface to Textual Information for the Visually Impaired”, Fourth IEEE International Conference on Multimodal Interfaces (ICMI’02).