Chan, Kee-Yip
Advisor: Roberto Manduchi
UC Santa Cruz, Summer 2007, SURF-IT Grant number NSF CCF-0552699

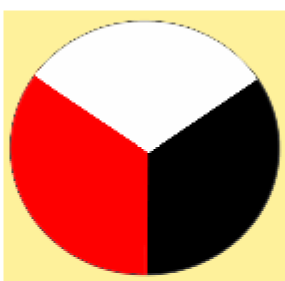
Continuation of Work on the Color Target for the Blind Cam Project

General Problem

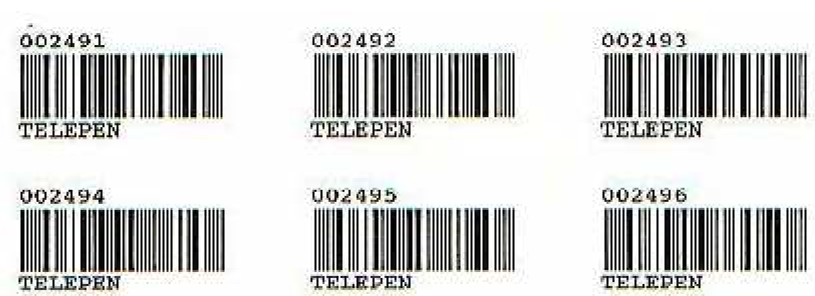
The Blind Cam project seeks to solve the problem of way-finding for blind people in office environments. Currently, a blind person must either stumble on his own or ask for help; unfortunately, it's not always convenient to find someone willing to help, and it is potentially hazardous for a blind person to stumble around. Our solution is to create a smart-phone application that can use the built-in camera to locate labels in the office environment, and report, in a synthesized voice, the messages that such labels convey. For ease of experimentation bar-codes were chosen because the algorithms for decoding bar-codes are much more efficient than those for alphabetical characters. But to decode the bar-code, it is necessary to first find its location. This process is made more efficient by placing a color target next to the bar-code.

Color Target

The color target simplifies localization because it is a fixed pattern of colors whereas the stripes of bar-codes vary in thickness for each unique message. Instead of employing a complicated algorithm to account for the variability of bar-code messages, the fixed design of the color target facilitates a much simpler algorithm.



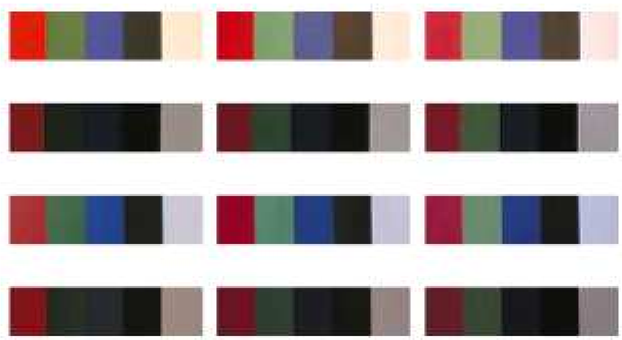
The fixed design of the color target simplifies localization.



bar-code messages are unique, making localization difficult.

Speed gains from simpler, more efficient algorithms are particularly important for reporting real-time information to the blind user. For example, the bar-code must first be found before the blind user can be alerted of his aiming accuracy; if this process is slow, the user may receive outdated information about his aim. Thus, speed gains from localizing the color target rather than the bar-code improves user interaction. As a consequence of printing the bar-code at a fixed distance and direction from the color target, finding the bar-code is a trivial step after finding the color target. In our trials, the bar-code is at a fixed distance to the right of the color target.

The algorithm for finding the color target filters the captured image for positive points. These points each represent the center of a color target in the environment, and are found by aligning a template to each point's neighboring pixels. The template matches color gradients rather than textured patterns or perceived colors because the color gradients of a scene have been observed to vary less than its perceived colors under slight lighting variations; color gradients are also computationally cheaper to analyze than textured patterns **(Coughlan, Manduchi)**.



Color gradients vary less than perceived colors.

Reporting Information

Our current implementation of the system selects a representative point from the handful of candidate points so that information relevant for user interaction can be extracted. When a single color target is expected to be located, relevant information that should be reported to the user are his aiming accuracy and distance to the color target.

The user's aiming accuracy is computed as the distance from the representative point to the center of the captured image, with accuracy rising as the point gets closer to the image center. We decided to interactively report this information to the user using an audio modality, where the pitch and volume of a short audio tone gets more intense as the user's aim gets better. It was found that this modality is quite effective; in our trials, we increased the pitch and volume by a constant amount whenever the aim changes, but varying the rate of change in pitch and volume is a task that warrants further experimentation.

In our trials a square audio tone of fifty milliseconds is played in unison to changes in aim, but further experimentation with the length and type of waveform is also warranted. We found that with the camera capturing at a low resolution (160 by 120 pixels), a pleasant change in audio pitch and volume at a rate of roughly three to four frames a second was achieved. Capturing at VGA resolution (640 by 480 pixels) resulted in a drop to roughly two frames per second, and the impression of real-time audio response was noticeably reduced. Dealing with the speed limits imposed by various image capture resolutions supported by the camera (the Nokia N70 phone does not support a resolution of 320 by 240 pixels, for example), is an important future task that involves consideration of the user's proposed environment and usage habits.

Assuming that the distance to the color target has been computed, the modality for reporting this information to the user is much more subjective than that of reporting aiming accuracy. For example, pitch might be used exclusively to report aim, while volume is used for reporting distance; or, it might be more appropriate to first get a good aim, followed by a synthesized voice reporting the distance in approximate meters.

This summer, the task of computing and reporting distance was left undone. Computing distance involves finding the size of the color target in the captured image and comparing the computed size with a predetermined base measurement of distance and color target size **(Coughlan, Manduchi)**. The size of the color target can be computed by several means, of which segmentation and edge detection are two candidates we intend to try in future experiments.

Development Tools and Programming

It is important to choose appropriate development tools for smart-phones, particularly because the capabilities and types of smart-phones vary widely. We chose to target the Symbian S60 platform because it is currently the world's market leader **(http://www.symbian.com/about/fastfacts/fastfacts.html)**. The S60 platform provides an integrated development environment called Carbide.C++, with C++ being the most supported programming language for Symbian S60 smart-phones.

Still, developing for smart-phones is difficult because of the limitations imposed by hardware and the development platform. For example, the smart-phone does not perform well with floating-point computations, and variable camera capabilities of smart-phone models limit the image capturing resolution. The S60 platform also provides asynchronous subsystems that must be properly managed. Our application utilizes three asynchronous subsystems: audio, camera, and TTS; the rest of the application consists of the S60 framework and our color target module.

Some problems faced early in the summer were due to careless usage of the S60 platform's feature set. For example in a previous version of our application, the audio component crashed frequently because requests for playback were made without waiting for the audio subsystem's asynchronous procedures to finish properly. To properly play a sound, the audio subsystem must first cancel all current playback, prepare the sound buffer, then only upon receiving notification that the preparation was successful should a request be made to start playback of a new sound. Also, it is very important to be careful in managing memory. Another problem encountered in previous versions of our application involved unrecognized ownership of the memory occupied by captured image data. When the camera hands off an image to our color target module, the color target module is the new owner of the image data, which must be freed at appropriate moments.

It is important also to understand the limitations of the subsystems used because potential problems are not necessary obvious, and the documentation can at times be sparse, so experimentation is important. For example, if an audio tone is longer in duration than the frame-rate, an abrupt noise occurs when the beginning of a frame necessitates a change in pitch or volume. A short-term solution is to use audio tones that are shorter in duration than the frame-rate, but a possible improvement would be to smooth the transition in pitch or volume. Although we use the Symbian S60 platform for developing our application, it may still be beneficial to try other development platforms and smart-phones.

Bibliography

1. Coughlan, James. Manduchi, Roberto. *Color Targets: Fiducials to Help Visually Impaired People Find Their Way by Camera Phone*

2. http://www.symbian.com/about/fastfacts/fastfacts.html