Real-time Computer Vision for Heading Correction in Mobile Augmented Reality Registration on Wind Farms

James Juett
University of Michigan – Ann Arbor
Computer Science & Engineering
jjuett@umich.edu

Georg Essl
University of Michigan – Ann Arbor
Computer Science & Engineering and Music
gessl@eecs.umich.edu

ABSTRACT
Accurate registration is critical for the precise location of information overlaid on a camera image in high quality augmented reality (AR) applications. We address outdoor mobile augmented reality in the context of wind farm maintenance and present a real-time computer vision-based algorithm that provides accurate registration by correcting error in the compass heading. By using a sub-sampled edge-finding algorithm based on known geometric features of the turbine towers and adaptively confining the search range of heading corrections, we achieve interactive rates on a contemporary mobile tablet device.

Author Keywords
Mobile Augmented Reality, Computer Vision

ACM Classification Keywords
H5.1. Artificial, augmented, and virtual realities. I.2.10 Vision and Scene Understanding. I.4.3 Registration. C.5.3 Portable devices.

General Terms
Algorithms, Performance, Design

INTRODUCTION
We are addressing the problem of augmented reality (AR) registration for mobile interactions in the context of wind farm maintenance tasks. In Augmented Reality, *registration* refers to the process of establishing a correspondence between real and virtual coordinate systems. This correspondence must be highly accurate to maintain the illusion that virtual information and/or objects coexist with the real world. Achieving this requires either very small error in sensed position and orientation or a closed loop method to correct for these errors given the current view of the real environment [1].

Methods for combining GPS, accelerometer, compass, and gyroscope data in order to facilitate AR registration are well known [2] and these sensors are included on many modern mobile devices. GPS allows determination of the user's position while readings from the remaining sensors can be fused to detect the user's orientation. These combined with the intrinsic parameters of the device's camera are sufficient for registration.

Errors in sensed data can prevent highly accurate registration. GPS error has been experimentally determined to be less than 3m in open sky environments for several

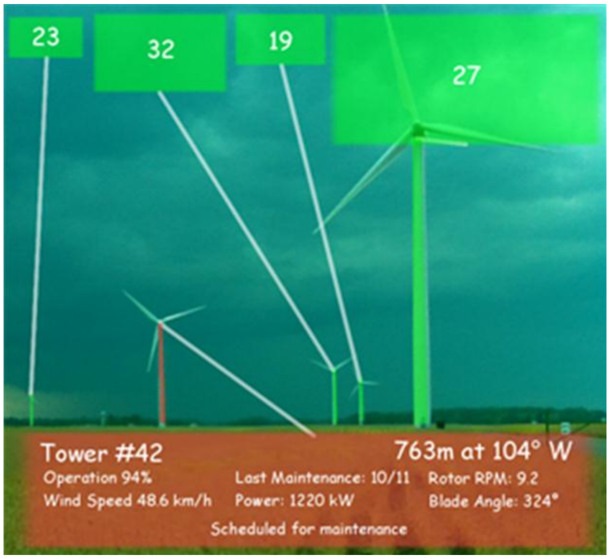

Figure 1. An example view from our prototype wind farm AR application. The data shown here is for illustrative purposes only and does not reflect actual turbine information.

consumer-grade GPS devices [3]. For objects viewed at a large distance (as is the case with wind turbines), it has been shown that the relative contribution of GPS errors of this magnitude is small [4]. Accelerometer and compass data contain noise, but the fusion with gyroscope data can compensate for and largely eliminate these problems [5]. It is straightforward to compensate for magnetic declination depending on a user's location.

More problematic are small scale variations in the local magnetic field around the device that create bias in the detected compass heading. A number of sources of interference (geologic or man-made) are present regardless of sensor quality and are quite difficult to correct for [4]. The focus of this work is to find appropriate heading corrections from live camera images as the system operates. In the particular context of AR on a wind farm, we use wind turbines as landmarks and exploit their known location and geometric features. We implemented a working prototype on an Apple iPad2.

RELATED WORK
There has been extensive work in the area of developing augmented reality solutions in the mobile setting [2]. The registration problem is well recognized as key to achieving

quality augmented reality [1], along with the challenges of making augmented reality work in outdoor settings [4].

Two key issues for achieving accurate registration are accurately finding an initial correspondence and then preserving that correspondence while moving the device. Previous work addressing the latter has focused on elimination of drift in order to accurately track relative orientation. Drift stabilization in gyroscopes has been addressed in [6] by incorporating vision data. Another approach is auto-calibration of the compass [7] in order to map relative magnetic distortions as users move from place to place.

Unfortunately, finding an initial accurate correspondence is made difficult because compass data is often inaccurate. Computer vision algorithms which detect landmarks whose position and structure are known provide an alternative means of addressing the registration problem. A common method for doing this is detecting fiducial markers prepared ahead of time, but this is impractical in a large outdoor setting.

In urban environments, registration has been addressed through image matching strategies. Coors et al. [8] describe a system of comparing pre-rendered images of a 3D modeled city scene against camera data and through matches determine the mobile user's position and registration within the city. Ramalingam et. al [9] specifically match against city skylines using an omni-directional camera. Reitmayr & Drummond [10] take a different approach by using a textured 3D model to generate a virtual rendering of what the current scene is predicted to look like. Edge detection on this virtual image then determines which edges are tracked to aid in registration. Our approach differs from these works in that we look to computer vision to correct only for compass heading error and adaptively constrain the search space of corrections. These considerations are a major contribution of this work and allow for highly accurate registration that is easily achieved in real time.

REGISTRATION MODEL AND COMPUTATIONS

AR registration involves finding a transformation between real and virtual world coordinate systems in order to properly align information on the real world image. This transformation is standard; hence we only sketch the outline of the procedure to help us define terms we will use later.

We use unit quaternions to represent rotations and orientations in 3-dimensional space. The use of quaternions is advantageous because it allows a singularity-free representation (avoids the "gimbal lock" of Euler angles). Our representation of world locations is based on the n-vector as described in [11].

The iOS 4.0 CoreMotion library provides access to a relative orientation of the device computed from gyroscope and accelerometer signals, which is then aligned to the sensed compass heading. To do this, we rotate the sensed magnetic field vector by the device's current orientation to put it into the world coordinate system. We then project the vector onto the XY-plane and measure the angle off of the positive Y-axis, which we take to be north, and add in the magnetic declination at the user's location. Finally, we combine a rotation of this angle with the relative orientation to get our sensed absolute orientation which we denote as $\boldsymbol{q}$.

Figure 2. Original (red) and heading-corrected (blue) placement of wind turbines.

For a particular point $\boldsymbol{W}$ in a 3-space whose origin is at the user's location, we can compute the onscreen location $\boldsymbol{w}$ by rotating according to the user's orientation and projecting according to our camera's intrinsic parameters (which can be found using a standard camera resectioning tool). The standard camera parameters f_x, f_y, c_x, c_y characterize the focal length and principal point of our camera projection. We found lens distortion in the iPad 2 camera feed to be negligible. We call this transformation $\mathrm{t}(\boldsymbol{W})$ and its computation is as follows:

$$t(\boldsymbol{W}) = \boldsymbol{w} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x * \dfrac{W'_x}{-W'_z} + c_x \\ f_y * \dfrac{W'_y}{-W'_z} + c_y \end{bmatrix}$$

$$where\ \boldsymbol{W}' = rotate(\boldsymbol{W}, \boldsymbol{q})$$

If W'_z is negative, the point is behind the user and cannot be seen.

In modeling a wind turbine, we only consider the tower itself and not the spinning blades. To display the tower onscreen, we compute the locations of its four "corners" $\boldsymbol{w_{BL}}$, $\boldsymbol{w_{BR}}$, $\boldsymbol{w_{TL}}$ and $\boldsymbol{w_{TR}}$. This is straightforward given a model of the wind farm and turbines. Using sensor data from the device's GPS, compass, accelerometer, and gyroscope we construct and initial guess transformation t_0. Unfortunately, as discussed above and as seen in Figure 2, this placement is often incorrect due to error in the sensor readings.

VISION-BASED HEADING CORRECTION

Now we are ready to describe our computer vision algorithm to correct for errors in registration from sensor data. The general approach is that we use t_0 as an initial guess in a search for the correspondence which best matches a virtual model of wind turbines against edges in the camera image. Under the assumption compass heading is the primary source of error, the search space can be considered a 1-dimensional interval of compass heading adjustments. Our results presented in a later section show that such an algorithm eliminates the majority of registration error and thus add credence to this assumption. First, we describe this process in detail.

In brief, our search problem is to find θ^*:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} f(\theta), \Theta = [-r, r]$$

The search interval has radius r determined by how accurate our initial guess is assumed to be. The first time the algorithm is run, we conservatively use a very large interval with $r = 45°$. After the algorithm has been run once, a successful registration can be used to seed subsequent iterations of the algorithm. Thus, t_0 will contain minimal error and we set $r = 0.5°$. This step is crucial in order to achieve good efficiency (discussed further below) as it greatly reduces the search space of heading corrections.

We discretize this interval and perform an exhaustive search. As a technical note, the increment used in discretization must be made small enough such that the correct orientation is not “skipped over”. In our system and for the results presented here, we used an increment of 0.001 radians (≈ 0.06°).

The design of the objective function $f(\theta)$ is to assign higher scores to headings which match more turbines against edges in the camera image. For each θ, we adjust t_0 in the natural way to construct the transformation t_θ and use it to compute the onscreen locations of the turbines. Then we compute a score for each turbine as follows. We uniformly sample 10 pixels from each side of the turbine and check the gradient in the perpendicular direction at each. If this gradient is greater than threshold α we consider this pixel matched against an edge. The score for the turbine is then the percent of pixels matched.

Towers with a score greater than threshold β are considered positively matched and we define $f(\theta)$ as the sum of the scores of these towers. In the experimental results presented below we used $\beta = 0.4$. Additionally, because very near or very distant towers are matched much less reliably, we limit consideration to those towers between 50m and 1500m of the user.

Figure 3 gives insight into the computation of $f(\theta)$ algorithm. Red lines show the placement of turbines using t_0. Blue lines show the placement using t_{θ^*} which incorporates the heading adjustment chosen by our algorithm. Other colors, one for each tower, show which pixels we actually sample when matching edges. These are overlaid on a gray edge image for context, but the only pixels the algorithm actually checks are those denoted by the colored lines.

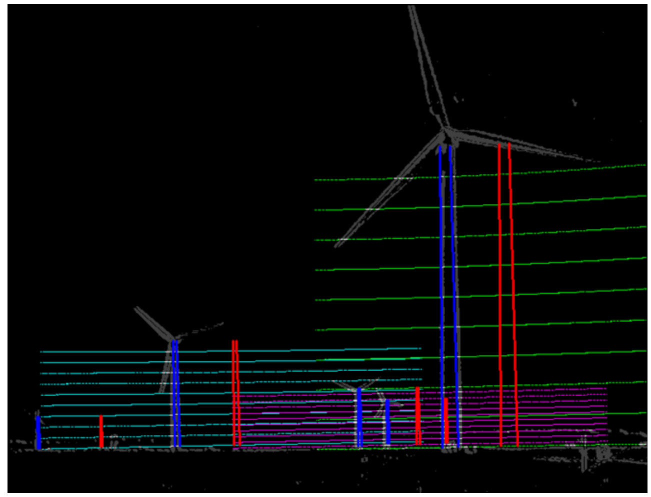

Figure 3. Pixels checked when using a large compass heading correction interval radius (15 degrees)

ACCURACY, RELIABILITY, AND EFFICIENCY OF HEADING CORRECTION

In order to test our heading correction algorithm, we compiled a set of 70 test images taken on two different days at five different locations on a wind farm along with the current sensor readings of the device at the time each was taken. The figures shown here have been cropped from their original size (768x1024 pixels) to the relevant regions for the sake of neatness.

Qualitative evaluation of the heading corrections produced by our algorithm for these images shows wind turbines are correctly placed in 59 out of 70 test images. Figure 2 and Figure 4 show several images with original and heading-corrected turbine placements. Our algorithm did not successfully compute the appropriate heading correction in the remaining images. In some of these images, the central problem is that our algorithm is unable to find edges within the image because of poor illumination as seen in Figure 5.

Although these pictures are illustrative, we have also devised a quantitative scheme to measure placement accuracy in images where good registration is achieved. For each of the 14 such test images, we manually annotated ground truth locations of the corners of each tower. We omit distant or obscured wind turbines whose corners cannot be annotated accurately. We define an error measure $J(t)$ as the mean squared pixel distance between all placed tower corners ($\boldsymbol{w_{BR}}$, ...) and their corresponding ground truth ($\boldsymbol{\widehat{w}_{BR}}$, ...).

A hypothetical perfect registration would result in $J(\hat{t}) = 0$. For the images in our dataset, we find that registration using only passive sensors has mean $J(t_0)$ = 196.5 pixels (stdv. 21.3). Using our line matching algorithm, we achieve mean $J(t_0)$ = 8.4 pixels (stdv. 0.6).

For the scene shown in Figure 1, processing a single frame of the camera feed takes on average 108ms (stdv. 4.9ms) on the iPad2 when the interval of possible compass headings

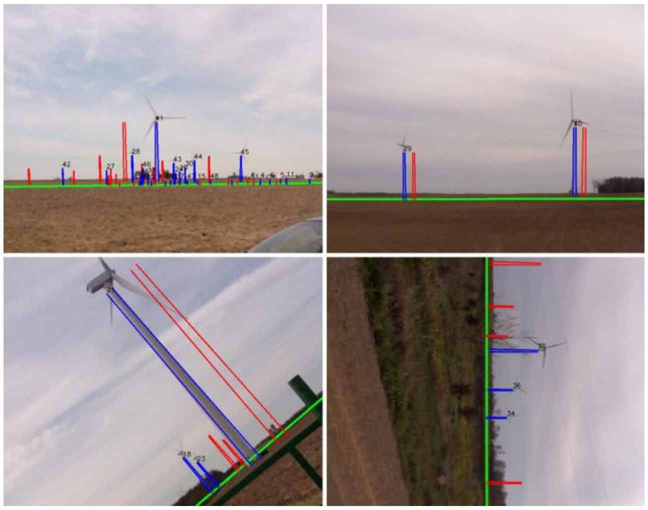

Figure 4. Example images with original turbine placements (red) and heading-corrected placements (blue).

has radius 45°. In subsequent iterations of the algorithm when the compass interval is adaptively constrained to be quite small, processing a single frame on the iPad2 takes 10-20ms. Although the time for processing a single image depends somewhat on the scene being viewed, the limit of matching only towers within 1500m keeps scenes with many distant towers from taking significantly longer to process. This processing time is certainly amenable to a real-time application.

CONCLUSIONS

In this paper we presented an efficient solution to the augmented reality registration problem for commodity mobile device augmented reality applications where geometric features and locations are known. In order to correct for error in the sensed compass heading, we employ a matching algorithm based on the known locations and typical geometry of wind turbines. Once an initial registration has been found, we can speed up subsequent iterations of the algorithm by adaptively constraining the search space of possible heading corrections. Experimental results demonstrate the algorithm offers good pixel accuracy at interactive rates on commodity mobile devices.

We expect this solution to be viable for a range of geo-location applications where sufficient geometric features are known. For example if architectural information such as edges of buildings are known, a similar vertical detection strategy is likely possible. Thorough investigation of the extension of this work to other application settings is intended for future work.

ACKNOWLEDGMENTS

We thank Sam Guertin, Li Chen, Will Kuchera, Mark Wiacek, Donald Perigo, and James Ranes of General Electric for collaborating with us on this project and gratefully acknowledge GE’s financial support of the project.

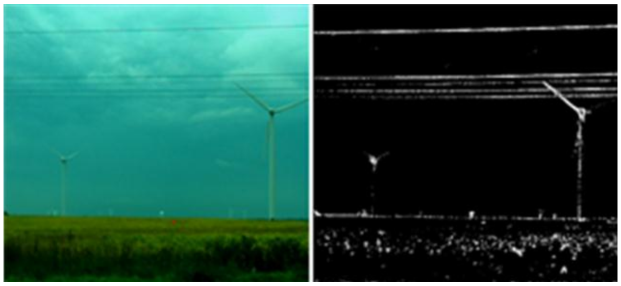

Figure 5. Poor illumination can inhibit edge detection. Left: original image. Right: edges detected.

REFERENCES

[1] R. T. Azuma, "Tracking requirements for augmented reality," *Communications of the ACM*, vol. 36, no. 7, pp. 50-51, 1993.

[2] T. Höllerer and S. Feiner, "Mobile augmented reality," in *Telegeoinformatics: Location-Based Computing and Services*, H. Karimi and A. Hammad, Eds., CRC Press, 2004.

[3] M. G. Wing, A. Eklund and L. D. Kellogg, "Consumer-grade global positioning system (GPS) accuracy and reliability," *Journal of Forestry,* vol. 103, no. 4, pp. 169-173, 2005.

[4] R. T. Azuma, "The challenge of making augmented reality work outdoors," *Mixed reality: Merging real and virtual worlds*, vol. 1, pp. 379-390, 1999.

[5] R. Azuma, B. Hoff, H. Neely and R. Sarfaty, "A Motion-Stabilized Outdoor Augmented Reality System," in *IEEE Virtual Reality*, Houston, TX, 1999.

[6] S. You and U. Neumann, "Fusion of vision and gyro tracking for robust augmented reality registration," in *IEEE Virtual Reality*, 2001.

[7] B. Hoff and R. Azuma, "Autocalibration of an Electronic Compass in an Outdoor Augmented Reality System," in *IEE and ACM International Symposium on Augmented Reality*, Munich, Germany, 2000.

[8] V. Coors, T. Huch and U. Kretschmer, "Matching buildings: Pose estimation in an Urban environment," in *ISAR 2000*, Munich, Germany, 2000.

[9] S. Ramalingam, S. Bouaziz, P. Sturm and M. Brand, "Skyline2gps: Localization in urban canyons using omni-skylines," in *International Conference on Intelligent Robots and Systems*, Taipei, 2010.

[10] G. Reitmayr and T. W. Drummond, "Going out: Robust Model-based Tracking for Outdoor Augmented Reality," in *IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2006.

[11] K. Gade, "A Non-singular Horizontal Position Representation," *The Journal of Navigation,* no. 63, pp. 395-417, 2010.