Developing a Vision-based Evaluation Testbed for Mobile Robot Navigation

Sangho Park
Department of Computer, Electronics and Graphics Technology
Central Connecticut State University

In STEM (Science, Technology, Engineering and Mathematics) education, mobile robot development and competition has become very popular. Mobile robot is an effective platform for stimulating student motivation at K-12 institutions as well as for rigorous engineering practices in colleges and universities. Objective estimation of mobile robot movements is critical in robot design and development at all levels of institution. However, the current state of the art in performance evaluation of mobile robot is typically based on human evaluator's manual intervention using chronometer to measure the time of completion in a given task or accuracy counting of pass / fail on the task. This paper reports an effort to develop a vision-based evaluation testbed using multi-camera vision system that can automatically record the movement of multiple robots and objectively estimate their navigation performance in terms of physics-based profiles: position, velocity, and acceleration of robot over time with respect to a given world-coordinate system. The current testbed uses two synchronized cameras, and they can be mounted in versatile manner: for top-down view or arbitrary oblique views depending on user need. Individual trajectory points of the robots are time-stamped, and the calculation of the position, velocity, and acceleration provides the full description of the robot movements. The testbed has an intuitive graphical user interface (GUI) with which students can run the system easily and visualize the trajectories intuitively. The testbed can be used to estimate mobile robot navigation in quantitative manner and to provide students and developers with insights about objective performance evaluation criteria based on the physics-based profile.

Corresponding Author: Sangho Park, spark@ccsu.edu

Introduction/Background

Mobile robot development and competition has become very popular in STEM[1] (Science, Technology, Engineering and Mathematics) education. Mobile robot is an effective platform for stimulating student motivation at K-12 institutions as well as a good tool for rigorous engineering practices in colleges, universities, and graduate schools. In robot design and development at all levels of institution, it is critically import to objectively measure and estimate the mobile robot navigation performance. However, the current state of the art in the performance evaluation of mobile robot navigation is typically based on the human evaluator's manual intervention using a chronometer to measure the time of completion in a given task or accuracy counting of pass / fail on the task. The manual intervention is error-prone and can be biased as well. We need a tool that reliably and objectively conducts the performance evaluation in an automated manner. It is also desirable to develop an unobtrusive method that does not require the attachment of any sensors, transponders, or beacons to the mobile robot since such attachment will change the weight of the robot not alone the complicacy of the installation and management of such attachment. A computer vision-based object detection and tracking system is a promising solution in this regard.

This paper reports an effort to develop a vision-based evaluation testbed for mobile robot navigation by using multiple cameras that automatically record the movement of the robots and objectively estimate their navigation performance. The proposed system evaluates the navigation performance in terms of the physics-based profiles: position, velocity, and acceleration of robot over time with respect to a given world-coordinate system.

Data/Formulation/Methodology

Our methodology starts by modeling the process of image formation when a scene is viewed through a camera. We adopt a *pinhole camera model* shown in Figure 1. Pinhole camera model[1] assumes that exactly one ray from each point in the scene passes through the pinhole lens and hits the image plane opposite to the scene, forming the inverted image. For convenience we use the virtual image in front of the pinhole, forming the new image plane in Figure 2. The mathematical description of the imaging process in Figure 2 denotes the world scene coordinates with uppercase roman letters $\{X, Y, Z\}$ and image coordinates with lowercase roman letters $\{x, y, z\}$. Note that the vectors pointing from camera center $\boldsymbol{C}$ to the world coordinate point and the corresponding image coordinate point are denoted by boldface symbols, such as $\mathbf{X}$ and $\mathbf{x}$, respectively. The

camera's imaging process is to map the point (*X,Y,Z*) in the 3D space to the point (*x,y*) on the 2D image plane, and is modeled as follows (,where the superscript *t* means column vector notation.)

$$(X,Y,Z)^t \xrightarrow{Projection} (x,y)^t$$

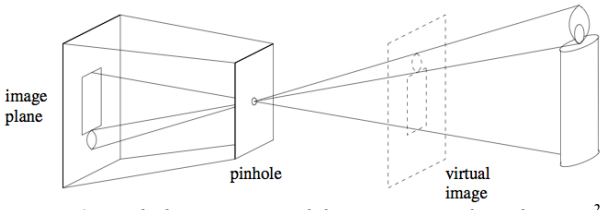

Figure 1: Pinhole camera model. From Forsyth and Ponce[2], 2003.

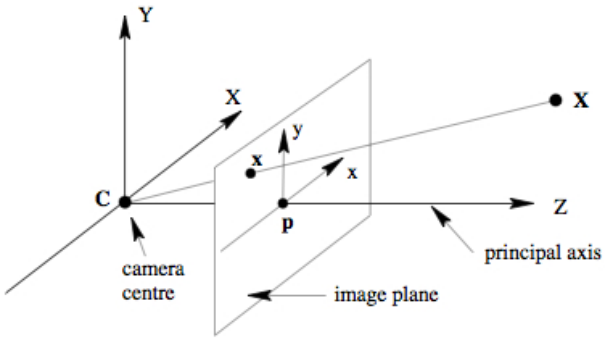

Figure 2: Image projection in a projective camera.

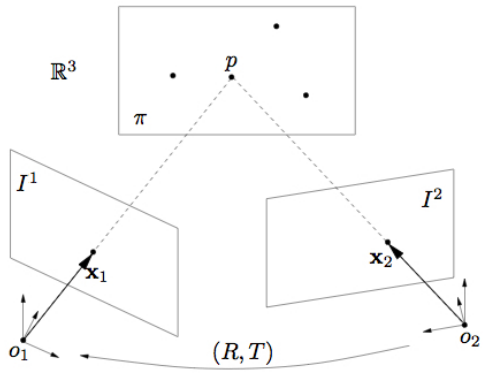

Figure 3: Two views of points on a plane π ⊂ R3. For a point p ∈ π, its two (homogeneous) images are the two vectors x1 and x2 with respect to the two vantage points O1 and O2, respectively. From Ma et al.[3], 2001.

As shown in Figure3, if we use two cameras, each of which forms image plane I^1 and I^2 with the corresponding camera origins $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$, respectively. We define the relative configuration of the two cameras in terms of the relative rotation *R* and translation *T*. Note that the same point *p* in the world scene π appears very different in the two image planes as vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, respectively, due to perspective distortion effect. Overall, the camera

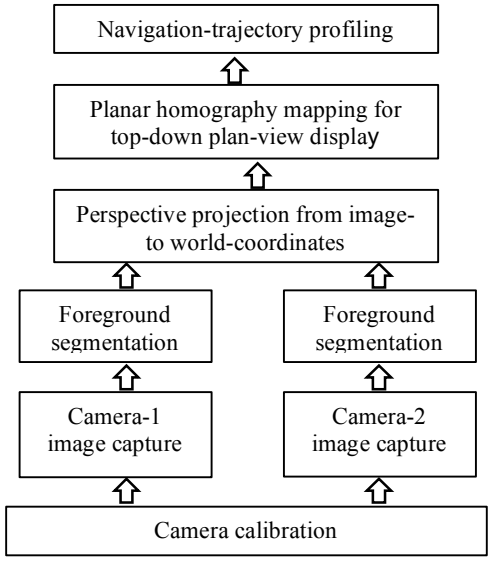

Figure 4: Overall system diagram depicting the processes from image capture to orthographic trajectory profiling.

imaging process is mathematically modeled as the mapping from the 3D-world scene coordinate (*X,Y,Z*) to the 2D image plane coordinate (*x,y*) of a viewed object, which results in inevitable loss of information during the downgrading transformation from the higher to lower dimension.

Our real goal with the computer vision system is to recover the inverse mapping from the 2D image coordinate $(x,y) \in \mathbb{R}^2$ of the viewed object on the image planes I^1 and I^2 to the world scene coordinate $(X,Y,Z) \in \mathbb{R}^3$ in the world scene π in order to achieve the accurate estimation of the world scene given only the 2D image data. We need at least two camera views to resolve the ambiguity caused by the information loss during the downgrade transformation. To resolve the ambiguity, we first conduct the camera calibration that establishes the camera configuration (*R,T*) in Figure 3 as:

$$\boldsymbol{x}_2 = R\boldsymbol{x}_1 + T$$

A perspective projection[4] is made to do the inverse mapping from image- to world-coordinate system. The two versions of the inverse mapping from each camera are joined to a common ground by the planar homography[3], which generates a virtual top-down plan-view display of the world scene π. Using the homography matrix *H*, we can write the transformation of points in 3D from camera 1 o camera 2 as:

$$X_1 = HX_2, \quad X_1, X_2 \in \mathbb{R}^3$$

By using the 4-point algorithm in Ma et al.[3], the homography matrix *H* can correct the projective distortion of image planes I^1 and I^2 in Figure 3 and map on to the virtual top-down view displayed on the monitor that represents the world scene plane π. This virtual top-

down view of the world scene is camera-independent and orthographic, and provides the objective measure of the actual scene dimensions without view-depending distortion. The orthographic virtual top-down view display enables the generation of the navigation trajectory profile of moving objects in terms of position, velocity, and acceleration on the world scene coordinate system.

The overall system diagram is depicted in Figure 4. The system starts from the camera calibration after deploying the cameras to specific locations. The calibrated and synchronized cameras (camer-1 and camera-2) keeps capturing the synchronized image frames. The computer vision algorithm conducts foreground segmentation to filter out background scene and segregate only the moving foreground objects. Perspective projection and planar homography mapping are performed as explained earlier. The navigation-trajectory profiling will be explained in the next section.

Analysis

We developed a testbed composed of two cameras, a frame-grabber board, a desktop computer, and a flat table as shown in Figure 5. The current testbed uses two synchronized cameras, and they can be mounted in versatile manner: for top-down view or arbitrary oblique views depending on user need. The left-view camera is installed on top of a tripod while the right-view camera was attached to a clamp fastened on a bookshelf cabinet. Part of the flat table is seen also.

The left- and right-view images captured by the cameras are shown in Figure 6. The appearance of the same flat table looks very different on the left- and right-view image frames due to the different perspective effect of the cameras. We define the world seen coordinate system on the flat table in terms of the origin *O* and *X*- and *Y*-axis extended from the origin. Note that it is conventional to define the image domain's Y-axis to extend toward row direction of the image (in Figure 6), while the mathematical definition of the Y-axis is opposite to it (in Figure 2.)

Camera calibration is achieved by using a standard checker board pattern shown in Figure 7. Multiple snapshots of the calibration board at different positions are used in the procedure. Figure 8 shows the left- and right-view snapshot of the experimental run of a micro-robot running on the flat table. Individual trajectory points of the robots are time-stamped, and the calculation of the position, velocity, and acceleration provides the full description of the robot movements. We used two different micro-robots of different running speed in the experiment.

The testbed has an intuitive user interface as in Figure 9 with which students can run the system easily and visualize the trajectories intuitively. The control software

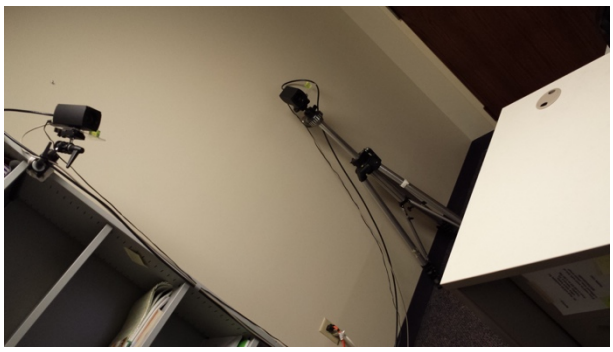

Figure 5: Testbed configuration using a flat table and two synchronized cameras. Left-view camera on the tripod, right-view camera on the clamp.

Figure 6: Left- and right-view images of the testbed. World coordinate system is defined on the testbed (overlaid for visualization on the right-view image.)

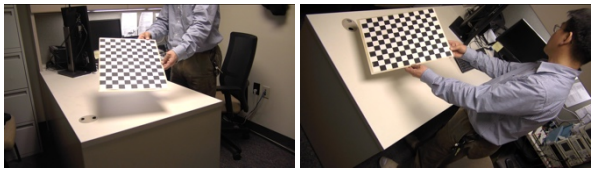

Figure 7: Snapshot of camera calibration using a standard calibration pattern.

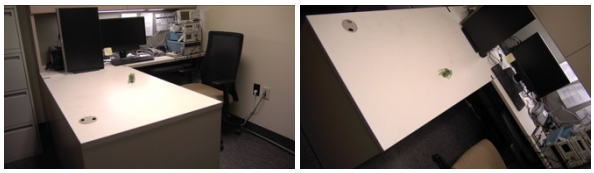

Figure 8: Snapshot of experiment using a small mobile robot on the testbed.

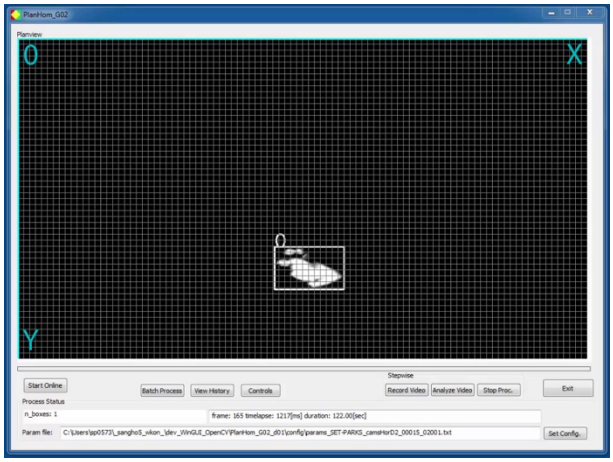

Figure 9: Graphical-user-interface frontend that controls the system and visualizes results. The overlaid world coordinate system of O, X, Y complies to that of the testbed in Figure 6.

for the video capture and the display on GUI frontend in Figure 9 was written in C++ language. We tested the proposed system with a general-purpose desktop PC installed with Microsoft Windows 7 operating system and 3.4 GHz Intel Core i7 CPU. Two micro-robots were used to test their navigation on the flat table. The current system setup was in an author's office space for the micro-robots, but the system is scalable to deploy to larger environments such as a bigger lab floor for mini-robots or a spacious gym floor for bigger robots. Different deployment options do not require the software code modification; only the camera calibration needs update for the new camera positions.

We have tested the system by running it up to the maximum image capture size and the most time-consuming storage option (i.e., storing the individual frames on the hard drive as independent image files.) The average frame rate of the capture and storage was 5.28 [frames per second, fps] in this condition that includes simultaneously capturing the left- and right-image frames of 1920×1200 pixel dimensions in RGB color and saving them as individual TIFF image files on a single internal hard drive. If the image frame size is reduced and a multi-disk RAID hard drive system is used, the frame rate will increase significantly.

The tracking algorithm of the control software successfully tracks the moving robots and assigns the bounding box with track ID to the segmented foreground region as shown in Figure 9. The center position (x_i, y_i) at the *i*-th frame of the *k*-th bounding box with the track ID, *k*, is regarded as the position of the *k-th* robot at the given time of the *i-th* frame capture. The instantaneous velocity is defined as a vector $\vec{v}_i = (v_x, v_y)_i$ in terms of the differential location in *x*- and *y*-axis between the current *i-th* frame and the previous (*i-1*)-*th* frame. v_x and v_y are defined as:

$$v_x = \frac{x_i - x_{i-1}}{\Delta t}$$
$$v_y = \frac{y_i - y_{i-1}}{\Delta t}$$

The instantaneous acceleration is defined by the vector difference between the current *i-th* frame and the previous (*i-1*)-*th* frame as:

$$\vec{a}_i = \vec{v}_i - \vec{v}_{i-1}$$

Our system provides reasonably accurate trajectory profiles of the robot navigation in terms of the position, velocity, and acceleration at every frame. Some inaccuracy is induced at some tile; most of the disturbance in the accuracy is from imaging noise during foreground segmentation. It is an open research issue in computer vision to achieve a robust method for perfect foreground segmentation.

Conclusions

We have presented a vision-based evaluation testbed for mobile robot navigation. The testbed can be used to estimate mobile robot navigation in quantitative manner and to provide students and developers with insights about objective performance evaluation criteria based on the physics-based profile. The proposed system is a general-purpose system that can be deployed to estimate the navigation of various moving objects including, but not limited to, mobile robots. The developed system is versatile in that the cameras can be installed in various configurations such as top-down viewing direction or arbitrary oblique viewing directions as long as the two views are not in parallel. The system is non-obtrusive since it is purely vision-based and does not require the attachment of any sensors, transponders, or beacons to the tested object. Our future research plan includes testing the system in various environments such as moving-object tracking on a wide-area.

References

1. National Science Board.(2010). "Preparing the Next Generation of STEM Innovators: Identifying and Developing our Nation's Human Capital," Retrieved from: http://www.nsf.gov/nsb/publications/2010/nsb1033.pdf
2. David Forsyth and Jean Ponce, "*Computer Vision: A Modern Approach*," Prentice Hall, 2003.
3. Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar Sastry, "*An Invitation to 3-D Vision: From Images to Geometric Models*," Springer, 2001.
4. Richard Hartley and Andrew Zisserman, "*Multiple View Geometry in Computer Vision*," 2nd ed. Cambridge University Press, 2003.

Acknowledgement

This research was supported and funded by AAUP MR&RC Research Grant from Central Connecticut State University.