2-2013

Visual homing with a pan-tilt based stereo camera

Paramesh Nirmal  
*Fordham University*

Damian M. Lyons  
*Fordham University*

Follow this and additional works at: https://fordham.bepress.com/frcv_facultypubs

Part of the *Robotics Commons*

Recommended Citation

https://fordham.bepress.com/frcv_facultypubs/15

This Article is brought to you for free and open access by the Robotics and Computer Vision Laboratory at DigitalResearch@Fordham. It has been accepted for inclusion in Faculty Publications by an authorized administrator of DigitalResearch@Fordham. For more information, please contact considine@fordham.edu.
Visual homing with a pan-tilt based stereo camera

Paramesh Nirmal and Damian M. Lyons
Department of Computer Science, Fordham University, Bronx, NY 10458

ABSTRACT

Visual homing is a navigation method based on comparing a stored image of the goal location and the current image (current view) to determine how to navigate to the goal location. It is theorized that insects, such as ants and bees, employ visual homing methods to return to their nest [1]. Visual homing has been applied to autonomous robot platforms using two main approaches: holistic and feature-based. Both methods aim at determining distance and direction to the goal location. Navigational algorithms using Scale Invariant Feature Transforms (SIFT) have gained great popularity in the recent years due to the robustness of the feature operator. Churchill and Vardy [2] have developed a visual homing method using scale change information (Homing in Scale Space, HiSS) from SIFT. HiSS uses SIFT feature scale change information to determine distance between the robot and the goal location. Since the scale component is discrete with a small range of values [3], the result is a rough measurement with limited accuracy.

We have developed a method that uses stereo data, resulting in better homing performance. Our approach utilizes a pan-tilt based stereo camera, which is used to build composite wide-field images. We use the wide-field images combined with stereo-data obtained from the stereo camera to extend the keypoint vector described in [3] to include a new parameter, depth ($z$). Using this info, our algorithm determines the distance and orientation from the robot to the goal location.

We compare our method with HiSS in a set of indoor trials using a Pioneer 3-AT robot equipped with a BumbleBee2 stereo camera. We evaluate the performance of both methods using a set of performance measures described in this paper.

Keywords: visual homing, visual navigation, stereovision, mobile robots

1. INTRODUCTION

Visual homing is a navigation method that an agent can use to return to a previously visited location (known as the home location or the goal location) by comparing the image currently in the agent’s view with a stored image of the goal location. It is theorized that biological agents such as ants, bees, and even humans, employ visual homing methods to return to their nest [1]. With visual homing, an agent can return to its goal location from any nearby point that has a sufficient degree of visual similarity [2]. It can be argued that most biological agents use more than just visual information to navigate from one location to another. For example: pigeons may use their built-in compass for homing purposes. However, the use of such sensors is for long-range navigation. For local navigation, an agent cannot solely rely on idiothetic information. They have to use external information, often provided by vision [5]. Visual homing is primarily used for local navigation [1]. Some animals, including human beings have extra perception techniques, such as a perception of depth, with which they can perceive some objects as being further away than others. It is believed that humans use several (biological) methods to estimate distance, one of them being binocular vision*. Stereovision has been a popular imaging tool in recent years, with applications ranging from robotics to personal imaging. For example, smartphones such as the HTC Evo 3D uses a stereo camera for 3D imaging. Stereo cameras capture two overlapping images of a scene, and in hardware or software measure the visual disparity of features in the scene, from which a depth map can be constructed [4].

In this paper, we present a novel visual homing algorithm inspired by binocular vision. The algorithm uses a pan-tilt based stereo camera. Also presented in this paper is a performance evaluation of our method compared with Homing in Scale Space (HiSS) [2].

2. LITERATURE REVIEW

Visual homing has been a prominent area of research in the field of robotics and computer vision in the past few decades. One of the main motivations behind visual homing is to mimic the homing behavior of biological agents. It is vital for many animal species to find their way back to an important location such as a shelter (their home), or a food source [5]. Möller et al. [1] have studied visual homing strategies of insects such as desert ants, and implemented those strategies on autonomous mobile robots, achieving comparable performance. Franz et al. [6] have studied visual navigation approaches for autonomous agents, with the aim of achieving high navigation flexibility similar to biological agents such as bees, ants, and migrating birds.

Existing visual homing methods can be divided into two main categories; holistic, and correspondence based.

2.1 Holistic approach

Holistic methods treat the image as a whole, performing the comparisons on the entire image using one of the three following techniques: warping methods, parameter methods, and DID (Descent in Image Distance) methods.

The warping method, originally proposed by Franz et al. [5] in 1998, warps the current image according to certain movement vectors. The search space is the space of movement vectors tested to warp the current image until a movement vector is found. The movement vector leads to a warped image that is similar to the goal image. This approach operates under the assumption that a distance distribution of landmarks in the scene is known a priori. Such an assumption is rarely fulfilled in practice [7]; however, in environments where the landmarks are relatively distant from the goal, the assumption is satisfied because it can be assumed that the landmarks are equidistant from the goal. Franz’s warping method was proposed for 1-D images. However, Möller [8] extended Franz’s approach for processing 2-D images.

Parameter methods operate on the assumption that the agent does not have a stored image of the goal location. Instead, the agent has stored an epitomized description of the landmarks in the goal image [9]. It has been hypothesized since the early 1980s by Wehner [10] that biological agents such as desert ants and honeybees store images in the form of condensed parameter descriptors. Parameter methods use very little memory and processing time because they require no image processing. Due to their simplicity, they have an additional value as a model of insect visual homing [9].

Descent in Image Distance methods rely on the fact that the distance measure between two images taken at different locations increases smoothly with increasing spatial distance, observed by Zeil et al. [11]. Möller and Vardy [12] leverage this fact for local visual homing using matched-filter descent in image distance. Estimating the spatial gradient using two matched-filters on the image planes produces the home vector, which is used to lead the agent back to the goal location. This method operates under the assumption that there is no change in orientation from the agent’s goal location and current location.

2.2 Feature-based approach

The feature-based approach, also known as the correspondence approach, establishes correspondences between regions, or features, in the current image and the goal image. This approach is based on feature detection and algorithms to match the features from the current image to the goal image. Each pair of correspondences describes a shift vector [2], representing the shift of features in image space (from one image to the other). This vector can be transformed into a home vector, which leads the agent back to its home location.

Much work has been done in the past to establish correspondences between two images. Vardy and Möller [13] apply optical flow techniques for visual homing. Block matching, and differential techniques are tested to be robust, accurate homing methods, despite their simplicity [13]. Cartwright and Collet [14] use high contrast features in landmarks to determine correspondences between the two images. Other feature-based methods utilize Harris corners [15].

In the past few years, the use of SIFT (Scale Invariant Feature Transform) to determine correspondence between the current and goal image has gained great popularity, due to the robustness of their descriptor vectors with respect to changes in scale, rotation, and lighting [3]. Pons et al. [18] use SIFT features to detect visual landmarks in the scene, followed by a matching and voting scheme of the detected features. Similar to [13], the homing algorithm presented in [18] is driven under the assumption that the current and goal images both have the same orientation.

We are interested in Churchill and Vardy’s Homing in Scale Space (HiSS) method [2] because of its simplicity, and robustness. Churchill and Vardy use SIFT to detect keypoints (key features) in each image, followed by the matching of the two sets of features to obtain correspondence pairs. They equip the autonomous robot with an omnidirectional
camera, allowing the robot to be at any orientation for homing to take place. HiSS uses the angular position of the features in the image to determine the change in heading. To determine distance to the goal, their method uses the scale component of the SIFT keypoint vector. HiSS determines whether the robot is closer to the goal location or further from the goal location by checking whether the features in the image have expanded or contracted. The expansion and contraction of each feature is deduced by examining the features change in scale. The scale component of the keypoint vector is discrete [3] with a small range of values, making the result a rough measurement, with limited accuracy. Because HiSS uses scale to determine distance to the goal, the robot never has a concrete estimate of how far the it has to travel to reach the goal location. To overcome the lack of knowledge about distance to the goal, the robot takes small steps towards or away from features in the scene until the current image and the goal image satisfy a certain degree of similarity [7]. This guarantees that the algorithm will converge to the goal location. However, convergence can be slow and inaccurate.

2.3 Sensor fusion for visual homing

Recently, visual homing methods have been extended with the use of sensor fusion. Choi et al [16] implement homing using a laser-camera fusion system. They demonstrate that using extra sensory equipment enables autonomous homing of a robot in large-scale indoor environments, as opposed to small-scale, local environments.

Implementing visual homing algorithms in real applications can raise a number of significant challenges such as dynamic obstacles, e.g. moving people. Liu et al [17] use visual information coupled with the robot’s odometry information to develop an indoor topological navigation framework, resulting in robust performance in real-time environments with scene variation. Our method does not use the robot’s odometry information for homing.

The method we present is based on the correspondence approach presented by Churchill and Vardy in [2]. Similar to their approach, we use SIFT to detect correspondences between the current image and the goal image. However, we leverage the use of stereovision to calculate the depth of each feature, and therefore obtaining a better estimate of orientation and distance to the goal. The stereo camera is mounted on a pan-tilt base, allowing the robot to capture a wide-field image, and thus guaranteeing that the goal location is in the robot’s field of view. This eliminates the requirement that the robot be facing towards the goal location. The robot used for our experimental trials is a Pioneer 3-AT robot, equipped with a BumbleBee2 stereo camera mounted on a directed perception pan-tilt unit.

3. HOMING WITH STEREOVISION

We will refer to our method, Homing with Stereovision as HSV. Let $C$ be the robot’s current image and $G$ be the image of the goal location (also known as the snapshot). Let $c$ and $g$ be the robot’s current position, and position at the goal location, respectively. Let $h$ be the vector describing the amount the robot has to rotate, and distance the robot has to travel to get from $c$ to $g$. $h$ is also known as the home vector, since it leads the robot back to its home location.

If $c$ and $g$ lie within the same plane, then the robot can home in one step [7] that is by rotating $\Theta$ degrees, which is the change in orientation between the robot’s current and goal position, and by travelling distance $D$, which is the straight-line distance between $c$ and $g$ (assuming that $\Theta$ and $D$ are known).

Our algorithm estimates $\Theta$ and $D$, therefore it is possible for the robot to reach its goal in one step. However, to achieve an accurate final position, we only move the robot by a fraction of $\Theta$ and $D$ in each step. Moreover, it is not assumed that the $c$ and $g$ lie within the same plane; therefore, it will not always be the case that the robot can home in one step. The home vector, $h$, is described as:

$$h = [\Theta, D]$$

We extract SIFT features from $G$ and $C$ using David Lowe’s SIFT tools*, followed by determining the correspondence pairs†. The $i^{th}$ extracted feature from image $G$ can be described as the following vector, known as the keypoint vector [7]:

$$G^i = \{G_{i,x}, G_{i,y}, G_{i,\theta}, G_{i,\sigma}, G_{i,d}\}$$

---

* Obtained from http://www.cs.ubc.ca/~lowe/keypoints/
† A correspondence pair is a pair of matched features from $C$ and $G$. 
Where \((G_I^i, G_I^v)\) is the feature location in the image \(G\). \(G_I^d\) and \(G_I^\sigma\) are the feature’s orientation and scale, respectively. \(G_I^d\) is the feature descriptor, that is a 128-dimensional vector. The feature descriptor \(G_I^d\) is used by the SIFT tools to detect a match between the feature in \(G\) and a feature in \(C\).

In order to calculate \(h\), we use information obtained from the stereo camera about the same features extracted using the SIFT tools. We redefine the keypoint vector as follows:

\[
f = \{f^x, f^y, f^z, f^\alpha, f^d\}
\]  
(3)

where \((f^x, f^y)\) is the feature location in the real world, in robot coordinates. \(f^z\) is the feature’s depth, i.e. the distance from the center of the robot to the feature, in the \(z\)-plane. And finally, \(f^\alpha\) is the angular position (azimuth) of the feature \(f\) with respect to the center of the robot. \(f^d\) remains as the feature descriptor vector. Figure 1 describes the coordinate frame used in our algorithm. It is possible for the position or depth of the feature, \((f^x, f^y, f^z)\), obtained from the stereo camera to be an inaccurate or false reading. To overcome this inconsistency, we collect a set of points around the feature, and perform the RANSAC algorithm to obtain consistent readings for \((f^x, f^y, f^z)\).

Our algorithm only makes use of information delivered by the stereo camera. We do not make use of the robot’s odometer, or any other sensor modules. We demonstrate how it is possible to achieve accurate homing results without using any equipment but a stereo camera.

\[
G^i_\alpha = \theta f G^i_x, G^i_z
\]  
(4)

And similarly,

\[
C^j_\alpha = \theta f C^j_x, C^j_z
\]  
(5)

The change in orientation that the robot has to exhibit \(\Theta\), is calculated as the average of the difference in angular position for all matched features, \(\theta^m\) where \(m\) is the matched feature \((i, j)\). If there are \(n\) matched features, and \(M\) is the set of matched features, then \(\Theta\) is calculated as follows:

\[
\Theta = \frac{1}{n} \sum_{m \in M} (\theta^m)
\]  
(6)

where \(\theta^m = G^i_\alpha - C^j_\alpha, m = (i, j)\)

Similarly, the distance the robot has to move, \(D\) is calculated as the average of the difference in depth for all matched features, \(d^m\) where \(m\) is the matched feature \((i, j)\):
\[ D = \frac{1}{n} \sum_{m \in M} (d^m) \] (7)

where \( d^m = G^z - C^z \), \( m = (i, j) \)

The images taken from the stereo camera are composite wide-field images, i.e. an image composed of the several images taken by panning the stereo-camera over a wide range, and simply concatenating the collected images. Figure 2 shows an example of a wide-field goal image matched with a wide-field current image. The wide-field images taken by the robot are not perfect panoramas; i.e. the scenes in the concatenated images exhibit some overlap. There are two reasons as to why we do not implement panoramic stitching into our algorithm:

1. Panoramic stitching can be computationally expensive, therefore increasing time taken to process the image
2. By allowing overlapping scenes, some features exist twice in the image thus, increasing the likelihood of the feature being matched with a feature in another image.

Although the wide-field images are not perfect panoramas, they can encompass up to 270° field of view. This eliminates the assumption that the robot must be facing the goal location. Since the image encompasses a 270° field of view, it is guaranteed that there will be overlapping scenes in the goal and current image, even if the robot is facing the opposite direction to the goal location (provided that there are no obstructions between the robot and the goal location).

The following illustrates why we move the robot by a fraction of \( h \):

Consider the detected features in Figure 3. The robot would exhibit rotation \( \theta_1 = \frac{\alpha_1 + \beta_1}{2} \)
and movement \( D_1 = \frac{1}{2}(\text{diff}(z_1) + \text{diff}(z_2)) \), where \( \alpha_1 = \alpha_1 - \beta_1 \), and \( \beta_2 = \alpha_2 - \beta_2 \). Let \( h_1 = [\Theta_1, D_1] \).

As you can see from Figure 3, \( h_1 \) would be a good home vector, as it has precisely lead the robot back to the goal location (assuming no error in motion).

Now consider the detected features in Figure 4. The robot would rotate by \( \theta_2 = \frac{\alpha_2 + \beta_2}{2} \)
and movement \( D_2 = \frac{1}{2}(\text{diff}(z_1) + \text{diff}(z_2)) \), where \( \alpha_2 = \alpha_2 - \beta_2 \), and \( \beta_2 = \alpha_2 - \beta_2 \). Let \( h_2 = [\Theta_2, D_2] \).

It is evident that in this case, rotating by \( \Theta_2 \), then translating the robot by \( D_2 \) will not accurately lead the robot back to its goal location \( (g \neq c) \). However, if the robot moves by a fraction of \( h_2 \) in every iteration, then the algorithm will converge to \( g \), resulting in accurate homing.

\[^{\text{diff}}(z) = G^z - C^z \], where \( G^z \) and \( C^z \) are the depths of the matched feature in the goal image, and the current image, respectively.
Once \( h \) has been calculated, we rotate the robot by \( g_a \cdot \Theta \) and translate the robot by \( g_d \cdot D \), where \( g_a \) is the angular gain, and \( g_d \) is the distance gain such that \( 0 < g_a, g_d < 1 \). This step is taken in order to guarantee convergence, and to preserve homing accuracy. We have empirically discovered that using values \( g_a = \frac{1}{2} \) or \( \frac{1}{3} \), and \( g_d = \frac{1}{2} \) suffices to achieve accurate homing performance.

Since the robot does not have any odometer information, it is unable to ascertain whether \( (g = c) \) holds. We implement the following termination condition:

\[
\text{If } \left( (\Theta < \epsilon) \text{ AND } (D < D_{tol}) \right) \text{ Return success;}
\]

where \( \epsilon \) is a small angular value and \( D_{tol} \) is a small distance tolerance. The algorithm runs in a loop until the termination condition is satisfied.
4. RESULTS

4.1 Experimental setup

In this section we compare the performance of our method with that of HiSS [2]. Both methods were tested on a Pioneer 3-AT robot using a BumbleBee2 stereo camera mounted on a pan-tilt base, with a set of goal locations and start locations at FRCV Lab (see Figure 5). Our experiments were conducted using two goal locations, and three start positions for each goal location, each at a different distance and angle from the goal location. All calculations were conducted on an Intel Core2 2GHz processor, using the onboard computer for the Pioneer robot.

Although it is possible to create a panoramic image by panning the camera and taking several images, it is not possible to precisely obtain the field of view engulfed by the panorama, due to image overlap. HiSS operates under the assumption that the field of view is known, in order to calculate the angular position of features in the image. For this reason, we compare both methods using single frame images, over a 70° field of view. Since HiSS does not consider distance to \( g \), we allow the robot to translate by +/- 200cm in each iteration.

![Panoramic image of FRCV Lab taken using a BumbleBee2 stereo camera mounted on a pan-tilt base.](image)

4.2 Performance measures

1. Position error

We compare the position error of our method with that of HiSS. The position error, \( \epsilon_p \) is the straight-line distance from the center of the robot (at its final location after homing) to the goal location. HiSS does not include a termination condition; thus, we use the closest position of the robot from the goal position after running HiSS for 12 iterations. The position error chosen, \( \epsilon_p = \min(\epsilon_p^{\text{det}}, \epsilon_p^{\text{undet}}) \) where \( \epsilon_p^{\text{det}} \) and \( \epsilon_p^{\text{undet}} \) are the position errors for detected and undetected returns, respectively. We do not consider position errors for cases where the robot fails to return to the goal.

2. Angular error

The angular error, \( \epsilon_\theta \) is defined as [2]:

\[
\epsilon_\theta = |\theta_{\text{ideal}} - \theta_{\text{homing}}|
\]

where \( \theta_{\text{homing}} \) is the angle computed by the homing algorithm in the first iteration. \( \theta_{\text{ideal}} \) is calculated as [2]:

\[
\theta_{\text{ideal}} = \text{atan2}(G'y - C'y, G'x - C'x) \quad *
\]

3. Return ratio [7]

The return ratio (RR) measures the number of times the robot was able to successfully navigate to the goal location. A successful return occurs when the robot returns to within 25cm of the goal location. Although our algorithm includes a termination step, we will count undetected returns as successful returns, because HiSS does not include a termination condition. An undetected return occurs when the robot comes within 25cm of the goal location, but does not stop.

4. Number of steps

* Positions are recorded in 2D world coordinates, where \([G', G'] = [0, 0] \).
We define $N$ as the number of steps the robot takes to return to the goal location. Our algorithm aims to return the robot back to the goal location in the fewest steps possible whilst maintaining the accuracy of the robot’s final position.

### 4.3 Results and discussion

The results are presented in Table 1. The first column lists the difference in start position and goal position of the robot. The remaining columns list the measurements obtained from our live trials, representing the performance of each method. We compute an average of the data obtained from all trails for comparison. The angular error ($\epsilon_\theta$) of HiSS is 145% larger than that of HSV. However, the position error ($\epsilon_p$) of the two methods has little difference. On a case-by-case basis, the results show that HiSS outperforms HSV in terms of position error for half of the chosen locations. We believe that $\epsilon_p$ can be controlled with limited accuracy, due to the fact that both algorithms are based on image comparison. Since the robot does not have any odometer information, solely the image features affect its position. Two images taken from different positions, that are 20cm apart are almost identical. Therefore, it is unrealistic to expect high accuracy for the robot position.

<table>
<thead>
<tr>
<th>Position (m)</th>
<th>$\epsilon_\theta$ (degrees)</th>
<th>$\epsilon_p$ (cm)</th>
<th>$\epsilon_p$ (degrees)</th>
<th>$\epsilon_p$ (cm)</th>
<th>$\epsilon_p$ (degrees)</th>
<th>$\epsilon_p$ (cm)</th>
<th>$\epsilon_p$ (degrees)</th>
<th>$\epsilon_p$ (cm)</th>
<th># Steps (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HSV</td>
<td>HiSS</td>
<td>HSV</td>
<td>HiSS</td>
<td>HSV</td>
<td>HiSS</td>
<td>HSV</td>
<td>HiSS</td>
<td>HSV</td>
</tr>
<tr>
<td>(2.0,1.2)</td>
<td>6.828</td>
<td>13.116</td>
<td>19.80</td>
<td>16.75</td>
<td>0.80</td>
<td>0.50</td>
<td>6.60</td>
<td>10.25</td>
<td></td>
</tr>
<tr>
<td>(1.0,2.0)</td>
<td>3.557</td>
<td>9.261</td>
<td>21.80</td>
<td>17.80</td>
<td>0.90</td>
<td>0.60</td>
<td>3.80</td>
<td>9.00</td>
<td></td>
</tr>
<tr>
<td>(1.5,0.5)</td>
<td>4.194</td>
<td>9.984</td>
<td>15.00</td>
<td>10.75</td>
<td>0.80</td>
<td>0.40</td>
<td>5.00</td>
<td>9.75</td>
<td></td>
</tr>
<tr>
<td>(2.0,2.0)</td>
<td>4.573</td>
<td>13.059</td>
<td>20.60</td>
<td>17.75</td>
<td>0.60</td>
<td>0.60</td>
<td>7.20</td>
<td>8.75</td>
<td></td>
</tr>
<tr>
<td>(2.0,0.0)</td>
<td>3.949</td>
<td>18.389</td>
<td>6.40</td>
<td>20.40</td>
<td>0.90</td>
<td>0.50</td>
<td>6.20</td>
<td>10.00</td>
<td></td>
</tr>
<tr>
<td>(0.5,2.0)</td>
<td>7.527</td>
<td>11.178</td>
<td>16.80</td>
<td>20.60</td>
<td>1.00</td>
<td>0.50</td>
<td>8.00</td>
<td>10.20</td>
<td></td>
</tr>
<tr>
<td>Total Average</td>
<td>5.105</td>
<td>12.498</td>
<td>16.733</td>
<td>17.342</td>
<td>0.833</td>
<td>0.517</td>
<td>6.133</td>
<td>9.658</td>
<td></td>
</tr>
</tbody>
</table>

HSV exhibits a superior return rate compared to HiSS. On average, the return rate of HSV is 61% larger than that of HiSS. We believe this is because HSV allows the robot to have high-level information of the position of features in its environment. On the other hand, HiSS allows the robot to have position info of its features only relative to the image space. Moreover, HSV has the advantage of having depth-info of each feature.

On average, HiSS requires 58% more steps than HSV. We believe this is because HSV has knowledge of the distance to $g$, whilst HiSS does not. If HiSS was equipped with some way of acquiring distance info; for example, by using an odometer, then it would be possible for the robot running HiSS to reach the goal in fewer steps.

### 5. Conclusions and Future Work

We have proposed a novel approach to visual homing using a stereo camera. Similar to HiSS, our method does not require the robot’s orientation from the home location to be known a priori. The only assumption that HSV requires is that there must be no obstructions between the robot and the home location. The robustness of both methods relies on having a number of SIFT feature matches. By comparing live robot trials using single frame images, we have shown that the angular error, and the position error of HSV are significantly less than that of HiSS. Moreover, we have shown that HSV uses fewer steps than HiSS for homing by comparing the number of homing steps. From these observations, we are able to conclude that HSV is an efficient, and accurate visual homing method. HSV requires a stereo camera to detect visual disparity in the scenes, making it more complex than methods that use an omnidirectional camera. However, many animals have two eyes that exhibit stereo overlap; even those without significant stereo overlap might use other depth perception cues; for example, by Accommodation i.e. changing the focal length by stretching the ciliary muscles to focus on objects that are further away.
Furthermore, HSV requires more time to capture wide-field images than methods that use an omnidirectional camera due to the time taken by the panning and tilting of the pan-tilt unit. HSV could be implemented using stereo-sphere vision, to eliminate the additional time requirements.

Achieving success with the use of single frame images has created the need to test HSV for multiple frame images (wide-field images). We will also compare HSV with HiSS using an omnidirectional camera. Additionally, we could apply HSV using hardware other than a stereo camera such as a Microsoft Kinect.

ACKNOWLEDGEMENTS

Funding for Paramesh Nirmal’s current graduate study at Fordham University comes in part from the Defense Threat Reduction Agency (DTRA), basic research award # HDTRA1-11-1-0038, to Georgia Tech and Fordham University.

REFERENCES