3D MAPPING TECHNIQUES USING A STEREO BOOM ON LOW-FLYING VEHICLES

Peter Fanto
Jason Gassaway
Kevin Kochersberger – Research Associate Professor
Lynn Abbott – Associate Professor
Unmanned Systems Lab
Virginia Tech
114 Randolph Hall
Blacksburg, VA 24060

ABSTRACT

The Unmanned Systems Lab at Virginia Tech has developed a calibrated 1.5 meter (5 feet) long stereo camera boom that can be attached to a low flying aerial vehicle in order to collect imagery for creating dense 3D reconstructions. To construct the 3D terrain maps, we use two separate techniques. The first is an altered version of the semi-global block matching (SGBM) algorithm. This implementation uses sequential stereo image frames to improve the resolution obtained by using SGBM through a bundle adjustment. This technique produces 3D reconstructions with height resolution greater than is possible using standard disparity mapping techniques. The second is an extension of the structure-from-motion program known as Bundler. The original Bundler uses a random set of images to determine camera locations and orientations in a local reference frame, and creates a sparse 3D model of the visible terrain. Our technique uses GPS data to remove the scale ambiguity of Bundler and to orient the models in the world reference frame. We then provide the Bundler results to PMVS software in order to build intricate models of the local terrain. The result of this process is a detailed map which is oriented in world coordinates. This paper describes the hardware, functionality, and limitations of the stereo boom system as well as describing the improvements to the 3D reconstruction algorithms.

KEYWORDS: Terrain mapping, Stereovision, Local dense bundle adjustment, Stereo boom, Unmanned aerial vehicles

INTRODUCTION

High resolution three-dimensional terrain mapping is important to a variety of fields and situations. For example, detailed mapping can provide a search and rescue crew valuable insights before entering an area. A disaster situation, whether natural or man-made, can leave the terrain unrecognizable and render existing maps obsolete. Three-dimensional mapping programs, such as 3D Blacksburg in Virginia (3D, 2011), accommodate urban planning activities by creating virtual city models. These models depict the terrain and building architecture and can aid in the design of new structures. Another application for 3D mapping is in law enforcement. Police officers can potentially use 3D models to determine the most likely escape routes for criminals based on the obstacles and buildings in an area. Other programs have been developed to create intricate building models for photo tourism (Snavely, 2010). Such programs allow users to experience historic sites within the comfort of their own homes, avoiding the expense and hassles of actual travel.

Many current 3D terrain mapping techniques rely on high altitude imagery or LiDAR to estimate the general layout of the terrain. However, these techniques are often expensive and produce 3D terrain maps with relatively low horizontal resolution. As an alternative to these approaches, the Unmanned Systems Lab at Virginia Tech has developed a 1.5 m long stereo camera boom that can be attached to a low flying aerial vehicle in order to capture imagery. These images can then be post-processed in order to create dense 3D reconstructions. This paper describes two algorithms that have been developed at the Unmanned Systems Lab to provide useful 3D reconstructions of a localized terrain.

The first mapping technique relies heavily on two open source pieces of software, Bundler and Patch-based Multi-View Stereo (PMVS). Bundler is a structure-from-motion system that uses a set of arbitrarily placed images to determine where the cameras were located and to create a sparse reconstruction (Snavely, 2010). PMVS is a dense reconstruction technique that can be used in conjunction with Bundler to provide intricate 3D models of an area...
GBMs are missing. The feature points between images to determine which ones contain similar views. At this point, there are three dense reconstructions, the three-dimensional position of points is optimized through a bundle adjustment. The result is an increased vertical resolution in comparison with the unmodified SGBM technique.

This paper will begin with a review of some previous works pertinent to our algorithms. The next section will then describe the aerial platform and the hardware that comprises the stereo boom system. This will be followed by an in-depth look at the mapping techniques that we developed. Afterwards, the results of the algorithms will be presented and conclusions will be drawn.

**PREVIOUS WORKS**

There are a large collection of works which attempt to create 3D maps of an area. Snavely et al. created the structure from motion program, Bundler, for photo tourism. Bundler is open source software available online (Snavely, 2010). The program uses unstructured and random photographs to perform a sparse bundle adjustment which determines where the cameras were located when the images were taken. It first extracts feature points within the images. Then, it matches the feature points between images to determine which ones contain similar views. At this point, the program performs a sparse bundle adjustment to determine camera positions and orientations based on the locations of the corresponding points in multiple images. The end result is a sparse 3D reconstruction of the environment that was being observed and the estimated camera poses (Snavely, 2007). The sparse reconstruction that is created from Bundler provides a general illustration of an area, but many terrain details are missing.

Furukawa developed Patch-based Multi-View Stereo (PMVS) which utilizes feature point matches between images and camera locations to build a more complete model. The PMVS software is available online in an open source format (Furukawa, 2010b). The software can be used in conjunction with Bundler to create a dense terrain map. It works by gathering dense feature points from Harris corner detection and a difference of Gaussians technique, and then explores the patches around the feature points. For each feature point that is viewed in multiple images, the area adjacent to the feature points is examined to estimate the surface normal directions based on photometric discrepancy function. Then, an optimization procedure estimates the 3D position of the patches. The end result is a three-dimensional dense reconstruction of the area (Furukawa, 2010a).

The Semi-Global Block Matching (SGBM) technique was developed to create dense 3D reconstructions based on a single stereo image pair (Hirschmuller, 2005). SGBM is a multidirectional dynamic programming technique that uses a cost function to estimate the 3D locations that correspond to matching pixels between images. The cost function is based on image disparities, the probability that the pixel location was estimated correctly, and a surface continuity constraint. By finding the lowest cost for all of the pixels, a dense reconstruction can be created. The result of running SGBM is a smooth map of the three-dimensional environment. One version of the SGBM algorithm is available in the open source software OpenCV (OpenCV, 2010) and will be discussed more thoroughly in the description of the Local Dense Bundling Algorithm.
HARDWARE

This section will introduce the hardware that we have used to capture all of our imagery. The aerial platform and the current and future setups of our stereo boom system will be described.

Yamaha RMAX

The Yamaha RMAX helicopter is the current flight platform used to carry the stereo boom and autonomously gather the aerial images. The helicopter itself has a mass of 94 kg and is capable of carrying 28 kg worth of payload for a maximum flight time of one hour. The remaining specifications of the RMAX are given in Table 1.

Table 1. Performance specifications of the Yamaha RMAX

<table>
<thead>
<tr>
<th>Specification</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Max payload</td>
<td>28 kg</td>
</tr>
<tr>
<td></td>
<td>Gross weight</td>
<td>94 kg</td>
</tr>
<tr>
<td></td>
<td>Flight duration</td>
<td>60 minutes</td>
</tr>
<tr>
<td></td>
<td>Range (visual observation)</td>
<td>150 m</td>
</tr>
<tr>
<td>Dimensions</td>
<td>Main rotor diameter</td>
<td>3.1 m</td>
</tr>
<tr>
<td></td>
<td>Tail rotor diameter</td>
<td>0.55 m</td>
</tr>
<tr>
<td></td>
<td>Overall length (including rotors)</td>
<td>3.6 m</td>
</tr>
<tr>
<td></td>
<td>Overall height</td>
<td>1.1 m</td>
</tr>
<tr>
<td>Engine</td>
<td>Water cooled, 2 stroke horizontally opposed 2 cylinder</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Displacement</td>
<td>246 cc</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>21 HP</td>
</tr>
<tr>
<td></td>
<td>Fuel</td>
<td>Gasoline / oil mixed 50:1</td>
</tr>
</tbody>
</table>

Current Stereo Boom Setup

The Unmanned Systems Lab currently uses a 1.524 meters (5 feet) long stereo boom constructed from a 0.076 meter (3 inch) diameter carbon fiber tube. The carbon fiber tube is both lightweight and sturdy, which is important in ensuring that the stereo cameras do not move with respect to each other. If the stereo cameras move an appreciable amount, our calibration results would not be useful for the stereo reconstruction.

The boom supports two Sony XCD-U100 grayscale FireWire cameras that are separated by a distance of 1.473 m (58 inches), which are used for the stereo rectification, and one Sony XCD-U100C FireWire color camera, which is used for a color overlay on the stereo map. The purpose behind using the grayscale cameras instead of color cameras for stereovision is that color images have a reduced effective resolution because of the Bayer pattern (Malvar, 2004). All of the cameras have a resolution of 1600x1200 pixels and are outfitted with 8 mm focal length wide angle lenses.

A Microstrain 3DM-GX3-35 inertial navigation system is also located on the boom in order to gather GPS and orientation measurements. These measurements will be used in both of our techniques and will be discussed in the following sections. The entire boom system is powered by an electronics box which is located on the belly of the Yamaha RMAX. The electronics box is powered from the alternator of the helicopter and distributes power to all the attached systems. As well as distributing power, the electronics box houses the image storage and capture hardware, specifically a WinSystems PC104 single-board computer.

The stereo boom is mounted on the rear of the RMAX as shown in Figure 1. The mount itself is composed of two steel brackets and contains a small amount of vibration absorbent foam to isolate the system from the helicopters vibrations. High magnitude vibrations can cause problems with the image capturing and stereo boom calibration. Good isolation was achieved with this mounting system. It has very high damping and an almost non-detectable resonance at approximately 5 Hz.
Figure 1. Stereo boom with cameras, mounted on the RMAX. On the left is a close up of the current stereo boom. The right image shows the stereo boom on the RMAX during flight.

New Stereo Boom

One issue with the current stereo boom setup is that it requires a separate electronics box in order to operate. This limits the ability to quickly switch the boom between platforms. A new stand-alone stereo boom system is currently being developed at the Unmanned Systems Lab. It is the same length as the old boom and is also made from a 0.076 meter (3 inch) diameter tube carbon fiber tube. The cameras are enclosed in vertical carbon fiber tubes which provide more protection for the cameras and lenses. The new design incorporates two color Kappa Zelos-655 cameras. The cameras capture high resolution color images which allow them to be used without large concerns about losses from the Bayer pattern interpolations. Like the old boom, the new boom contains an inertial navigation system to provide GPS and attitude information. The boom has a box containing a FitPC2i for data collection, a LiPo battery for power, and circuitry for distributing the power to the cameras and for hardware triggering the cameras. The hardware triggering feature allows images to be captured simultaneously. In the previous iteration of the boom, the cameras were triggered by software which resulted in a time delay between capturing the images. As the helicopter flew along and gathered images, there was an offset between the images due to the delay. This offset complicates stereo reconstructions by requiring additional image processing.

The new boom can be powered from either an external source or from the internal battery, allowing the user to operate the boom as a standalone system or as part of a larger one. The boom is currently complete from a hardware perspective, as seen in Figure 2; however, software development remains to be finished before the system is usable. Some more of the specifications are shown in Table 2.

Figure 2. This image shows the newly constructed stereo boom. The cameras and lenses are fully enclosed in the vertical tubing for protection. The box in the middle contains all of the electronics required to run the boom. This image contains radial distortion which makes the boom look curved although it is not.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Requirements</td>
<td>2.25 Amps @ 12V</td>
</tr>
<tr>
<td>Battery Life</td>
<td>1.75 hrs</td>
</tr>
<tr>
<td>Weight</td>
<td>3.74kg (8.25 lbs)</td>
</tr>
</tbody>
</table>

Table 2. Additional new boom specifications
TERRAIN MAPPING TECHNIQUES

The Unmanned Systems Lab uses two principal techniques to create the terrain maps. One technique rectifies the 3D reconstructions that Bundler and PMVS output into a geospatial reference frame. Bundler produces models which are not scaled correctly and have a pseudo-random orientation. We therefore use a technique that utilizes GPS information to determine the proper scale and orientation of the model. The second technique, Local Dense Bundle Adjustment, merges multiple spatially separated stereo frames to increase the positional accuracy of the points in the reconstruction. It relies on the SGBM to gather the initial information about point locations and then optimizes the point positions based on the shared information between stereo frames. The following sections will delve into each approach with greater detail.

Geo-rectification of Point Clouds

In many applications, it is important have models of an environment in real world coordinates and at proper scaling. Unfortunately, the available version of Bundler meets neither of these requirements and in turn causes the PMVS results to have the same issue. There are multiple ways to address this problem. One way is to provide the sparse bundle adjustment routine that Bundler uses with estimated camera positions in real world coordinates, as was mentioned as a possible improvement in the Bundler paper (Snively, 2007). However, this approach requires heavy modification of the Bundler software since it is nested within the Bundler architecture. Another solution is to use the camera positions from the Bundler point cloud and perform a coordinate transform to place the cameras in their proper real world locations. This is technique that we selected.

The first step in the process is to run Bundler on a set of images. We use imagery from both cameras because it allows us to perform a scale refinement. Our method does not require stereo imagery, but allows this scale refinement at the end of the process. At this point, Bundler produces the sparse 3D reconstructions that need to be placed in the geodetic coordinate frame. From here, we need to determine where the cameras are located in the Bundler results so that they can be related to a world reference frame. This requires us to know what order the cameras were added in. Bundler adds cameras based on the feature point correspondences between images, and so the camera order is therefore unrelated to the order which the images were captured. We modified Bundler to output the order that it chose the imagery. This modification allowed us to create a correspondence between camera locations in the 3D models and the world reference frame. Each camera in the Bundler output is then assigned to a GPS position, based on which real world camera gathered that image.

The proper coordinate transform can be determined by using a Helmert transformation. The Helmert transformation determines the rotation, translation, and scaling between coordinate frames based on point correspondences (Watson, 2006). Those parameters can be obtained by using seven or more points that have known 3D coordinates in two reference frames. Typically, a least-squared-error procedure is applied to solve for the transformation parameters. Since the Helmert transformation only utilizes one scaling term, all of the axes must be scaled the same. The GPS coordinates therefore cannot be used in their original form because they are in degrees and the altitude is given in meters. To solve this problem, the GPS coordinates are converted into the Universal Transverse Mercator (UTM) coordinate system. These coordinates are in terms of meters.

We can then determine an initial estimate of coordinate transformation based on the point correspondences; however, this is merely an estimate because Bundler will occasionally place the cameras in incorrect locations. These incorrect camera locations generate significant errors in the coordinate transform. We remove these outliers from the data by comparing the newly estimated camera positions, based on the coordinate transformation, to the UTM coordinates that were previously gathered for each camera. If the distance between any two of these points is greater than the mean distance between all of the pairs, it is considered an outlier and removed from the data. This cutoff was chosen empirically because it worked effectively to remove strong outliers while keeping a substantial amount of inliers. The coordinate transformation is again computed and the whole process is repeated. This is performed until the distances between pairs are below the error in the GPS location, which is around 0.5 meters, or only 7 cameras remain (minimum required amount for the Helmert transformation). Further computation proves not to be useful because camera positions are only known to that GPS accuracy. After the proper coordinate transformation is found, the Bundler results are oriented in the world reference frame.

The next step is to perform a final adjustment to the scale of the different 3D maps. Since we use both the left and right images to make the initial reconstruction, we know that the distance between the cameras must be 1.473 meters. So, we perform a technique which is similar to the one we used to find the correct coordinate transformation. We find the distances between right and left camera pairs and compare their mean to the known 1.473 meters. This determines a new scale to adjust the reconstructions to ensure that the cameras are 1.473 meters apart. The newly
oriented Bundler outputs are run through PMVS. This produces dense reconstructions of the area which are properly oriented to UTM coordinates. All of the 3D models are converted back to the GPS coordinate frame.

**Local Dense Bundle Adjustment**

This section describes our other technique, Local Dense Bundle adjustment. The method borrows from earlier work (e.g., Gassaway 2011) in order to optimize dense 3D structure from multiple frames of aerial stereo imagery. It works by gathering sparse 2D feature points, gathering dense stereo information, and performing a sparse bundle adjustment. The main purpose of this algorithm is to improve the vertical resolution of two-image stereo reconstructions by incorporating the forward trajectory of the helicopter to gather multiple stereo frames and then optimize their dense structures.

The Local Dense Bundle Adjustment algorithm operates on pairs of stereo frames. Again, a frame consists of a left and right image, captured simultaneously, along with GPS and INS data. The algorithm first refines the initial stereo camera pose information using a grid-based iteratively reweighted least squares technique, then calculates epipolar rectification between the left images of the two different stereo frames. Semi-global block matching is run between the left and right images of both stereo frames and also between the two rectified left images. The three resulting disparity maps define the dense matches between the stereo frames. The dense matches and the refined stereo camera pose are optimized using a sparse bundle adjustment to produce the final point cloud.

The initial pose refinement begins by gathering sparse 2D feature points, shown (C) of Figure 3, in the images of both stereo frames by way of the Speeded-Up Robust Features (SURF) algorithm (Bay, 2006) which is available in OpenCV. For this paper, it is not necessary to detail how SURF works beyond the fact that the algorithm finds sparse, distinctive features which can be matched between images. The feature points between the two images (left and right) of a stereo frame are compared with a nearest-neighbor search. The stereo boom’s known calibration is used to project the 2D SURF matches from within a stereo frame into 3D SURF points, which are then matched between different stereo frames via a grid-based iteratively reweighted least squares search (IRLS). Initial camera poses are determined by GPS and INS data. The search space across possible rotations and translations for each stereo camera is divided up into a grid and the cost for each grid point is calculated by summing the difference of matching SURF descriptors. The IRLS algorithm then optimizes across this gridded search space to refine the initial camera pose.

To connect points between multiple stereo frames, the correspondence between the two image frames must be determined. The SURF feature points are matched between the left images from both stereo frames. These matched points are used to determine the Fundamental Matrix relating the two left images. The Fundamental Matrix describes the relationship between point correspondences in uncalibrated stereo cameras. It is essential for determining the proper image rectification. The Random Sample Consensus Algorithm (RANSAC) algorithm is used to remove outlier SURF matches with respect to the estimated Fundamental Matrix. After removal from the data set, a new Fundamental Matrix is calculated, which is used to rectify the left camera images of both stereo frames with respect to each other as shown in (D) of Figure 3.

With the images properly rectified, the next step is to find the associated dense stereo information. We have chosen to use the OpenCV adaptation of the SGBM algorithm. The algorithm works by determining the disparity, or the distance between the coordinates of a pixel as shown in different images. It determines pixel correspondence by minimizing a cost function. Let \( p \) be a pixel in one image and let \( q \) be a pixel in the other image. The base cost function is the maximum of the intensity difference between \( p \) and the maximum intensity of \( q \), minus the minimum intensity of \( q \), and zero. The maximum pixel intensity of \( q \) is taken as the maximum of three things. They are the intensity of \( q \), and the intensities half way in between \( q \) and \( q+1 \) and \( q \) and \( q-1 \). The minimum pixel intensity is the smallest of these values. In SGBM, the cost is comprised of the summed cost in a local window around a pixel. This determines the majority of the cost function. SGBM also incorporates a surface continuity constraint to smooth resulting disparity images and improve correspondence in low-texture regions.
A constraint is applied along eight different directions for each neighboring pixel. The cost function is minimized to create a disparity map, which correlates pixels between the two base images. Three disparity maps are calculated, one for the left, right pairs within the two stereo frames and one between the two left images as shown in Figure 4.

The next step is to relate the three dense disparity maps. Let $I_{l,i}$ represent the $i^{th}$ left image, $I_{r,i}$ be the $i^{th}$ right image, $D_{l,iR}$ be the disparity map between the $i^{th}$ stereo frame, and $D_{l,iL}$ be the disparity map between two successive left images. The algorithm begins by creating a list of linked points which we will define as $G_d$. This list contains all of the 3D locations of the points from the disparity maps as well as which images these points can be seen in. So, $G_d = \{ A_1, A_2, ..., A_i, ..., A_n \}$, where

$$A_j = (X_{i,l}, x_{i,1l}, x_{i,2l}, x_{i,3l}, ..., x_{i,nl}, x_{i,nr}) \quad (1)$$

$X_i$ is the 3D location $i^{th}$ point and $x_{i,1l}, x_{i,1r}$ are the 2D coordinates in each image that they are found in. The 3D location of the images can be calculated by a back projection of the 2D points from 2D to 3D. Multiplication of a 2D point location with the back projection matrix, $Q$, gives the location of the point in camera coordinates.

$$w \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & f \\ 0 & 0 & -1/B & (C_{xR} - C_x)/B \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} \quad (2)$$

where $X_{ic} = [X_c, Y_c, Z_c]^T$ is the 3D point in camera coordinates, $C_x, C_y$ is the center point of the rectified left image, $C_{xR}$ is the center point of the rectified right image, $B$ is baseline distance between the stereo cameras, $f$ is the focal length of the cameras, $w$ is a scaling term, and $d$ is the disparity value at $D_{l,iR}(x,y)$. $X_c$ is converted to world coordinates using the refined pose estimate $RT_i = [R \ T]$, where $R$ is a 3x3 orthogonal rotation matrix and $T$ is a 3x1 translation matrix for frame $i$. $R$ and $T$ act as the homogeneous inverse of the projection matrix $P$, such that $P = R^{-1} [I \ -T]$. Both $RT_i$ and $Q$ are referenced to the left camera. The 2D points $x_{i,1l}$ and $x_{i,1r}$ are computed as $x_{i,1l} = (x, y), x_{i,1r} = (x - d, y)$.

After the initial list $G_d$ is created, an index map is created which relates the pixel coordinates of image $I_{il,(x,y)}$ to $G_d(A_j)$ by

$$j = D_{ind}(x,y) \quad (3)$$

All invalid disparities $D_{ind}(x_{invalid_d}, y_{invalid_d})$ are set to $j = -1$. This index map allows for fast updating of the list during the next iteration of the algorithm.

$G_d$ only contains the points from $D_{l,iR}$ at first. This is because there is no previous left image. For each subsequent stereo frame, the dense algorithm creates both $D_{l,iR}$ and $D_{l,iL}$ and can therefore add the points from all of the cameras to the list as shown in Figure 4. New local dense lists are created and populated for all of the frames to follow. Each new list is then merged with the original dense list, $G_d$, to create a complete list of 3D points and image correspondences. It merges these lists by determining whether or not a point in the new list corresponds with a point in $G_d$. If it does, it is added as a 2D coordinate in $A_j$. If it does not, then it is added as a new 3D point. After list merging, a new index map is created from $D_{l,iR}$, which corresponds to the updated list $G_d$.

The next step in the process is to use stereo-constrained local dense bundle adjustment to optimize $G_d$, which will improve the accuracy of the stereo reconstruction. A bundle adjustment simultaneously optimizes the location of 3D points and the pose estimates of the cameras. Our technique is known as a local bundle adjustment, as opposed to global bundle adjustment, because it can run on a subset of the entire dataset. So, the bundle adjustment uses the information from $G_d$ to determine the best location for the points and

ASPRS 2012 Annual Conference
Sacramento, California ♦ March 19-23, 2012
cameras. Currently, we only run the local bundle adjustment on two frames at a time. This is an artifact of the high memory and processing power requirements needed to optimize the large amount of dense stereo points.

The algorithm implements the local bundle adjustment with the sparse bundle adjustment (SBA) C++ package from Lourakis and Argyros (Lourakis, 2009). The SBA code requires three main inputs: a parameter vector \( p \), which contains the 3D point positions and camera pose; a measurement vector \( x \), which contains the original 2D image coordinates of the 3D points as seen from each camera; and a functional relation \( fn \) that relates the two. The dense algorithm supplies SBA with the parameter vector

\[
p = (RT_i, RT_{i-1}, RT_{i-2}, ..., RT_{i-N}, X_1, X_2, ..., X_j)
\]

for all points \( j \) in \( G_d \) and the total number of frames, \( N \). The measurement vector \( x_m \) consists of the camera measurements

\[
x_m = \begin{bmatrix}
x_{(i)}L & x_{(i)}R & x_{(i-1)}L & x_{(i-1)}R & x_{(i-2)}L & x_{(i-2)}R & \cdots & x_{(i-N)}L & x_{(i-N)}R \\
x_{(i+1)L} & x_{(i+1)R} & x_{(i+2)L} & x_{(i+2)R} & x_{(i+3)L} & x_{(i+3)R} & \cdots & x_{(i-N-1)R} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_{(i)R} & x_{(i+1)R} & x_{(i+2)R} & x_{(i+3)R} & x_{(i+N)}L & x_{(i+N)}R & \cdots & x_{(i-N)R}
\end{bmatrix}
\]

(4)

(5)

Since not all 3D points are visible in every camera, the 2D coordinates for invisible points are supplied as null values and are not used by the SBA software. The functional relationship \( fn \) is defined as a projection of each 3D point onto a stereo camera system,

\[
f = \begin{bmatrix} f_{n_{11}} & f_{n_{21}} & \cdots & f_{n_{1L}} & f_{n_{2L}} & \cdots & f_{n_{1R}} & f_{n_{2R}} & \cdots & f_{n_{iL}} & f_{n_{iR}} \end{bmatrix}
\]

\[
f_{ni} = \begin{bmatrix} \hat{x}_{iL} & \hat{x}_{iR} \end{bmatrix}
\]

\[
\hat{f}_{ni} = \begin{bmatrix} K \cdot P_l \cdot X_j & K \cdot P_{IR} \cdot X_j \end{bmatrix}
\]

(6)

(7)

(8)

where \( P_l \) is the projection matrix for the left camera of frame \( f_i \), obtained from inverting the camera pose \( RT_i \) so that \( P_l = R_i^{-1} [I, -T_i] \). \( K \) is a matrix which contains the intrinsic camera parameters. \( P_{IR} \) is the projection matrix for the right camera, which is found by inverting camera pose \( RT_i \) and adding the stereo camera horizontal baseline \( B \) to produce

\[
P_{IR} = R_i^{-1} [I, (-T_i + \begin{bmatrix} B \\ 0 \end{bmatrix})]
\]

(9)

Stereo-constrained sparse bundle adjustment functions by minimizing the distance of the residual error \( \epsilon^T \epsilon \), where the residual error is given by \( \epsilon = [x_{ijL} - \hat{x}_{ijL}, x_{ijR} - \hat{x}_{ijR}] \). The stereo-constrained SBA operates thrice to optimize the local dense results. Initially, the SBA optimizes the pose refinement from IRLS and GPS/IMU across a smaller subset of the dense point list. Due to the time consuming nature of optimizing hundreds of thousands of points, and since only a fraction of the points are needed for good pose estimate, a small subset is sufficient to optimize the camera pose. The algorithm extracts a random subset points \( G_r \), arbitrarily set at a level over 9000, from the dense list \( G_d \). Both pose and 3D points are optimized across \( G_r \) using stereo-constrained SBA for frames \( f_i, f_{i-1}, ..., f_{i-(N-1)} \), while \( f_{i-N} \) is held fixed. Then, the algorithm passes the results through an error-checking function to remove outliers based on their reprojection error.

After optimizing camera pose, the dense algorithm fixes all camera poses and optimizes the structure of points within \( G_d \) that are visible only in the last frame, \( f_{i-N} \), to produce the optimized list \( \hat{G}_d \). After optimizing the structure of \( \hat{G}_d \) to form \( G_d \), the dense algorithm then performs error checking to detect and remove mistakes from the SGBM procedure. The SBA optimization process minimizes a stereo constraint with projection error; however, the local dense bundling algorithm can also use the known distance accuracy of the stereo system. The accuracy of the stereo system is directly dependent on the baseline of the stereo system, the altitude, and the focal length of the cameras. The local dense bundling algorithm uses both projection error and the vertical accuracy to reject points. Before optimization, a copy of the un-optimized 3D point positions \( X_{est} = \{X_1, X_2, ..., X_j\} \) is created. After local bundling, the optimized results are extracted from \( \hat{G}_d \) in the form \( \hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_j) \). The local dense bundling algorithm calculates the distance each point moved during optimization and compares this distance to the expected stereo.
accuracy with respect to the left camera in frame $f_{i-N}$. Expected stereo accuracy, acts as a maximum acceptable translation in 3D space for each point $X_j$, and is calculated by the following

$$\Delta X_{\text{max}, j} = \frac{T_{z,i-N} \Delta d}{Bf} \|T_{i-N} - X_j\|$$  

(10)

where $\Delta X_{\text{max}, j}$ is the maximum acceptable movement for a 3D point $X_j$, $T_{i-N}$ is the 3D location of the left camera for frame $f_{i-1}$ as stored in $RT_{i-1}$, $T_{z,i-N}$ is the $z$ component of $T_{i-N}$, $B$ is the baseline distance between cameras, $f$ is the focal length, and $\Delta d$ is the minimum value of one pixel of disparity change. If the optimized movement of $X_j$ is greater than $\Delta X_{\text{max}, j}$, the point is rejected as a mismatch. Also, if the reprojection error in any of the cameras exceeds 2 pixels, heuristically determined by experimentation, the point is rejected as a mismatch. After error checking, the local dense bundling algorithm replaces $G_d$ with the optimized, error-corrected list $G^\prime_d$ and moves to the next frame $i + 1$ to continue iteration until all frames are optimized. The end results are 3D point clouds which have more accurate positions than the single-frame stereovision counterparts.

RESULTS

This section will present the results from the georectification of point clouds and the Local Dense Bundling algorithms. The rectified PMVS point clouds, which came from real flight imagery, will be presented and discussed. The Local Dense Bundling section will provide results and analysis from both synthetic and real flight imagery. The synthetic imagery was a simulated flight over a terrain built and imaged through the open source software Blender (Blender, 2011). These results will then be compared against single frame stereovision results of the same area.

The real flight imagery, which was used for both algorithms, was gathered with an RMAX flight over a small patch of woods at Kentland Farms in Virginia (approximately 37.19768 latitude and -80.57788 longitude). A W-shaped flight pattern at a height of 40 meters was flown to ensure that Bundler and PMVS could find sufficient overlap between the images to create the reconstructions. The flight occurred in late October. This meant that the leaves had fallen from the trees, which caused issues with some of the reconstructions.

Georectification of Point Clouds Results

After collecting the imagery, the Bundler and PMVS algorithms were run. They produced point clouds of the area which were then processed to be oriented in geodetic coordinates. The results of the rectified Bundler reconstruction can be seen in Figure 5. The Bundler estimated camera positions are shown above the terrain. As you may notice, there are a fair amount of outliers in the estimated positions of the camera. The strongest of these outliers can be seen at top right of Figure 5. The whole flight occurred at forty meters which means that the cameras should all lie roughly on a plane. However, some cameras significantly deviate from that and are therefore outliers. That is what prompted us to use multiple refinements in order to find the correct orientation of the point cloud. A top down view of the rectified PMVS dense stereo is shown in Figure 5. The holes in the PMVS reconstruction are an artifact of the trees without leaves. Unfortunately at this time, we don’t have a metric for gauging how accurate our georectifications are. This is due to the fact that we don’t have ground truth models of the terrain to compare our results with. However, the images appear to be oriented correctly based on satellite imagery at Kentland Farms. So at this time, we can just provide a qualitative measure of the success of the algorithm.
Local Dense Bundling Results

To understand the effectiveness of the Local Dense Bundling algorithm, it is very advantageous to test the model on a terrain with known dimensions. This is difficult to do with real world terrains because there are so many additional factors. For these reasons, we decided to use the open source 3D modeling program, Blender, in order to create an environment with known dimensions. Blender provides the ability to create synthetic terrain and then gather images of the environment from different positions. This allows us to gather imagery as if we were flying above the terrain. The model that we will use is shown in Figure 6.

The flights were initially simulated on 11 stereo frames at heights of 40, 60, and 80 meters with a distance of 8 meters between frames. The distance between the left and right images was the same 1.473 meters as our stereo boom. Both the Local Dense Bundling algorithm and the SGBM were run on the images. The results were then compared in order to gauge the relative effectiveness of the two methods. The average errors (meters) in the pixel locations for both methods are shown in Figure 6. The important thing to notice in the figure is the error between the two methods. The Local Dense Bundling algorithm always has comparable or better results than the SGBM method at all distances. The discontinuities in the data, which can be seen around 50 and 70 meters, are artifacts of how the terrain map was created. As you can see, the Blender terrain model is comprised of multiple tiers. These tiers made it possible to analyze the effects at a wide range of altitudes simultaneously and also brought about the discontinuities.
We also wanted to gauge the effect of varying the distance between the stereo frames. We ran the analysis for distances of 4, 8, and 12 meters between stereo frames at different altitudes. Again, we compare the errors in the results to those of the raw SGBM method which can be seen in Table 3. As you can see, the error gets smaller as the distance between frames gets larger. This is expected because as the cameras are spread apart, the synthetic baseline, or distance between the cameras, allows for increased vertical resolution of the terrain. In the best case (40 meters altitude and 12 meter baseline), the error is reduced by 491% when compared with the raw SGBM. This means that the error of the pixel location is reduced by nearly five times. Noticing that the error decreases as the distance increases, the next logical step is to increase the distance further to improve performance. However, larger distances between cameras results in less overlap between images and low overlap causes failures during the optimization routine. This limits the maximum physical distance between stereo frames.

The results which have been presented thus far illustrate the effectiveness of the algorithm in ideal conditions; however, there is no noise in the data, whether in the images themselves or in the camera location and pose. We therefore tested the algorithm on real aerial imagery. The Local Dense Bundling algorithm resulted in a vertical resolution of around 0.1 meters, while the raw SGBM algorithm created terrain models with around 0.6 meters vertical resolution. These results can be seen in Figure 8. Notice that the distance between the tiers (vertical resolution) in the image is smaller for the Local Dense Bundling. The vertical resolution is a good way to determine how well a method performed because it is related to the amount of error that will arise in the system. The SGBM model appears smoother because it uses sub-pixel estimation which fits the points to a quadratic function. So, points visually appear to have better resolution, but increased accuracy is actually provided only near discontinuities. The effect can be seen clearly in Figure 7. The sub-pixel estimation is something that we would like to incorporate into our system, but has yet to be done. Even without sub-pixel estimation our system increases point accuracy of the raw SGBM algorithm by integrating and optimizing multiple stereo frames.

![Figure 7](image1.jpg)

**Figure 7.** The mountain side on the left shows the results from the raw SGBM method. The one on the right shows the results of the Local Dense Bundling algorithm.

![Figure 8](image2.jpg)

**Figure 8.** The top image depicts the raw SGBM algorithm results and the bottom one shows the Local Dense Bundling results as viewed from a similar vantage point.

<table>
<thead>
<tr>
<th>Altitude (m)</th>
<th>Basic SGBM</th>
<th>4 m baseline</th>
<th>8 m baseline</th>
<th>12 m baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0.4095</td>
<td>0.2167</td>
<td>0.0969</td>
<td>0.0693</td>
</tr>
<tr>
<td>60</td>
<td>0.8244</td>
<td>0.4268</td>
<td>0.2764</td>
<td>0.5212</td>
</tr>
<tr>
<td>80</td>
<td>1.2894</td>
<td>1.0009</td>
<td>0.4503</td>
<td>0.3653</td>
</tr>
</tbody>
</table>

**Table 3.** Average error (in meters) for 3D reconstructed model constructed from ideal imagery across varying altitudes and virtual baselines.
CONCLUSIONS

The Unmanned Systems Lab at Virginia Tech has created a stereo boom system that can be attached to a low flying aerial vehicle in order to gather images. These images can be turned into useful terrain maps by rectifying the outputs of the Bundler and PMVS as well as performing a Local Dense Bundle Adjustment.

We have developed the capability to properly scale and orient the Bundler and PMVS into the world coordinate frame. This is done by correlating the camera positions, as output by Bundler, with the camera positions from the GPS and attitude information. These matches determine a rotation, translation, and scaling by way of the Helmert transformation.

We have demonstrated that the Local Dense Bundling algorithm has greater vertical accuracy than the SGBM stereovision technique. Our technique works by determining 2D correspondences between multiple stereo frames. Dense disparity maps are then generated within frames and in between the left images of sequential stereo frames. These disparity maps provide three-dimensional information about the environment and create a relationship between the different stereo frames. With this correlation, the location of the pixels in the 3D environment can be optimized through a bundle adjustment. The result is increased resolution for the Local Dense Bundling algorithm when compared to the SGBM technique. Using synthetic imagery, the Local Dense Bundling method has demonstrated a reduction in pixel location error up to a factor of 5. This improvement in accuracy should lead to more accurate dense terrain maps from low-flying aerial vehicles.

ACKNOWLEDGEMENTS

We would like to thank our sponsors at Savannah River National Labs for providing us with the resources that we needed to complete this work.

REFERENCES