UAV navigation system using line-based sensor pose estimation

Julien Li-Chee-Ming and Costas Armenakis

Department of Earth and Space Science and Engineering, Geomatics Engineering, Lassonde School of Engineering, York University, Toronto, Canada

ABSTRACT
This work presents a mapping and tracking system based on images to enable a small Unmanned Aerial Vehicle (UAV) to accurately navigate in indoor and GPS-denied outdoor environments. A method is proposed to estimate the UAV's pose (i.e., the 3D position and orientation of the camera sensor) in real-time using only the on-board RGB camera as the UAV travels through a known 3D environment (i.e., a 3D CAD model). Linear features are extracted and automatically matched between images collected by the UAV's onboard RGB camera and the 3D object model. The matched lines from the 3D model serve as ground control to estimate the camera pose in real-time via line-based space resection. The results demonstrate that the proposed model-based pose estimation algorithm provides sub-meter positioning accuracies in both indoor and outdoor environments. It is also that shown the proposed method can provide sparse updates to correct the drift from complementary simultaneous localization and mapping (SLAM)-derived pose estimates.

1. Introduction
UAVs require precise pose estimation when navigating in both indoor and GPS-denied outdoor environments. The possibility of crashing in these environments is high, as spaces are confined, with many moving obstacles. A real-time indirect sensor pose method to estimate the UAV's pose (i.e., the 3D position and orientation of the camera sensor) is proposed. It is based on images captured by an onboard single RGB camera as the UAV travels through a known 3D CAD model of the environment. The 3D model provides the control features for the UAV's pose estimate for trajectory determination and flight control in GPS-denied outdoor and indoor environments.

The indoor/outdoor mapping and tracking system is based on the Arducopter quadrotor UAV (Unmanned Aerial Vehicle) (Figure 1). For prototyping, a Pixhawk autopilot is used comprised of a GPS sensor that provides positioning accuracies of about 3 m and an Attitude and Heading Reference System (AHRS) that estimates attitude to about 3°. The mapping sensors consist of a forward-looking 0.3 MP Minoru 3D stereoscopic web camera and an Occipital Structure 3D scanner, which is a 0.3 MP depth camera, capable of measuring ranges up to 10 m (±1% of the range).

2. Related work
Model-based tracking, also known as markerless natural feature-based tracking, characterizes this field of research. Lahdenoja et al. (2015) reviewed the advances of model-based tracking from 2005 to 2014. Lepetit et al. (2005) reviewed the model-based tracking literature from 1990 to 2005. Model-based tracking is typically used to obtain the 3D position and the orientation (i.e., pose) of the camera when a complete or partial 3D model of the environment pre-exists. Common applications of model-based tracking include augmented reality (AR), robotic navigation (e.g., automotive), and robotic object manipulation. A typical model-based tracker updates the pose frame-by-frame by maintaining image-to-object feature correspondences as the camera moves. The pose is resolved from the feature correspondences, using for example photogrammetric space resection via the collinearity equations. Because of the tracker’s sequential nature, where the current pose is estimated using the previous frame’s pose, it is required that the pose of the first frame is provided, either by a manual process or a separate automated algorithm (Skrypnyk and Lowe 2004; Wiedemann et al. 2008; Li-Chee-Ming and Armenakis 2014).

In model-based tracking, two categories of features are used:
(1) Edge-based, where camera pose is estimated by matching a wireframe 3D model of an object with the real world image edge information. Wuest, Vial, and Stricker (2005) use a gradient-based approach, where model edges are matched with gradients in the image without explicitly extracting the image primitives (i.e., lines or polygons). Lowe (1992) proposed a contour-based method, where image contours, such as straight line segments and the model outlines, are extracted and matched to image contours.

(2) Texture-based techniques rely on information provided by pixel values inside the object’s projection. Basu, Essa, and Pentland (1996) use optical flow, which estimates the relative movement of the object projection onto the image. Jurie and Dhone (2001) use template matching, which applies a distortion model to a reference image to recover rigid object movement. Vacchetti, Lepetit, and Fua (2004) proposed a keypoint-based approach, which takes into account localized features in the camera pose estimation.

Lahdenoja et al. (2015) suggest that the current state-of-the-art and the most widely used methods of model-based tracking are edge-based methods. Edge-based methods are both computationally efficient and relatively easy to implement. They are also naturally stable to lighting changes, even for specular materials, which is not necessarily true of texture-based methods. Liu and Huang (1991) also identified some advantages of using lines rather than point features. First, lines are often easier to extract from a noisy image than point features; second, the orientation of a line may be estimated with sub-pixel accuracy; third, lines have simple mathematical representation; and fourthly, image lines can be easily found in architectural and industrial environments.

3. Methodology

3.1. Review of previous contributions

Despite being in development since the 1990s, model-based tracking has only been applied to tracking small and simple objects in small and uncluttered environments (Lahdenoja et al. 2015). Li-Chee-Ming and Armenakis (2015) assessed the performance of state-of-the-art model-based trackers in larger environments, tracking larger and more complex objects. Specifically, an open source software implementation of a state-of-the-art edge-based model tracker, called ViSP (Visual Servoing Package) (Marchand and Chaumette 2005), was used to track the outdoor facades of a 3D building model using the RGB camera of a UAV. Li-Chee-Ming and Armenakis (2016) used ViSP to track a 3D indoor model using the RGB camera of a UAV. In both situations, the tracker generated the camera’s pose for each image frame with respect to the 3D model’s georeferenced coordinate system. The results revealed ViSP’s shortcomings, which explained why the sizes of objects and working environments in previous work were limited in size and complexity. Their experiments revealed gaps in the trajectory where tracking was lost. This was due to a lack of model features in the camera’s field of view, and also because of rapid camera motion. Further, the pose estimate was often biased due to incorrect edge matches. Lastly, ViSP specifies that the faces of the model are not systematically modeled by triangles because the lines that appear in the model must match image edges. Due to this condition, 3D models must be converted from the commonly used triangular mesh model to a 3D polygon model. An algorithm was developed to perform this task (Li-Chee-Ming and Armenakis 2015), but it is preferred not to process the reference data (i.e., the 3D model).

In this work, an improved model-based tracker that overcomes the aforementioned deficiencies is proposed. The experiments demonstrate two novel applications of the proposed model-based tracking.
(1) An improved navigation technology for bridging GPS gaps and aiding the navigation sensors (GPS/IMU) using only one RGB camera.

(2) A standalone indoor navigation technology for UAVs using only one RGB camera.

The proposed model-based tracker uses the Gazebo Robot Platform Simulator (Open Source Robotics Foundation, 2014). Gazebo is capable of simulating robots and a variety of sensors in complex and realistic indoor and outdoor environments. An additional contribution of this project is utilizing Gazebo in real-time applications. That is, as the model-based tracker operates, the pose of a virtual camera is dynamically updated in the Gazebo simulator using the estimated pose of the previous frame. The proposed model-based tracker automatically matches features from the real camera image with features in the virtual camera image, referred to herein as a synthetic image, to estimate the current pose. Notably, other graphics frameworks may be used instead of Gazebo to render the model and manipulate the virtual camera. Popular options are Unreal Engine (http://unrealengine.com) and Unity (http://unity3d.com). Hempe (2016) provides a survey of applicable graphics and simulation frameworks. An alternative would be to implement software to project visible parts of the 3D model to the real camera’s image plane, as done within ViSP (Marchand and Chaumette 2005). However, the aforementioned graphics frameworks are specialized in such tasks, focusing on quality and performance. Firstly, they optimize rendering to allow for large-scale, high-LOD (level of detail) 3D models to be visualized in real-time with low latency, which enables the use of low-powered CPUs. Secondly, they offer networking capabilities, to enable real-time loading of 3D models from a database. Lastly, they efficiently handle occlusions and ray casting, thus facilitating the automatic collection of 3D object coordinates needed for real-time camera pose estimation. Other applicable features include shading and lighting, physics, and path planning. Therefore, the use of graphics and simulation tools is strongly recommended. The following section describes the workflow of the proposed tracker in more detail.

![Figure 2. The proposed pose estimation workflow.](image-url)
3.2. Visual navigation solution

The proposed approach is applicable in urban environments and is based on a visual navigation solution that automatically identifies corresponding features between the images streamed from a UAV flying and a given 3D model of the environment. The camera pose resolved from the corresponding feature points provides a precise navigation solution with respect to the 3D environment’s coordinate system. The following steps, describe the pose estimation process and Figure 2 illustrates diagrammatically the overall workflow.

1. The user must provide an initial approximation of the camera pose for the first frame. Straight line segments are then extracted from the image frame. These lines are referred to as real image lines. The very simple and efficient line extraction techniques presented by Smith, Reid, and Davison (2010), such the length, midpoint, quarter point and Bresenham walk checks, are used. This line extraction method was selected because it was designed to function in real-time, and it is capable of extracting long, uninterrupted lines, and merging lines based on proximity and parallelism. The extracted lines are then matched with lines extracted from a virtual camera's synthetic image using the same line extraction approach; these are referred to as virtual image lines. This virtual camera is positioned in Gazebo’s environment using the previous frame’s estimated pose. Both real and virtual image lines are represented by their 2D start and end points.

2. Two horizontal lines and two vertical lines are randomly selected from both the real and virtual image lines until the lines satisfy the following conditions: (i) all lines must be longer than a user specified length, (ii) the vertical separation across the image between the two horizontal lines must be larger than a user specified distance, and (iii) similarly, the horizontal separation between the two vertical lines must be larger than a user specified distance (a default value of 100 pixels was used for all three thresholds in the experiments). These three conditions aim to yield the minimum number of lines that can be used to accurately and reliably estimate the camera pose. For both the real and virtual lines, the two horizontal lines are sorted top to bottom, and the two vertical lines are sorted left to right. The top real horizontal line is then paired with the top virtual horizontal line. Similarly, the bottom real horizontal line is paired with the bottom virtual horizontal line, the left real vertical lines and the left virtual vertical lines are paired, and finally the right real vertical line is paired with the right virtual vertical line.

3. The correspondence of these four real-virtual line pairs is tested by a line-to-line (L2L) space resection (Meierhold, Bienert, and Schmich 2008). This method of space resection estimates the camera pose given corresponding 2D image lines and 3D object lines. 3D object lines are referred to in this work as virtual object lines and are represented by their 3D start and end points. In this case, the 3D start and end points of a virtual object line are calculated via ray-casting, that is, by casting a ray starting from the virtual camera, through its respective 2D image point, and determining where it intersects with the 3D model. The real and virtual image lines are considered correctly matched if the pose covariance matrices from the space resection are within a user defined tolerance and if the estimated camera pose is within the error ellipsoid of the camera pose predicted by the Kalman filter of the navigation system. If either condition is not met, four new lines are randomly sampled from the sets of real and virtual lines. Matching lines via this line-to-line resection was chosen because it does not require the corresponding lines to have corresponding start and end points. This is a very important ability because this requirement is seldom met, due to occlusions for instance.

4. A more accurate pose estimate can then be obtained by a point-to-line (P2L) space resection (Meierhold, Bienert, and Schmich 2008). This approach not only solves for the camera pose, but also estimates the 3D coordinates of the corresponding object point for each real image line's start and end points.

5. Finally, because corresponding image and object points are now available (from the P2L resection), a traditional point-to-point (P2P) space resection is done. This provides a more accurate estimate for the camera pose than the point-to-line approach because the measurement redundancy is higher (i.e., the point-to-point resection only estimates the six camera pose parameters while the point-to-line space resection estimates the six camera pose parameters plus one additional parameter for every 2D image point measurement). This parameter identifies the point on the 3D model where the ray from the UAV’s camera, passing through the real image point, intersects the 3D virtual object line (Meierhold, Bienert, and Schmich 2008).

For example, Figure 3A shows a sample image captured by the UAV’s camera and Figure 3B shows its corresponding synthetic image in the Gazebo simulator.
4.1. UAV localization in GPS-denied outdoor environments

This experiment assessed the proposed model-based tracker in tracking a georeferenced 3D building model using the RGB camera of a UAV for the purpose of UAV localization in GPS-denied outdoor environments. Previous model-based trackers have been demonstrated to perform well in tracking small, simple objects in small and cluttered indoor environments (Lepetit et al. 2005; Lahdenoja et al. 2015). This work demonstrates a novel model-based tracker capable of tracking larger, complex objects (i.e., building models) in large outdoor environments by UAVs.

The 3D virtual building model of York University’s Keele Campus (Armenakis and Sohn 2009) was used as a known environment. The model consists of photorealistic 3D TIN (Triangulated Irregular Network) reconstructions of buildings, trees, and terrain (Figure 4). The model was generated from building footprint vector data, Digital Surface Models (DSM) with 0.75 m ground spacing, corresponding orthoimages at 0.15 m spatial resolution, and terrestrial images. The 3D building models

Figure 3. Line matching example: Real image (A) and corresponding synthetic image (B), each with 2 randomly sampled horizontal lines, sorted top to bottom (labeled as 1 and 2) and 2 randomly sampled vertical lines, sorted left to right (labeled 3 and 4). The real image lines’ corresponding virtual image line start and end points estimated by the point-to-line space resection (C).

4. Data collection

The following experiments demonstrate two novel applications of model-based tracking methods:

1. A UAV localization technology for bridging GPS gaps and aiding other navigation sensors (GPS/IMU) using only one RGB camera.
2. An indoor localization technology for UAVs using only one RGB camera.

Figure 4. York University 3D campus model.
were further refined with airborne lidar data having a point density of 1.9 points per square meter (Corral-Soto et al. 2012). The 3D CAD model served two purposes in the proposed approach. First, it provided the necessary level of detail such that individual buildings could be uniquely identified by the model-based tracker. Second, it provided ground control points to photogrammetrically estimate the camera pose. The geometric accuracy of the building models is in the order of 10 to 40 cm.

The UAV flew over York University, up to approximately 40 m above the ground. In total, 1137 frames were processed at 6 frames per second. The camera’s intrinsic parameters were calibrated beforehand and held fixed in the pose estimation process. The image frames were tested at 480, 720, and 1080 pixels resolution. Trials revealed that, at 40 m above ground level, the 480 pixels video performed the best in terms of processing speed and tracking performance. The tracking performance improved because the noisy edges detected at higher resolutions were removed at lower resolutions, leaving only the stronger edge responses.

4.2. UAV localization in indoor environments

The second experiment assessed the proposed model-based tracker in tracking a georeferenced 3D indoor model using the camera of a UAV for the purpose of UAV localization in GPS-denied indoor environments. The experiment was performed on the second floor of York University’s Bergeron Centre for Engineering Excellence. The geometric accuracy of the indoor building model is less than 10 cm. The experiment began with the user providing an initial approximation of the camera pose. Corresponding edges between the image sequence and the 3D indoor model were then automatically identified and continuously tracked. At each frame, the corresponding edges were used in a space resection to estimate the camera pose. Figure 5 shows the trajectory of the UAV within the 3D CAD model of the building.

5. Data processing

Li-Chee-Ming and Armenakis (2015) showed that state-of-the-art model-based trackers achieved similar positional accuracies to single point positioning solutions obtained from single-frequency (L1) GPS observations – standard deviations ranging from 2 to 10 m. These are large errors considering the geometric accuracy of the building models were in the order of 10 to 40 cm. Further, Li-Chee-Ming and Armenakis (2016) demonstrated that model-based trackers fail when the model is not present in the camera’s field of view and when features cannot be matched between the image and the model. Failures can occur if the environment has few features or the model has few features, i.e., the model has a low LoD. For these reasons, this work proposes to “bridge the gaps” between sparse, yet stringently checked (i.e., accurate), pose updates from the model-based tracker with high-frequency, and precise, simultaneous localization and mapping (SLAM) pose updates.

Monocular visual SLAM has been widely used by UAVs in outdoor environments. Suitable open-source methods that process RGB image sequences include parallel tracking and mapping (PTAM) (Klein and Murray 2007), large-scale direct monocular (LSD) SLAM (Engel, Schops, and Cremers 2014), and Oriented FAST and Rotated BRIEF (ORB) -SLAM (Mur-Artal, Montiel, and
This work also proposes that model-based trackers can be used to overcome the weaknesses of map-building localization. That is, if loop-closures are not performed or detected, there will be large drift errors in the pose estimates of the map-building localization solution. Further, the memory and computational requirements of map-building algorithms increases significantly as the mapped area grows because of the large amount of data being processed simultaneously. Model-based trackers “close loops” on a pre-existing model, this removes the need in SLAM to maintain the entire map for the purposes removing drift with loop-closures. Instead, the SLAM process can be restarted every time and the loop is closed on the pre-existing map (with a model-based tracker pose update). For example, Figure 6 is a schematic diagram of a mobile camera traversing from a start position (symbolized by a star) to an end position (symbolized by a triangle). Each frame’s estimated position is shown using a circle. The camera pose’s accuracy is symbolized by the circle’s radius. The error is reduced by the model-based tracker updates. In this example, doors appearing in the camera image are matched with doors in the 3D CAD model. The error is reduced by estimating the 3D transformation that aligns the door extracted from the image to the door from the 3D CAD model.

Similarly, Tamaazousti et al. (2011) used points from a partial 3D model, and Prisacariu et al. (2013) used shape information from a very coarse 3D model in a SLAM system, to simultaneously reconstruct the whole scene or refine the 3D model, with pose estimation benefiting from this reconstruction.

Figure 7 shows the strategy for integrating the proposed model-based tracker (MBT) and an existing SLAM method. Firstly, the user initializes the MBT by providing an approximate camera pose estimate of the first frame. SLAM then begins mapping and tracking the camera pose, and continues until the MBT algorithm detects corresponding image and model feature points. The MBT then corrects the camera pose and the SLAM method is reset at the MBT’s provided pose.

Figures 8 and 9 show five frames from the outdoor and indoor scenarios, respectively, where the model-based tracker pose updates controlled the drift error in the SLAM-derived pose. The top row shows the real image captured by the UAV’s camera. In the bottom row, below each image of the top row it is its corresponding synthetic image. The red lines show the two horizontal lines (labeled as 1 and 2) and two vertical lines (labeled as 3 and 4) that were matched using line-to-line space resection. Tables 1 and 2 show the average and standard deviation of the pose standard deviations from the least squares adjustment for the five frames in Figures 8 and 9, respectively. The pose includes the orientation angles (ω, φ, κ) and the position (X_o, Y_o, Z_o) of the camera’s perspective center in relation to a start position (symbolized by a star) to an end position (symbolized by a triangle). Each frame’s estimated position is shown using a circle. The camera pose’s accuracy is symbolized by the circle’s radius. The error is reduced by the model-based tracker updates. In this example, doors appearing in the camera image are matched with doors in the 3D CAD model. The error is reduced by estimating the 3D transformation that aligns the door extracted from the image to the door from the 3D CAD model.

**Table 1.** Outdoor experiment: average (̄x) and standard deviation (σ_0) of the pose standard deviations from the least squares adjustment for the 5 frames in Figure 8.

| Method | σ_x ± σ_x (m) | σ_y ± σ_y (m) | σ_z ± σ_z (m) | ̄ω_x ± σ_ω_x (°) | ̄κ_x ± σ_κ_x (°) | ̄φ_x ± σ_φ_x (°) |
|--------|----------------|----------------|----------------|------------------|------------------|------------------|
| L2L    | 1.9801 ± 1.7554 | 1.0257 ± 0.4939 | 1.1548 ± 0.6985 | 1.7651 ± 0.7624  | 2.5170 ± 2.3949  | 1.0366 ± 0.4400  |
| P2L    | 1.2640 ± 1.3191 | 1.0775 ± 0.6008 | 0.8854 ± 0.3753 | 1.7061 ± 0.6505  | 1.6619 ± 1.7744  | 0.9746 ± 0.5706  |
| P2P    | 0.2199 ± 0.0498 | 0.3322 ± 0.0674 | 0.2528 ± 0.0208 | 0.4761 ± 0.1109  | 0.3214 ± 0.0890  | 0.2676 ± 0.1005  |

**Table 2.** Indoor experiment: average (̄x) and standard deviation (σ_0) of the pose standard deviations from the least squares adjustment for the 5 frames in Figure 9.

| Method | σ_x ± σ_x (m) | σ_y ± σ_y (m) | σ_z ± σ_z (m) | ̄ω_x ± σ_ω_x (°) | ̄κ_x ± σ_κ_x (°) | ̄φ_x ± σ_φ_x (°) |
|--------|----------------|----------------|----------------|------------------|------------------|------------------|
| L2L    | 0.1277 ± 0.1231 | 0.1273 ± 0.1278 | 0.1494 ± 0.1207 | 2.4163 ± 2.5766  | 1.4356 ± 1.2055  | 1.6718 ± 1.9057  |
| P2L    | 0.0852 ± 0.0426 | 0.1022 ± 0.0692 | 0.0930 ± 0.0278 | 1.8315 ± 1.2969  | 1.0480 ± 0.6618  | 1.5405 ± 1.3427  |
| P2P    | 0.0072 ± 0.0050 | 0.0068 ± 0.0034 | 0.0118 ± 0.0071 | 0.2403 ± 0.1826  | 0.1099 ± 0.0819  | 0.1815 ± 0.1436  |
the point-to-line space resection. The point-to-point space resection provided the most accurate estimates. In both experiments, sub-meter positioning accuracies were achieved from the P2P space resection.

The green circles in Figures 8 and 9 indicate check points. The 3D object coordinates of these points were estimated by the point-to-line resection and compared with the ground truth coordinates (manually extracted from the model's vertices). Tables 3 and 4 show the differences between the estimated 3D object coordinates and the true model coordinates from the outdoor and indoor scenarios, respectively. There were 19 check points from horizontal lines and 21 check points for vertical lines for outdoor scenario. The indoor scenario had 9 check points from horizontal lines and 10 check points for vertical lines. In both experiments, the $Z$ component of the vertical lines was the noisiest. This could be due to the distribution of vertical lines across the images, outliers, or inaccuracies of the 3D model. In both the indoor and outdoor cases, the standard deviations are within the accuracies of their respective 3D models, suggesting that the proposed tracker effectively identified corresponding features and rejected outlier matches.

Figure 7. MBT / SLAM integration workflow.

to the object reference system, this is also referred to in literature as the exterior orientation parameters (EOPs). As expected, the line-to-line space resection yielded the least accurate pose estimates, followed by the point-to-line space resection. The point-to-point space resection provided the most accurate estimates. In both experiments, sub-meter positioning accuracies were achieved from the P2P space resection.

The green circles in Figures 8 and 9 indicate check points. The 3D object coordinates of these points were estimated by the point-to-line resection and compared with the ground truth coordinates (manually extracted from the model's vertices). Tables 3 and 4 show the differences between the estimated 3D object coordinates and the true model coordinates from the outdoor and indoor scenarios, respectively. There were 19 check points from horizontal lines and 21 check points for vertical lines for outdoor scenario. The indoor scenario had 9 check points from horizontal lines and 10 check points for vertical lines. In both experiments, the $Z$ component of the vertical lines was the noisiest. This could be due to the distribution of vertical lines across the images, outliers, or inaccuracies of the 3D model. In both the indoor and outdoor cases, the standard deviations are within the accuracies of their respective 3D models, suggesting that the proposed tracker effectively identified corresponding features and rejected outlier matches.

Figure 8. Outdoor experiment: Tracking a 3D building model using the camera of a UAV.

Figure 9. Indoor experiment: Tracking a 3D indoor model using the camera of a UAV.
Table 3. Outdoor experiment’s check point statistics: the averages and standard deviations of the differences between ground truth model coordinates and object coordinate calculated by the point-to-line resection for horizontal (H) lines and vertical (V) lines.

| Line | No. | \( \Delta X \pm \sigma_{\Delta X} \) (m) | \( \Delta Y \pm \sigma_{\Delta Y} \) (m) | \( \Delta Z \pm \sigma_{\Delta Z} \) (m) |
|------|-----|---------------------------------------|---------------------------------------|---------------------------------------|
| H    | 19  | 0.009 ± 0.240                        | −0.221 ± 0.440                        | 0.026 ± 0.221                        |
| V    | 21  | 0.001 ± 0.300                        | −0.006 ± 0.050                        | −0.117 ± 1.065                       |

Table 4. Indoor experiment’s check point statistics: The averages and standard deviations of the differences between ground truth model coordinates and object coordinate calculated by the point-to-line resection for horizontal (H) lines and vertical (V) lines.

| Line | No. | \( \Delta X \pm \sigma_{\Delta X} \) (m) | \( \Delta Y \pm \sigma_{\Delta Y} \) (m) | \( \Delta Z \pm \sigma_{\Delta Z} \) (m) |
|------|-----|---------------------------------------|---------------------------------------|---------------------------------------|
| H    | 9   | −0.007 ± 0.04                         | 0.001 ± 0.017                         | −0.001 ± 0.005                        |
| V    | 10  | −0.001 ± 0.01                         | −0.009 ± 0.022                        | 0.007 ± 0.090                         |

6. Conclusions

This work provided a navigation solution for UAVs in GPS-denied outdoor and indoor environments using only the onboard RGB camera. An improved model-based tracker capable of tracking large and complex objects in large environments has been implemented to provide a real time estimate of the relative pose between the camera and a georeferenced 3D model.

This work demonstrated that by integrating the MBT and a SLAM method, not only SLAM improve MBT’s performance, but the MBT substitutes SLAM’s loop closing, while improving runtime performance. The proposed integrated solution is analogous to GPS/IMU integration, where the MBT provides a low-frequency and drift-free pose estimate that is registered to a (geo-referenced) 3D model, similar to GPS. Complementarily, the SLAM solution provides a high frequency pose estimate that drifts over time, similar to the behavior of an Inertial Measurement Unit (IMU).

The results demonstrated that the proposed model-based pose estimation algorithm provides sub-meter positioning accuracies in both indoor and outdoor environments. Future work includes automating the tracker’s initialization using only the RGB camera, integrating the proposed visual navigation solution with the UAV’s autopilot solution, and extension to non-urban environment using digital surface models.

Acknowledgments

We thank the Planning & Renovations, Campus Services & Business Operations at York University for providing the 3D model of the Bergeron Centre of Excellence in Engineering. Special thanks are given to the members of York University’s GeoICT Lab, Damir Gumerov, Yoonseok Jwa, Brian Kim, and Yu Gao, who contributed to the generation of the York University 3D Virtual Campus Model. We also thank Professor James Elder’s Human & Computer Vision Lab in York University’s Centre for Vision Research for providing the UAV video data.

Funding

This work was supported by NSERC through a Discovery Grant [grant number RGPIN-2015-06211].

Notes on contributors

Julien Li-Chee-Ming holds a PhD degree from York University, where he also obtained a BSc and MSc in Geomatics Engineering. His research interests include low-cost navigation systems, rapid mapping, and urban 3D modeling. His research involves utilizing unmanned aerial vehicles in real-time mapping, monitoring, and tracking applications.

Costas Armenakis, PhD, PEng, is an associate professor and undergraduate program director of the Geomatics Engineering program, York University. His research interests are in the areas of photogrammetric engineering and remote sensing mapping, focusing on unmanned mobile sensing and mapping systems and the use of unmanned aerial vehicle systems for geomatics applications.

ORCID

Julien Li-Chee-Ming [http://orcid.org/0000-0002-8646-3991]

References

Armenakis C., and G. Sohn. 2009. “[Campus: 3D modeling of York University Campus.” Paper presented at the 2009 Conference of the American Society for Photogrammetry and Remote Sensing, Baltimore, MD, March 9–13.

Basu S., I. Essa, and A. Pentland. 1996. “Motion Regularization for Model-Based Head Tracking.” Paper presented at the 13th International Conference on Pattern Recognition, Vienna, August 25–29, 611–616.

Corral-Soto E. R., R. Tal, L. Wang, R. Persad, L. Chao, S. Chan, B. Hou, G. Sohn, and J. H. Elder. 2012. “3D Town: The Automatic Urban Awareness Project.” Paper presented at the 9th Conference on Computer and Robot Vision, Toronto, ON, May 28–30, 433–440.

Endres, F., J. Hess, J. Sturm, D. Cremers, and W. Burgard. 2014. “3D Mapping With An RGB-D Camera.” IEEE Transactions on Robotics 30 (1): 177–187.

Engel J., T. Schops, and D. Cremers. 2014. “LSD-SLAM: Large-Scale Direct Monocular SLAM.” Paper presented at the European Conference on Computer Vision, Zurich, September 6–12, 834–849.

Hempe N. 2016. “Bridging the Gap Between Rendering and Simulation Framework.” Springer Fachmedien Wiesbaden, 244

Jurie F., and M. Dhome. 2001. ”A Simple and Efficient Template Matching Algorithm.” Paper presented at the Eighth IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, July 7–14, 544–549.

Klein G., and D. Murray. 2007. “Parallel Tracking and Mapping for small AR Workspaces.” Paper presented at the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, November 13–16, 1–10.

Lahdenoja O., R. Suominen, T. Säntti, and T. Lehtonen. 2015. “Recent advances in monocular model-based tracking: A systematic literature review.” University of Turku Technical Reports 8.
Lepetit, V., and P. Fua. 2005. "Monocular Model-Based 3D Tracking of Rigid Objects." *Foundations and Trends in Computer Graphics And Vision* 1 (1): 1–89.

Li-Chee-Ming, J., and C. Armenakis. 2014. "Feasibility Study for Pose Estimation of Small UAS in Known 3D Environment Using Geometric Hashing." *Photogrammetric Engineering & Remote Sensing* 80 (12): 1117–1128.

Li-Chee-Ming J., and C. Armenakis. 2015. "A Feasibility Study on Using ViSP’S 3D Model-Based Tracker for UAV Pose Estimation in Outdoor Environments." Paper presented at the UAV-g 2015 International Conference on Unmanned Aerial Vehicles in Geomatics, Toronto, August 30–September 2, XL-1/W4: 329–335.

Li-Chee-Ming J., and C. Armenakis. 2016. "Augmenting ViSP’s 3D Model-Based Tracker with RGB-D SLAM for 3D Pose Estimation in Outdoor Environments." Paper presented at the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, July 12–19, XLII-B: 925–932.

Liu, Y., and T. S. Huang. 1991. "Determining Straight Line Correspondences from Intensity Images." *Pattern Recognition* 24 (6): 489–504.

Lowe, D. 1992. "Robust Model-Based Motion Tracking Through the Integration of Search and Estimation." *International Journal of Computer Vision* 8 (2): 113–122.

Marchand, E., and F. Chaumette. 2005. "Feature Tracking for Visual Servoing Purposes." *Robotics and Autonomous Systems* 52 (1): 53–70.

Meierhold N., A. Biemert, and A. Schmich. 2008. "Line-Based Referencing Between Images and Laser Scanner Data for 1 Mage Based Point Cloud Interpretation in a Cad-Environment." Paper presented at the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, XXXVII-B5: 437–443.

Mur-Artal, R., J. Montiel, and J. Tardós. 2015. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System." *IEEE Transactions on Robotics* 31 (5): 1147–1163.

Open Source Robotics Foundation. 2014. Gazebo, robot simulation made easy. Accessed January 1, 2017. [http://gazebosim.org/](http://gazebosim.org/)

Prisacariu V., O. Kähler, D. Murray, and I. Reid. 2013. "Simultaneous 3D Tracking and Reconstruction on a Mobile Phone." Paper presented at the IEEE International Symposium On Mixed and Augmented Reality (ISMAR), Adelaide, SA, October 1–4, 89–98.

Skrypnyk I., and D. Lowe. 2004. "Scene Modelling, Recognition and Tracking with Invariant Image Features." Paper presented at the IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA, November 2–5, 110–119.

Smith P., I. Reid, and A. Davison. 2010. "Real-Time Monocular SLAM with Straight Lines." Paper presented at the British Machine Vision Conference (BMVC), Edinburgh, August 31–September 3, 7–17.

Tamaazousti M., V. Gay-Bellile, S. Collette, S. Bourgeois, and M. Dhome. 2011. "Nonlinear Refinement of Structure from Motion Reconstruction by Taking Advantage of a Partial Knowledge of the Environment." Paper presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, June 20–25, 3073–3080.

Vacchetti, L., V. Lepeit, and P. Fua. 2004. "Stable Real-Time 3D Tracking using Online and Offline Information." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (10): 1385–1391.

Wiedemann U., and C. Steger. 2008. "Recognition and Tracking of 3D Objects." *Pattern Recognition, Lecture Notes in Computer Science* (Volume 5096), Berlin: Springer-Verlag, 132–141.

Wuest H., F. Vial, and D. Stricker. 2005. "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality." Paper presented at the Fourth IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), Vienna, October 5–8, 62–69.