CAMERA POSE ESTIMATION USING VISUAL SERVOING FOR AERIAL VIDEO CHANGE DETECTION

Nicolas Bourdis, Marraud Denis, Hichem Sahbi

CAMERA POSE ESTIMATION USING VISUAL SERVOING FOR AERIAL VIDEO CHANGE DETECTION

Nicolas BOURDIS, Denis MARRAUD

EADS France
Image Solutions Department
Suresnes, France

Hichem SAHBI

CNRS, LTCI Telecom ParisTech
Paris, France

ABSTRACT

Aerial image change detection is highly dependent on the accuracy of camera pose and may be subject to false alarms caused by misregistrations. In this paper, we present a novel pose estimation approach based on Visual Servoing that combines aerial videos with 3D models.

Firstly, we introduce a formulation that relates image registration with the poses of a moving camera observing a 3D plane. Then, we combine this formulation with Newton's algorithm in order to estimate camera poses in a given aerial video. Finally, we present and discuss experimental results which demonstrate the robustness and the accuracy of our method.

Index Terms— Aerial video, visual servoing, pose estimation, 3D appearance modeling.

1. INTRODUCTION

Aerial video monitoring platforms are currently receiving a growing interest as they enable fast and effective ways to analyze vast regions. However, their high frame rates result in huge amounts of data which are difficult to process manually. Hence, designing automatic video processing methods, for different tasks, is crucial in order to alleviate the load of human operators.

In this paper, we focus on the specific task of change detection in aerial videos, as a way to filter out uninteresting data. Change detection [1] refers to the problem of detecting significant and possibly subtle changes between reference and test data (e.g. appearing or disappearing buildings or vehicles), while ignoring insignificant ones. General change detection is particularly challenging due to environmental changes (illumination, weather, ...) and parallax effects (trees, buildings, relief ...) caused by camera motion. Current approaches designed to correctly handle parallax effects either use the epipolar geometry or exploit 3D models. This second category of approaches is particularly interesting as it is more precise when details in scenes are modeled accurately. This modeling is time demanding and usually achieved offline as a pre-processing step. Another pre-processing step is camera pose estimation, performed in order to align test data with respect to 3D reference models prior to achieving comparison and change detection.

In this work, we introduce a method for accurate camera pose estimation based on 3D reference models and Visual Servoing. More precisely, we use a Newton optimization strategy guided by a specific Jacobian matrix in order to actively adjust the parameters of the pose. These parameters are updated until the rendered 3D model and the acquired test images are aligned precisely. Our algorithm follows two steps: the first one predicts *coarse* pose parameters while the second step *refines* these parameters using the 3D reference model. Our experiments clearly show that this scheme is effective in order to estimate the pose parameters accurately.

The remainder of this paper is organized as follows. Section 2 introduces our change detection framework based on 3D appearance modeling. Section 3 describes details about our camera pose estimation algorithm while Section 4 presents experimental results.

2. 3D APPEARANCE MODELING

3D appearance modeling is an extension of the Foreground / Background segmentation framework introduced in [2], which may be applied to change detection in videos acquired by mobile cameras. The main idea is to learn variations of scene appearances (e.g. shadows, illumination, noise ...) from a set of reference data, and organizing these appearance models in the 3D model of the scene. Changes (which may be viewed as Foreground in segmentation terms) may then be detected as observations which deviate from the learned reference model. Following this idea, [3] used a voxel-based approach including Gaussian Mixture Models for appearance modeling. [4] extended this method with a continuous volumetric scene model implemented using an efficient Octree structure. Such models use a full 3D modeling of scenes, which enable accurate reconstructions but may raise convergence issues during reconstruction.

However, in the context of aerial videos, such 3D models put unnecessary effort in modeling scenes as volumetric objects, even though scenes are generally observed from a certain distance and therefore may be more effectively modeled as surfaces with variable heights. This is why, in this work, we used a 3D appearance model based on a Height Map organized in a Quad-Tree structure instead. Each cell of this Quad-Tree includes both height and appearance information.

Given a set of reference videos, the underlying 3D appearance model is generated offline. Heights are initially set using DTM or DEM data and refined using the videos. Given two video frames $\mathbf{I}_n$ and $\mathbf{I}_{n+w}$ (where w is the size of a temporal window), we estimate parallax using optical flow constrained with epipolar geometry [5]. Height is then inferred from parallax disparity using the equations described in [6]. Figure 1 presents visualizations of a 3D appearance model obtained from three aerial videos acquired along different trajectories.

3. CAMERA POSE ESTIMATION

One of the important aspects in change detection is to ensure that reference and test data are well aligned before achieving comparison.

Fig. 1. Left: Visualization of the heights in the 3D appearance model, where the lowest regions are shown in blue and the highest ones in red. Middle: Visualization of the appearance information contained in the model. Right: Model rendering generated from a given viewpoint.

In this section, we present our camera pose estimation approach that allows us to recover the pose parameters accurately.

3.1. Problem Statement

Let C be a moving camera in a given scene and let $\{\mathbf{I}_n\}_{n\in\mathbb{N}}$ be frames acquired by C at different instants. Consider $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$, $\{\mathbf{K}_n\}_{n\in\mathbb{N}}$ as their actual unknown camera poses (i.e. 3D position and 3D orientation) and calibration parameters respectively. In the following, we denote the focal ratios by (fx_n, fy_n), the principal point coordinates by (ox_n, oy_n) and their ratios as $rx_n = \frac{ox_n}{fx_n}$, $ry_n = \frac{oy_n}{fy_n}$.

Starting from inaccurate pose parameters[1] (denoted $\{\widetilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$) and the frames $\{\mathbf{I}_n\}_{n\in\mathbb{N}}$, we are interested in obtaining accurate estimates of these poses (denoted $\{\widehat{\mathbf{X}}_n\}$). The proposed algorithm proceeds in two steps described in sections 3.2 and 3.3. Prior to introducing these steps, we revisit the preliminary background below.

We consider two camera sensors C_1 and C_2, with acquisition parameters $(\mathbf{X}_1, \mathbf{K}_1)$, $(\mathbf{X}_2, \mathbf{K}_2)$, both observing a plane Π (with a normal vector $\mathbf{n}_\pi$ and a shift to the origin $-d_0$). It was shown in [7] that the projective matrix F_H, registering two images acquired by C_1, C_2, may be analytically expressed using the extrinsic and intrinsic parameters of the sensors C_1, C_2 as:

$$\mathbf{p}_2 = F_H(\mathbf{X}_1, \mathbf{K}_1, \mathbf{X}_2, \mathbf{K}_2, \Pi) \cdot \mathbf{p}_1, \tag{1}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are points in each image, expressed in homogeneous coordinates. Under a small variation of the pose between C_1, C_2, one may approach the underlying rigid transformation (rotation and translation) using the following first order approximations:

$$\begin{bmatrix} \mathbf{Vx_2}^t \\ \mathbf{Vy_2}^t \\ \mathbf{Vz_2}^t \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Vx_1}^t \\ \mathbf{Vy_1}^t \\ \mathbf{Vz_1}^t \end{bmatrix}^t \approx \begin{bmatrix} 1 & d\phi & -d\psi \\ -d\phi & 1 & d\theta \\ d\psi & -d\theta & 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} \mathbf{Vx_2}^t \\ \mathbf{Vy_2}^t \\ \mathbf{Vz_2}^t \end{bmatrix} \cdot [\mathbf{c_2} - \mathbf{c_1}] \approx \begin{bmatrix} \mathbf{Vx_1}^t \\ \mathbf{Vy_1}^t \\ \mathbf{Vz_1}^t \end{bmatrix} \cdot [\mathbf{c_2} - \mathbf{c_1}] \equiv \begin{bmatrix} dX \\ dY \\ dZ \end{bmatrix}$$

where V^t stands for the transpose of a matrix V and $d\phi$, $d\theta$, $d\psi$, dX, dY, dZ are the unknown parameters of the rigid transformation between C_1, C_2. In the above equation, $\mathbf{c_1}$, $\mathbf{Vx_1}$, $\mathbf{Vy_1}$, $\mathbf{Vz_1}$ (resp. $\mathbf{c_2}$, $\mathbf{Vx_2}$, $\mathbf{Vy_2}$, $\mathbf{Vz_2}$) correspond to the optical center and the three principal axes of C_1 (resp. C_2) in world coordinates. Considering these definitions, F_H may be linearized as follows:

$$F_H(\mathbf{X}_1, \mathbf{K}_1, \mathbf{X}_2, \mathbf{K}_2, \Pi) \approx \tag{3}$$
$$H_{Id}(\mathbf{K}_1, \mathbf{K}_2) + J(\mathbf{X}_1, \mathbf{K}_1, \mathbf{K}_2, \Pi) \cdot \begin{bmatrix} d\psi \\ d\theta \\ d\phi \\ dX \\ dY \\ dZ \end{bmatrix}$$

with:

$$H_{Id}(\mathbf{K}_1, \mathbf{K}_2) = \begin{bmatrix} \frac{fx_2}{fx_1} \\ 0 \\ ox_2 - fx_2 \cdot rx_1 \\ 0 \\ \frac{fy_2}{fy_1} \\ oy_2 - fy_2 \cdot ry_1 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

The Jacobian matrix $J(\mathbf{X}_1, \mathbf{K}_1, \mathbf{K}_2, \Pi)$ is defined in table 1, and $d_1 = {\mathbf{n}_\pi}^t \mathbf{c_1} - d_0$ represents the orthogonal distance of C_1 from plane Π. Equation 3 may be used in order to infer the projective registration matrix corresponding to a slight variation in the camera pose parameters. Conversely, knowing the observed plane Π, the pose parameters $\mathbf{X}_1$, $\mathbf{K}_1$ of the first camera and the projective registration matrix F_H, one may infer the extrinsic parameters $\mathbf{X}_2$ of the second camera. This result is used by the two steps described below.

3.2. Step 1: Coarse Pose Prediction

The prediction algorithm makes it possible to estimate the current camera pose $\widehat{\mathbf{X}}_n$ using the previous pose $\widehat{\mathbf{X}}_{n-1}$, the current and previous images $\mathbf{I}_n$ and $\mathbf{I}_{n-1}$, as well as the parameters of the dominant plane Π of the observed scene. Let $\mathbf{H}_{n\leftarrow n-1}$ be the homography registering $\mathbf{I}_{n-1}$ with respect to $\mathbf{I}_n$. $\mathbf{H}_{n\leftarrow n-1}$ may be estimated using any registration algorithm (see [8] for a survey of these algorithms). Our approach aims to find the shift in camera pose $\Delta\widehat{\mathbf{X}}_n = \widehat{\mathbf{X}}_n - \widehat{\mathbf{X}}_{n-1}$ which best explains the measured homography $\mathbf{H}_{n\leftarrow n-1}$. This is equivalent to finding a camera pose vector $\widehat{\mathbf{X}_n}$ for which the function $x \mapsto F_H(\widehat{\mathbf{X}}_{n-1}, \mathbf{K}_{n-1}, x, \mathbf{K}_n, \Pi) - \mathbf{H}_{n\leftarrow n-1}$ is equal to zero. This problem is solved using Newton's method for non-linear functions of several variables, leading to Algorithm 1.

This algorithm is very fast, but the accuracy of the estimated pose parameters $\widehat{\mathbf{X}}_n$ is directly related to the accuracy of the previ-

[1] In the context of aerial monitoring, pose parameters are usually available with the current camera devices, but are inaccurate due to sensor inaccuracies and possible synchronization issues.

$$J(\mathbf{X}_1,\mathbf{K}_1,\mathbf{K}_2,\Pi) = \begin{bmatrix} \frac{ox_2+rx_1\cdot fx_2}{fx_1} & -ry_1\cdot\frac{fx_2}{fx_1} & 0 & \frac{fx_2}{fx_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vx_1}}{d_1} & 0 & \frac{ox_2\cdot(\mathbf{n}_\pi\cdot\mathbf{Vx_1})-fx_2\cdot(\mathbf{n}_\pi\cdot\mathbf{Vc_1})}{fx_1\cdot d_1} \\ 0 & -\frac{ox_2}{fy_1} & \frac{fx_2}{fy_1} & \frac{fx_2}{fy_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vy_1}}{d_1} & 0 & \frac{ox_2}{fy_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vy_1}}{d_1} \\ -(1+rx_1^2)\cdot fx_2 & rx_1\cdot ry_1\cdot fx_2 & -ry_1\cdot fx_2 & fx_2\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vc_1}}{d_1} & 0 & rx_1\cdot fx_2\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vc_1}}{d_1} \\ \frac{oy_2}{fx_1} & 0 & -\frac{fy_2}{fx_1} & 0 & \frac{fy_2}{fx_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vx_1}}{d_1} & \frac{oy_2}{fx_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vx_1}}{d_1} \\ rx_1\cdot\frac{fy_2}{fy_1} & -\frac{oy_2+ry_1\cdot fy_2}{fy_1} & 0 & 0 & \frac{fy_2}{fy_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vy_1}}{d_1} & \frac{oy_2\cdot(\mathbf{n}_\pi\cdot\mathbf{Vy_1})-fy_2\cdot(\mathbf{n}_\pi\cdot\mathbf{Vc_1})}{fy_1\cdot d_1} \\ -rx_1\cdot ry_1\cdot fy_2 & (1+ry_1^2)\cdot fy_2 & rx_1\cdot fy_2 & 0 & fy_2\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vc_1}}{d_1} & ry_1\cdot fy_2\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vc_1}}{d_1} \\ \frac{1}{fx_1} & 0 & 0 & 0 & 0 & \frac{1}{fx_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vx_1}}{d_1} \\ 0 & -\frac{1}{fy_1} & 0 & 0 & 0 & \frac{1}{fy_1}\cdot\frac{\mathbf{n}_\pi\cdot\mathbf{Vy_1}}{d_1} \end{bmatrix}$$

Table 1. Expression of the Jacobian matrix used in the camera pose estimation algorithms.

Algorithm 1 Coarse Pose Prediction Algorithm

Inputs: Previous and current images $\mathbf{I}_{n-1}$ and $\mathbf{I}_n$ with calibration parameters $\mathbf{K}_{n-1}$ and $\mathbf{K}_n$, previous estimated pose $\widehat{\mathbf{X}}_{n-1}$, current noisy pose $\widetilde{\mathbf{X}}_n$, and parameters of the plane Π
Output: Current estimated pose $\widehat{\mathbf{X}}_n$

Find $\mathbf{H}_{n\leftarrow n-1}$ registering $\mathbf{I}_{n-1}$ with $\mathbf{I}_n$
$x \leftarrow \widetilde{\mathbf{X}}_n$
Do
  Compute $F_H(\widehat{\mathbf{X}}_{n-1},\mathbf{K}_{n-1},x,\mathbf{K}_n,\Pi)$
  Solve following equation w.r.t dx: $J(\widehat{\mathbf{X}}_{n-1},\mathbf{K}_{n-1},\mathbf{K}_n,\Pi)\cdot dx = \mathbf{H}_{n\leftarrow n-1} - F_H(\widehat{\mathbf{X}}_{n-1},\mathbf{K}_{n-1},x,\mathbf{K}_n,\Pi)$
  $x \leftarrow x + dx$
Until convergence (dx close to zero)
$\widehat{\mathbf{X}}_n \leftarrow x$

Algorithm 2 Fine Pose Correction Algorithm

Inputs: Current image $\mathbf{I}_n$, current (possibly corrupted) pose $\widetilde{\mathbf{X}}_n$, calibration parameters $\mathbf{K}_n$, and 3D model of the scene
Output: Current estimated pose $\widehat{\mathbf{X}}_n$

Compute Π by least square fitting using the 3D model
$x \leftarrow \widetilde{\mathbf{X}}_n$
Do
  Render image $\mathbf{I}_r(x)$ using 3D model
  Compute $\mathbf{H}_{n\leftarrow r}$, registering $\mathbf{I}_r(x)$ with $\mathbf{I}_n$
  Solve following equation w.r.t dx: $J(x,\mathbf{K}_n,\mathbf{K}_n,\Pi)\cdot dx = \mathbf{H}_{n\leftarrow r} - \mathbf{H}_{Id}(\mathbf{K}_n,\mathbf{K}_n)$
  $x \leftarrow x + dx$
Until convergence (dx close to zero)
$\widehat{\mathbf{X}}_n \leftarrow x$

ous estimate $\widehat{\mathbf{X}}_{n-1}$. Assuming the camera pose of the first frame of a video is known, the following camera poses may be estimated, with a possible accumulation of errors. In order to prevent possible drift in pose estimation, we introduce in the following section an extra correction step based on Visual Servoing, which performs alignment with respect to a 3D reference model.

3.3. Step 2: Fine Pose Correction

This step enables the refinement of the current camera pose $\widehat{\mathbf{X}}_n$ based on the knowledge of the current image $\mathbf{I}_n$, the coarse estimate of the current pose $\widetilde{\mathbf{X}}_n$ and a 3D model of the scene. Let $\mathbf{I}_r(x)$ be the image obtained by rendering the 3D model from the camera pose x, and let $\mathbf{H}_{n\leftarrow r}(x)$ be the homography registering $\mathbf{I}_r(x)$ with respect to $\mathbf{I}_n$. Again, this homography may be estimated using any registration algorithm (see [8]).

The goal of this refinement step is to find the camera pose $\widehat{\mathbf{X}}_n$ for which $\mathbf{H}_{n\leftarrow r}(\widehat{\mathbf{X}}_n)$ equals the identity matrix and this translates into finding $\widehat{\mathbf{X}}_n$ which makes the function $x \mapsto \mathbf{H}_{Id}(\mathbf{K}_n,\mathbf{K}_n) - F_H(x,\mathbf{K}_n,\mathbf{X}_n,\mathbf{K}_n,\Pi)$ equal to zero. However, as the pose $\mathbf{X}_n$ is unknown, we replace the analytical homography expression $F_H(x,\mathbf{K}_n,\mathbf{X}_n,\mathbf{K}_n,\Pi)$ by the empirical homography $\mathbf{H}_{n\leftarrow r}(x)$, assuming that the parameters of the plane Π describe the dominant plane in the observed scene. As earlier, this problem may also be solved using Newton's method for non-linear functions of several variables, leading to Algorithm 2.

Notice that Algorithm 2 is slower than the previous one, as it requires rendering the 3D model at each iteration. Notice also that its precision depends on the registration step (between the acquired image and the rendered model) which succeeds only if the initial pose is close to the actual one. In practice, this initial pose is taken from the output of Algorithm 1, which is more precise than the original (corrupted) pose. Finally as shown in section 4, there is no accumulation of errors, as Algorithm 2 uses the 3D reference model at each iteration in order to refine the outputs of Algorithm 1, hence providing more accurate estimates of the pose parameters.

4. EVALUATION

Estimation performances. Previous related work [4] uses Visual Servoing in order to estimate camera poses by employing a 3D model. However, it requires accurate initialization (within 1 degree of the actual pose, and initial mean reprojection error of less than 50 pixels). In contrast, our two-step approach is able to recover from severe perturbations of camera poses (up to 45 degrees of the actual pose and an initial mean reprojection error up to 1000 pixels, in our experiments).

In order to evaluate our algorithm, we used the reference 3D model shown in Fig. 1 obtained from three different aerial videos of 750 frames and ran our algorithm on a fourth video with 350 frames for which ground truth camera poses were known. We generated random camera pose perturbations for each frame of the test video, consisting of severe translations (magnitude up to 50 meters) and rotations (up to 45 degrees around each 3D axis). In our proposed scheme, only the camera pose of the first frame is assumed to be known and poses are estimated for the subsequent frames using our two-step approach.

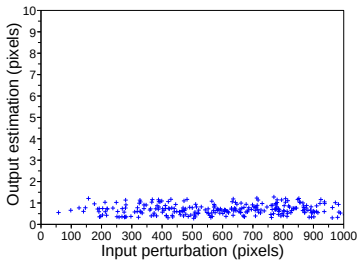

Fig. 2. This plot shows the average reprojection error of control points in image coordinates before (x-axis) and after (y-axis) the refinement of the pose.

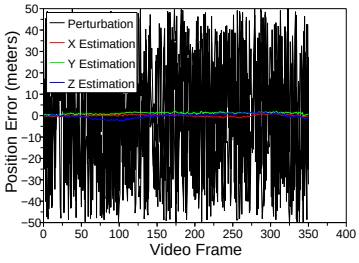

Fig. 3. This graph shows the error on camera position after the execution of our two-step algorithm. Errors are shown in different colors for the X, Y and Z coordinates, and initial perturbations are shown in black.

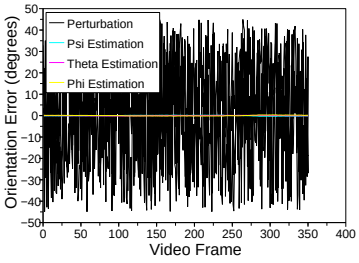

Fig. 4. This graph shows the error on camera orientation after the execution of our two-step algorithm. Errors are shown in different colors for the ψ, θ, and Φ angles, and initial perturbations are shown in black.

Fig. 2 shows the average reprojection errors of control points in image coordinates, before and after refinement of the pose. We clearly see that average reprojection errors after refinement are systematically below 2 pixels, independent from the input reprojection error. Fig. 3 and 4 show the estimation errors related to camera position and orientation. We see that there is no drift, even though the estimation of late frame poses is based on possibly corrupted poses in the early frames. The method is clearly accurate, as it estimates the camera position within 3 meters and its orientation within 0.5 degrees.

Convergence and runtime. When run on a 2.4 Ghz PC, our prediction algorithm takes, on average, 5.6s and 78 iterations before convergence. This processing time is dominated by the registration step, which consists of extracting SURF features [9] from two given images and computing a homography using RANSAC. On the other hand, the correction algorithm requires, on average, 6 iterations before convergence and 13s per iteration. This increased complexity is due to model rendering and image registration, which are performed at each iteration. Fig. 5 shows the reprojection error as a function of algorithm iterations. In the case of the prediction algorithm, reprojection error reaches its minimum in approximately 30 iterations. The refinement algorithm further decreases error slightly, but more importantly it eliminates drift in pose estimation, resulting in a great reduction of the reprojection error for all frames.

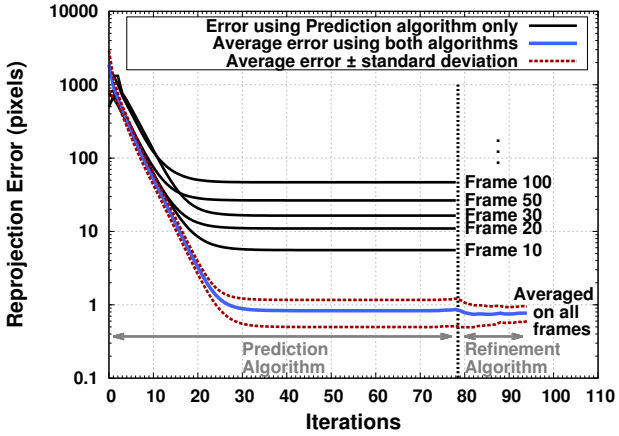

Fig. 5. This plot shows reprojection error as a function of algorithm iterations and emphasize the role of the refinement algorithm. Black curves show the drifting reprojection error for frames 10, 20, 30, 50 and 100, when refinement algorithm is not used, causing estimation error to accumulate. Blue curve show reprojection error averaged on all frames when Prediction and Refinement algorithms are used.

5. REFERENCES

[1] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, 2005.

[2] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[3] T. Pollard and J.L. Mundy, "Change detection in a 3-d world," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[4] D. Crispell, *A Continuous Probabilistic Scene Model for Aerial Imagery*, Phd, Brown University, 2010.

[5] N. Bourdis, D. Marraud, and H. Sahbi, "Constrained optical flow for aerial image change detection," in *2011 IEEE International Geoscience and Remote Sensing Symposium*. 2011, IEEE Computer Society.

[6] R. Kumar, P. Anandan, and K. Hanna, "Shape recovery from multiple views: A parallax based approach," *Proceedings of the International Conference on Pattern Recognition*, 1994, Kumar94icpr.

[7] R. Hartley and A. Zisserman, *Multiple view geometry*, vol. 642, Cambridge university press, 2000.

[8] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Proceedings of the European Conference on Computer Vision*, pp. 404–417, 2006.