

Vision-Based UAV Flight Control and Obstacle Avoidance

Zhihai He, Ram Venkataraman Iyer, and Phillip R. Chandler

Abstract—In this work, we explore various ideas and approaches to deal with the inherent uncertainty and image noise in motion analysis, and develop a low-complexity, accurate and reliable scheme to estimate the motion fields from UAV navigation videos. The motion field information allows us to accurately estimate ego-motion parameters of the UAV and refine (or correct) the motion measurements from other sensors. Based on the motion field information, we also compute the range map for objects in the scene. Once we have accurate knowledge about the vehicle motion and its navigation environment (range map), control and guidance laws can be designed to navigate the UAV between way points and avoid obstacles.

I. INTRODUCTION

In vision-based UAV navigation control [1], the video data captured from the on-board camera has two major purposes: 1) It will be used to determine the ranges of scene objects. Based on the range information, a guidance law can be designed for the UAV such that it is able to avoid obstacles during its navigation between waypoints. 2) The angular velocity estimated from the gyroscope often has a significant amount of noise, especially in the yaw angle or when the wind effect is strong. The vision information, as another source of sensor information, is able to help us refine the estimation of angular velocity for flight control purposes.

We assume that the UAV has a fairly good knowledge (from GPS data) about its linear velocity in the inertial frame. The gyroscope on the UAV is able to give us a rough estimation of the vehicle's orientation. At this moment, we assume that the camera is located at the center of gravity of the UAV, and the camera orientation is the same as the UAV's body orientation, as illustrated in Fig. 1. We map the linear velocity from the inertial frame into the camera coordinate system and denoted it by $V_b = (V_{b1}, V_{b2}, V_{b3})$.

Fig. 2 shows the block diagram of the proposed vision-based motion analysis and obstacle avoidance system. We denote the video frame (also called an image) captured from the camera by $I_t(x, y)$, $0 \leq x \leq W$, $0 \leq y \leq H$, where t is the time index, W and H are the width and height of the frame. Using two consecutive video frames, $I_{t-1}(x, y)$ and $I_t(x, y)$, we are going to determine the angular velocity Ω of the camera and the range map $d_t(x, y)$. Here, $d_t(x, y)$ is the distance between camera and the scene object represented by pixel (x, y) . To determine Ω and $d_t(x, y)$, we need an accurate estimation of motion field

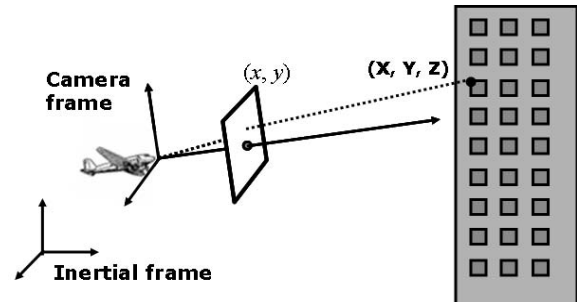


Fig. 1. Coordinate systems in UAV navigation.

$\{M_t(x, y) | 0 \leq x \leq W, 0 \leq y \leq H\}$, where $M_t(x, y)$ represents the motion vector (or velocity vector) of pixel (x, y) . The major challenge in computer-based motion field estimation is uncertainty. To manage the uncertainty and accurately estimate the camera motion, we equally partition the image $I_t(x, y)$ into blocks and classify them into two groups: *structural* (e.g. buildings edges, corners, tree top, etc) and *non-structural* blocks (e.g. texture). The structural blocks will have distinctive features for accurate motion estimation. To obtain a reliable estimation of camera motion, we propose a new motion estimation scheme called reliability-based motion field estimation (rMFE), which will be explained in Section II. Based on the motion information of the structural blocks, we determine the angular motion of camera. Once we have a complete knowledge about the camera motion, we can roughly predict the actual motion of each object in the scene. In other words, for each non-structural block, we can determine a small image region where its motion vector should lie within. Inside this small region, we find the best match for the block and determine its motion vector. Once the motion vector is obtained for each block (both structural and non-structural blocks), we can then compute the range for each block and obtain the range map $d_t(x, y)$. Based on the range map, a guidance law can be designed to control the UAV such that is able to avoid the obstacle and maintain a steady flight between waypoints.

II. MOTION FIELD ANALYSIS

Our objective is to develop a low-complexity and robust motion analysis scheme to accurately estimate the camera motion and range map for UAV navigation. Several approaches have been developed in the literature, including feature tracking and structure from motion [1], [2]. The major problems with the existing methods are: 1) *High computational complexity*. Feature tracking and structure analysis often involve computation-intensive computer vision

Z. He is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211. Email: HeZhi@missouri.edu.

R. V. Iyer is with the Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042.

P. Chandler is with the U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433-7531.

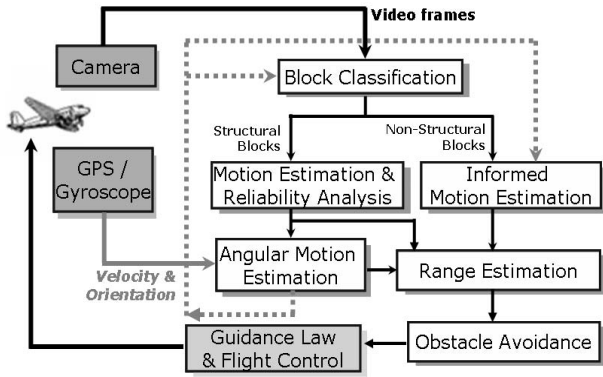


Fig. 2. Block diagram for vision-based ego-motion analysis and obstacle avoidance.

tasks, such as feature extraction and geometric modeling. Computation-intensive motion analysis is not affordable on micro-UAV platforms which have limited on-board computation capability. 2) *Constrained navigation environments*. Many existing algorithms are developed based on assumptions on the image content [1], [6]. This type of algorithms work efficiently in constrained environments with desired image features, but may fail in other navigation environments.

A. Uncertainty in Motion Field Analysis

During UAV navigation, the dominant change in image data from time $t - 1$ to t is caused by camera motion, such as location change, camera rotation, and camera zoom in/out. In addition to this dominant motion, there could be moving objects, such as vehicles or persons on the ground, which cause local changes in the image data. Another major source that cause image content change is noise, including video capture noise, changes in lighting condition, etc. A typical approach to motion field estimation is to partition the image I_t into blocks. For each block \mathbf{B} , we find its best match \mathbf{A} in the previous frame (also called reference frame) I_{t-1} . To find the best match, we need to define a distance (or similarity) metric to measure how close block \mathbf{A} is to \mathbf{B} . This distance metric is denoted by $d(\mathbf{A}, \mathbf{B})$. With this metric, we can then search in the previous frame within a neighborhood of \mathbf{B} to find the best-match block which minimizes the distance metric [4]. In block-based motion analysis, especially for images with noise and little distinctive features, it is sometimes very hard to find out the exact motion based on only local information, even with human eyes. For example, as shown in Fig. 3, the blocks on building edges, regions with flat colors, or texture areas, such as grassland, are able to find a number of “best matches” within its neighborhood [5]. In this case, there is a lot of uncertainty and ambiguity in determining the exact motion of the block. If wrong motion vectors are selected, the camera motion parameters and range map estimation will be inaccurate.

We propose three major ideas to deal with the uncertainty in motion field analysis. 1) We classify the image blocks into two groups: structural and non-structural blocks. The

structural blocks with distinctive features will have reliable motion estimation. 2) We allow the motion estimation to find multiple “best” motion vectors, instead of one single best motion vector for each block as in the conventional motion estimation. We also define a reliability metric to measure how reliable is the motion estimation in each block. 3) We use the reliable motion information from the structural blocks to estimate the camera motion. Accurate knowledge about the camera motion will help us to reduce and uncertainty in motion analysis for non-structural blocks and find the true motion for them.

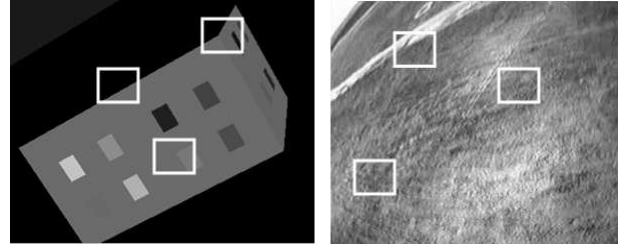


Fig. 3. Sample video frames in UAV navigation.

B. Classification Image Blocks

Structural blocks, such as buildings edges, corners, road lines, and tree tops, have distinctive features for accurate motion estimation. In the frequency domain, a structural block has a major portion of its total energy in the low-to-medium frequency bands. The proposed classification algorithm has three major steps: **Step 1)** We apply discrete cosine transform (DCT) to the block and order the transform coefficients x_i , $0 \leq i \leq S$ from low to high frequencies. **Step 2)** The first 20% of the coefficients, except the DC coefficient, contains a significant amount of structural information, such as edges, corners, and other patterns. For each block, we define an energy ratio for the structural coefficients:

$$r = \frac{\sum_{i=1}^{0.2S} x_i^2}{\sum_{i=1}^S x_i^2}. \quad (1)$$

Step 3) We select the portion of blocks, for example top 15%, which have the highest structural energy ratios, as the structural blocks, and with the rest being classified as non-structural ones. Fig. 4 shows the classification results. The structural blocks are highlighted with white boxes.

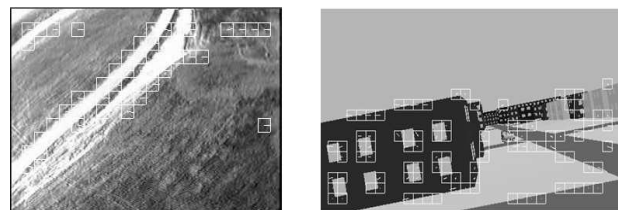


Fig. 4. Classification into structural (highlighted with white boxes) and non-structural blocks.

C. Distance Metric for Motion Search

In motion field analysis for UAV navigation, a desired distance (or similarity) measure should be invariant to camera motion, local object motion, and robust to image noise. To measure the similarity between two blocks \mathbf{A} and \mathbf{B} , we need to take two steps: 1) First, we extract a set of features from each block. 2) Second, we compute the distance between these two sets of features. In conventional motion estimation, the pixel value is often used as the feature, and the distance measure is simply given by

$$d_0(\mathbf{A}, \mathbf{B}) = \sum_{ij} |a_{ij} - b_{ij}|, \quad (2)$$

which is the SAD (sum of absolute difference) measure used in many video compression systems. The SAD metric is invariant only under translational motion.

To handle other types of camera motions, such as rotation, zoom in/out, and perspective changes, we introduce one additional feature, called *intensity profile*. The intensity profile aims to characterize the intensity distribution in an image region. Let $O_B = (x_B, y_B)$ be the center position (pixel) of block \mathbf{B} . Let $C(O_B, r)$ be a circle centered at O_B with a radius r , as illustrated in Fig. 5. The average intensity on this circle is given by

$$m(O_B, r) = \frac{1}{|C(O_B, r)|} \oint_{C(O_B, r)} I_t(x, y) dx dy, \quad (3)$$

where $0 < r \leq R$, and R is the maximum radius to search. For example, we can set R to be the block width. The function $m(O_B, r)$ is called the intensity profile for pixel O_B or block \mathbf{B} . Similarly, we can define the intensity profile for the center pixel of block \mathbf{A}

$$m(O_A, r) = \frac{1}{|C(O_A, r)|} \oint_{C(O_A, r)} I_{t-1}(x, y) dx dy. \quad (4)$$

We can see that if pixel O_A in frame I_{t-1} moves to O_B in frame I_t , the intensity profiles $m(O_B, r)$ and $m(O_A, r)$ will be the same even with camera rotation. However, with camera zoom, $m(O_B, r)$ and $m(O_A, r)$ will be different. For example, if the camera zooms out, $m(O_A, r)$ will match the first segment of $m(O_B, r)$ after being scaled horizontally (either compressed or stretched), as illustrated in Fig. 5. Based on this observation, we can define another distance measure as follows

$$d_1(\mathbf{A}, \mathbf{B}) = \min_{1-\delta \leq \lambda \leq 1+\delta} \max_{0 \leq r \leq R/\lambda} |m(O_A, \lambda \cdot r) - m(O_B, r)|, \quad (5)$$

where λ is the scaling factor, and $[1 - \delta, 1 + \delta]$ is the search range for λ . It can be seen that the distance (or similarity) metric $d_1(\mathbf{A}, \mathbf{B})$ is invariant under camera rotation and zoom. The distance metrics $d_0(\mathbf{A}, \mathbf{B})$ and $d_1(\mathbf{A}, \mathbf{B})$ captures different information about the similarity between blocks. We form a comprehensive distance metric for motion search as follows

$$d(\mathbf{A}, \mathbf{B}) = w \cdot d_0(\mathbf{A}, \mathbf{B}) + (1 - w) \cdot d_1(\mathbf{A}, \mathbf{B}), \quad (6)$$

where w is the weighting factor which can be adjusted according to the amount of camera motion in rotation and zoom. For example, if the angular velocity of the camera is relatively small, we can choose a smaller value of w . Once the distance metric is established, we can then search the neighborhood of block \mathbf{B} in the previous frame I_{t-1} , denoted by $\mathcal{N}(\mathbf{B})$, to find the best block \mathbf{A}^* which has the minimum distance to \mathbf{B} ,

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathcal{N}(\mathbf{B})} d(\mathbf{A}, \mathbf{B}) \quad (7)$$

The difference vector between the center positions of blocks \mathbf{A}^* and \mathbf{B} is the motion vector.

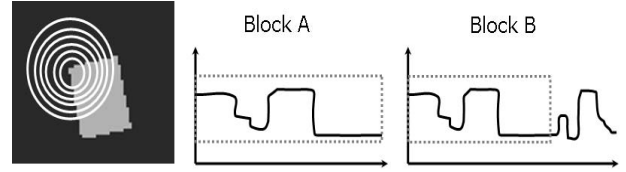


Fig. 5. Definition of intensity profile for blocks.

D. Reliability-Based Motion Field Estimation

As discussed in Section I, the image is equally partitioned into blocks, $\{\mathbf{B}^n | 1 \leq n \leq N\}$. Based on the frequency-domain information, we classify these blocks into structural and non-structural blocks, as discussed in Section II-B. We denote the structural blocks by $\{\mathbf{B}^m | 1 \leq m \leq M\}$, $M \leq N$. As discussed in Section II-A, because of the inherent uncertainty and ambiguity in motion analysis, each block may be able to find multiple “best” matches according the distance metric in (7). In addition, because of image noise, the true motion vector may even not have the minimum distance. To deal with this problem, we propose a reliability-based motion field analysis scheme as explained in the following.

For each structural block \mathbf{B}^m , we find the top L best matches for \mathbf{B}^m in the previous frame I_{t-1} and the estimation results are denoted by $\Lambda = \{(\mathbf{V}_j^m, d_j^m) | 1 \leq j \leq L\}$, where $\mathbf{V}_j^m = (\hat{x}_j^m, \hat{y}_j^m)$ represents the motion vector and d_j^m is the corresponding distance. Based on the data set $\{(\mathbf{V}_j^m, d_j^m) | 1 \leq j \leq L\}$, we extract a representative motion vector, denoted by \mathbf{V}^m , and define a reliability measure γ^m . We assume \mathbf{V}^m is the “true” motion vector, which could be wrong. Our basic idea is as follows: if we choose \mathbf{V}^m as the representative motion vector, there will be a number of other motion vectors in the set whose distance measurements are also very close to the one of \mathbf{V}^m . Certainly, the larger the number is, the more uncertainty we have, and the less reliability the motion estimation is. Let

$$d_-^m = \min_j d_j^m, \quad d_+^m = \max_j d_j^m. \quad (8)$$

Let

$$d_0^m = d_-^m + \alpha \cdot (d_+^m - d_-^m), \quad (9)$$

where α is a threshold value between 0 and 1. By default, we set $\alpha = 0.1$. The physical meaning of $\alpha = 0.1$ is noise level.

We pick out a subset of those motion vectors in Λ whose distance measurements are very close to the minimum d_-^m , and denote this subset by

$$\Lambda_- = \{(\mathbf{V}_k^m, d_k^m) | d_k^m \leq d_0^m\}. \quad (10)$$

Here, we re-label the elements in the set Λ_- by index k , $1 \leq k \leq K_m < L$. We choose the mean of those motion vectors as their representative

$$\mathbf{V}^m = \frac{1}{K_m} \sum_{k=1}^{K_m} \mathbf{V}_k^m. \quad (11)$$

We define the reliability measure as

$$\gamma^m = \frac{1}{1 + \sum_{k=1}^{K_m} \|\mathbf{V}_k^m - \mathbf{V}^m\|_2}. \quad (12)$$

Here, $0 < \gamma^m \leq 1$. If a motion search is reliable, either the value of K_m will be small (close to one which implies a single minimum) or the motion vectors \mathbf{V}_k^m will be very close to each other. In this case, the corresponding reliability measure γ^m will be very close to 1.

In camera motion parameter estimation as discussed in the next section, the reliability measure γ^m will act as a weighting factor. Those motion vectors with lower reliability, i.e., higher uncertainty, will have less influence when determining the camera motion.

III. CAMERA MOTION PARAMETER ESTIMATION

From the motion field analysis, we have obtained a representative motion vector \mathbf{V}^m (assumed to be the true motion) and an associated reliability measure γ^m for each structural block \mathbf{B}^m , $1 \leq m \leq M$. (An example value of M is 60.) Based on this data set, we are going to estimate the camera motion parameters.

Let (X, Y, Z) be the coordinate of the object. Let (x, y) be the pixel coordinate in the image, as illustrated in Fig. 1. Based on camera view geometry analysis [7], we have

$$\begin{aligned} \dot{x} &= \frac{V_{b1}}{Z} \left(x + \frac{V_{b3}}{V_{b1}} \right) \\ &+ \Omega_X xy - \Omega_Y (1 + x^2) + \Omega_Z y \end{aligned} \quad (13)$$

$$\begin{aligned} \dot{y} &= \frac{V_{b1}}{Z} \left(y - \frac{V_{b2}}{V_{b1}} \right) \\ &+ \Omega_X (1 + y^2) - \Omega_Y xy - \Omega_Z x \end{aligned} \quad (14)$$

where $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ is the angular velocity; (\dot{x}, \dot{y}) is the motion vector of the object which is obtained from the motion field analysis. The unknown variables are $(\Omega_X, \Omega_Y, \Omega_Z)$ and the range Z for each pixel (x, y) .

A. Estimate the Angular Velocity

The angular velocity, denoted by $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ plays an important role in UAV navigation and obstacle avoidance. From other sensors, such as gyroscope, we can get a rough estimation of body orientation, and the angular velocity $\boldsymbol{\Omega}$ can be roughly estimated by taking the difference between body orientation measurements at two time

instances. Vision information, as another important source of information, is able to help us refine the estimation of the angular velocity $\boldsymbol{\Omega}$. Combining Eqs. (13) and (14) and getting rid of the pixel-dependent variable Z , we have

$$\begin{aligned} D_1(x, y)\dot{x} &- D_2(x, y)\dot{y} = C_1(x, y)\Omega_X \\ &+ C_1(x, y)\Omega_Y + C_3(x, y)\Omega_Z, \end{aligned} \quad (15)$$

where

$$\begin{aligned} D_1(x, y) &= \left(y - \frac{V_{b2}}{V_{b1}} \right), \\ D_2(x, y) &= \left(x + \frac{V_{b3}}{V_{b1}} \right), \\ C_1(x, y) &= xy \left(y - \frac{V_{b2}}{V_{b1}} \right) - (1 + y^2) \left(x + \frac{V_{b3}}{V_{b1}} \right), \\ C_2(x, y) &= -(1 + x^2) \left(y - \frac{V_{b2}}{V_{b1}} \right) + xy \left(x + \frac{V_{b3}}{V_{b1}} \right), \\ C_3(x, y) &= y \left(y - \frac{V_{b2}}{V_{b1}} \right) + x \left(x + \frac{V_{b3}}{V_{b1}} \right). \end{aligned} \quad (16)$$

The unknown variables $(\Omega_X, \Omega_Y, \Omega_Z)$ can be obtained with Least Mean Square Error (LMSE) fitting weighted by the reliability. Let (x^m, y^m) be the pixel coordinate of the center of block \mathbf{B}^m . From the motion field analysis, we have obtained the motion vector (\dot{x}^m, \dot{y}^m) for this pixel and the associated reliability measure γ^m . The weight LMS can be written into a matrix form as

$$\boldsymbol{\Gamma} \mathbf{A} \boldsymbol{\Omega} = \boldsymbol{\Gamma} \mathbf{b}, \quad (17)$$

where

$$\mathbf{A} = \begin{bmatrix} C_1(x^1, y^1) & C_2(x^1, y^1) & C_3(x^1, y^1) \\ C_1(x^2, y^2) & C_2(x^2, y^2) & C_3(x^2, y^2) \\ \vdots & \vdots & \vdots \\ C_1(x^M, y^M) & C_2(x^M, y^M) & C_3(x^M, y^M) \end{bmatrix} \quad (18)$$

and

$$\mathbf{b} = \begin{bmatrix} D_1(x^1, y^1)\dot{x}^1 + D_2(x^1, y^1)\dot{y}^1 \\ D_1(x^2, y^2)\dot{x}^2 + D_2(x^2, y^2)\dot{y}^2 \\ \vdots \\ D_1(x^M, y^M)\dot{x}^M + D_2(x^M, y^M)\dot{y}^M \end{bmatrix}, \quad (19)$$

and

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma^1 & & & \\ & \gamma^2 & & \\ & & \dots & \\ & & & \gamma^M \end{bmatrix}. \quad (20)$$

The solution is given by

$$\boldsymbol{\Omega} = [(\boldsymbol{\Gamma} \mathbf{A})^t (\boldsymbol{\Gamma} \mathbf{A})]^{-1} (\boldsymbol{\Gamma} \mathbf{A})^t (\boldsymbol{\Gamma} \mathbf{b}). \quad (21)$$

B. Range Estimation

Once the camera motion is known, we are able to determine the range (or depth) Z for each block in the scene. As discussed in Section II, the image is partitioned into blocks, \mathbf{B}^n , $1 \leq n \leq N$. Let $\Lambda = \{(\dot{x}_j^n, \dot{y}_j^n) | 1 \leq j \leq L\}$ be the top candidate motion vectors. If the block, which corresponds to

an object in the scene, is stationary, the true motion vector must satisfy Eqs. (13) and (14). Denote the right-hand sides of (13) and (14) by

$$f(x, y, Z) = \frac{V_{b1}}{Z}(x + \frac{V_{b3}}{V_{b1}}) + \Omega_X xy - \Omega_Y(1 + x^2) + \Omega_Z y, \quad (22)$$

$$g(x, y, Z) = \frac{V_{b1}}{Z}(y - \frac{V_{b2}}{V_{b1}}) + \Omega_X(1 + y^2) - \Omega_Y xy - \Omega_Z x. \quad (23)$$

If the $(\hat{x}_j^n, \hat{y}_j^n)$ is the true motion, then the range of this block can be determined by least mean squared error estimation

$$Z^j = \arg \min_Z [(\hat{x}_j^n - f(x, y, Z))^2 + (\hat{y}_j^n - g(x, y, Z))^2]. \quad (24)$$

The corresponding fitting error is denoted by

$$E^j = [\hat{x}_j^n - f(x, y, Z^j)]^2 + [\hat{y}_j^n - g(x, y, Z^j)]^2. \quad (25)$$

Certainly, the true motion must have the minimum fitting error. Let

$$j^* = \arg \min_j E^j. \quad (26)$$

The range of the block is given by Z^{j^*} , and the associated motion vector is $(\hat{x}_{j^*}^n, \hat{y}_{j^*}^n)$.

IV. EXPERIMENTATION AND PERFORMANCE ANALYSIS

The proposed motion field estimation, reliability analysis, camera motion estimation, and range map computation have been implemented with C code. We take two major steps to evaluate this vision analysis system. First, we are going to test the system using the multi-UAV simulator. The simulator is able to generate a video sequence for the UAV camera view and the associated camera coordinate system and orientation. In addition, since the simulator knows which point object in the world scene is mapped to which pixel in a video frame, we have the ground truth for the motion fields, as well as the range map.

In simulator-based performance evaluation, the inputs to the vision analysis system, are the video frames, the linear velocity, as well as the camera orientation. We add some level of noise to the camera orientation to simulate the sensor measurement noise. The output of the system will be the angular velocity estimation and range map. We are going to check if the vision information is able to reduce the noise in angular velocity and camera orientation measurement. We also compare the estimated range map against the ground truth. In the second stage of evaluation, we will use the flight test data. The flight data has video from the UAV camera, its GPS location, and orientation (with noise). It also has the GPS location of the targets. The vision analysis system will estimate the range of the targets, and we will compare the estimate against the actual measurement. In addition to the performance evaluation, we will also analyze the impact of noise in linear velocity on the estimation accuracy.

V. CONCLUDING REMARKS

We propose a hierarchical framework to deal with uncertainty and noise in motion field analysis, so as to develop a low-complexity and reliable vision analysis system for UAV navigation. First, we classify the image data into structural and non-structural, and only use the reliable motion information from structural blocks for camera motion estimation. Second, we introduce reliability analysis into motion field estimation and let those motion vectors with higher reliability play an influential role in camera motion estimation. In this way, even if the local motion estimation could be wrong inside some image regions, the overall camera motion estimation is still accurate and robust due to those highly reliable structural blocks. Third, we use the accurate estimation of camera motion to constrain the motion search for non-structural blocks, and this reduces the uncertainty, as well as computational complexity significantly.

REFERENCES

- [1] T. Kanade, O. Amidi, and Q. Ke, "Real-Time and 3D Vision for Autonomous Small and Micro Air Vehicles," *43rd IEEE Conference on Decision and Control (CDC 2004)*, December, 2004.
- [2] M. Meingast, C. Geyer, and S. Sastry, "Vision based terrain recovery for landing unmanned aerial vehicles," *Proceedings of the Conference on Decision and Control*, December, 2005, Bahamas.
- [3] Horn, B. K. and Schunck, B. G., Determining Optical Flow, *Artificial Intelligence*, Vol. 17, 1981, pp. pp. 185203.
- [4] F. Dufaux, F. and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," *Proceedings of the IEEE*, Volume 83, Issue 6, June 1995 Page(s):858 - 876.
- [5] Y. Huang, K. Palaniappan, X. Zhuang, and J.E. Cavanaugh, "Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 17, Issue 12, Dec. 1995 Page(s):1177 - 1190.
- [6] Gui-Qiu Bao, Shen-Shu Xiong, and Zhao-Ying Zhou "Vision-based horizon extraction for micro air vehicle flight control," *IEEE Transactions on Instrumentation and Measurement*, Volume 54, Issue 3, June 2005 Page(s):1067 - 1072.
- [7] Richard Hartley, Andrew Zisserman *Multiple View Geometry in Computer Vision*, Cambridge University Press; 2 edition (March 25, 2004).