Vision-based Detection and Pose Estimation for Formation of Micro Aerial Vehicles

Mengmi Zhang*, Feng LIN†, Ben M. Chen*
*Department of Electrical and Computer Engineering
National University of Singapore
117576 Singapore
Email:{a0091624, bmchen}@nus.edu.sg
†Temasek Laboratories
National University of Singapore
117411 Singapore
Email:{tsllinf}@nus.edu.sg

Abstract—This paper proposes an innovative method to detect micro aerial vehicles (MAVs) and estimate their relative pose in formation using a monocular on-board camera. Haar classifier is trained for autonomously detecting MAV in open scenes, like grasslands or obstruct-free playgrounds. In order to increase the robustness of the detection, a Kalman filter has been employed to conduct image tracking. Contours of detected MAV have been extracted for shape matching. Contours match with the given point sets using Hungarian algorithm and relaxation labeling based on shape contexts. Two techniques, affine transformation and thin plate spline (TPS) transformation, are explored, while TPS is better in dealing with distorted shapes. In experiments, we develop and implement an innovative 2D shape-based pose estimation method by using only one monocular camera which results in fast and accurate performances.

I. INTRODUCTION

MAV swarm has been widely studied in potential applications, such as surveillance, navigation, and transportation. Especially, in the extreme situation, swarm formation is required without inter-vehicle communication. In such a situation, vision sensing can be used as important feedback information to measure the relative positions between the members of a formation group.

Although many different algorithms exist to perform object detection, each has its own weakness and strengths. Faced with challenges such as its large computational costs, relatively long processing time and limited on-board processing power, Viola et al. have introduced an object detection framework based on boosted cascade of features which provide competitive object detection rates in real time [12]. In our paper, we employ the Haar-like feature based target detection method in Section II as an extension of their methods. To the best of our knowledge, current two-frame tracking can be accomplished using optical flow technique such as contour-based moving object tracking proposed in [8], and changing based object tracking method, e.g. color-based particle filter is illustrated in [9].

Vision tracking algorithm by Kalman Filter is illustrated in details in Section III. It helps accelerate the processing speed and improve target detection accuracy. Within the regions of interest, a view-based method for recognizing 3D objects from 2D can be employed on conditions that the object has to be rigid. In other words, the notion of shape similarity between views has to be distinct. One of the most efficient and fastest metrics for 2D shape matching is based on curve matching. Point sets can be extracted from contours of shapes. Unknown shapes can be iteratively transformed using transformation matrix estimated from the current point correspondence. The most optimal solution is the one with most number of points matched with the given point sets. Pose estimation can then be determined by comparing the unknown shape with prototypes set up before. We adopted the general procedure on shape matching from [11] and [13] with some related concepts and algorithms introduced in Section IV. We evaluate both methods and extend their working principles in Section V. In the end, we provide experimental results and discussions in Section VI and give conclusions in Section VII.

II. TARGET DETECTION

There were a lot of techniques published for the target detection such as Haar wavelet based AdaBoost cascade, Histogram of oriented gradient features and others [3], [2]. Among them, the first method is more applicable to our scenario and real-time processing. The principle idea for the target detection is to combine a set of the Haar wavelet features to separate a specific target from background. To realize real-time operation, the set of the features have been formed in a cascade manner to reduce the computation cost. The Haar wavelet features are employed here due to their rich image representation and very fast computation by using the integral representation and very fast computation by using the integral image [12], which is suitable for the onboard processing.

Normally, more than one hundred of thousands of such features can be extracted from a small image sub-window (e.g. 24 × 24 pixels) by varying the scale and location of the feature. Each Haar wavelet feature can be used to construct a weak classifier to identify the target.

\[
h_j(x) = \begin{cases} 
1 & \text{if } p_j f_j(x) < p_j \theta_j \\
0 & \text{otherwise}
\end{cases}
\]

where \( x \) is a sampled sub-window in an image. \( \theta_j \) is a
threshold to achieve minimal misclassification. \( p_j \in \{-1, 1\} \) is used to define the direction of inequality.

A single weak classifier cannot perform very well. Fortunately, as proved in [10], it is theoretically possible to combine multiple weak classifiers to be a better classifier, namely strong classifier. In [4], the AdaBoost algorithm is proposed to form a strong classifier by selecting a small set of weak classifiers and calculating their corresponding weight based on training samples. The detailed procedure and statistical behavior of the final strong classifier can be found in [4], [14]. The final strong classifier \( H(x) \) is formed as a linear combination of selected \( n \) weak classifiers.

\[
H(x) = \sum_{j=1}^{n} \alpha_j h_k(x),
\]

where \( \alpha_j \) is a weight to minimize the exponential loss of the classifier. If \( H(x) \geq \eta_t, \eta_t = \frac{1}{2} \sum_{j=1}^{n} \alpha_j \), the target is identified. The strong classifier will have better performance than each weak classifier.

Additionally, we can sort the weak classifiers according to the weight \( \alpha_j \) and compute the strong classifier iteratively.

\[
H_m(x) = H_{m-1}(x) + \alpha_m h_m(x), \quad \alpha_m < \alpha_{m-1},
\]

where \( H_{m-1}(x) = \sum_{j=1}^{m-1} \alpha_j h_j(x) \), and \( H_0 = 0 \). In each iteration, if \( H_m(x) + \sum_{j=m+1}^{n} \alpha_j < \eta_t \), we can conclude that the target is not in the sampled sub-window, and the rest weak classifiers \( (H_j, m < j \leq n) \) do not need to be checked in the sampled sub-window. Thus, the computational cost can be reduced significantly and this is the principle idea of the cascade of the classifiers. This structure is suited to the scenario that there is an overwhelming majority of negative sub-windows in an image and the negative sub-windows have less pattern than the target that can be identified by using less features.

### III. IMAGE TRACKING

The target detection methods mentioned above are normally used to initialize image tracking in many vision applications. An effective combination of detection and motion estimator is able to speed up the processing, as well as deal with moving targets and uncertainties of outdoor environments, such as variations in lighting, background, etc. Image tracking typically involves mathematical tools such as the Kalman filter, Bayesian network [6], [5], [15]. This image tracking method can be referred to as Filtering and Data Association approach, considered as a top-down process.

To predict the target location in the image, a motion model is required. It is well known that the motion of a point mass in the two-dimensional plane can be defined by its two-dimensional position and velocity vector. Let \( x = [x, \dot{x}, y, \dot{y}]^T \) be the state vector of the centroid of the tracked target in the Cartesian coordinate system. Non-manoeuvring motion of the target is defined by it having zero acceleration: \( [\ddot{x}, \ddot{y}]^T = [0, 0]^T \). Strictly speaking, the motion of the intended targets may be manoeuvring with unknown inputs. Nevertheless, we assume the standard 4-th order non-manoeuvring motion model by setting the acceleration as \( [\ddot{x}, \ddot{y}]^T = w(t) \), where \( w(t) \) is a white noise process [7], which should be sufficient for the target tracking in the image.

In addition, it is necessary to consider the motion of the pan/tilt camera, which can be compensated using its steady state value, since it normally has higher bandwidth than the dynamics of the UAV. The discrete-time model of the target motion can be expressed as

\[
\begin{align*}
\{ x(k+1) & = \Phi x(k) + F_{\text{cam}} \Delta u_{\text{cam}} + \Delta w(k), \\
\{ z(k)   & = H x(k) + u(k),
\end{align*}
\]

where

\[
\Phi = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} T_s^2 & 0 \\ T_s & 0 \\ 0 & T_s^2 \\ 0 & T_s \end{bmatrix},
\]

\[
H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},
\]

\[
u_{\text{cam}} = \begin{bmatrix} \Delta \phi_h \\ \Delta \theta_h \end{bmatrix}, \quad F_{\text{cam}} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix},
\]

\( T_s \) is the sampling period of the vision software. A Kalman filter can then be designed based on the above motion model to estimate the states of the target in the image plane. Based on the motion model, we can predict the possible location of the target in the next frame, and also define the region of interest \( N_s(k) \) as a neighbourhood of the predicted location of the target in the image \( I(x, y) \), which is given below.

\[
N_s(k) = \{ I(x, y) \mid x \in [x_1, x_2], \ y \in [y_1, y_2] \},
\]

where

\[
\begin{align*}
x_1 &= \dot{x}(k|k-1) - 0.5 \times (w_x(k|k-1) + \varepsilon \sigma_x(k|k-1)), \\
x_2 &= \dot{x}(k|k-1) + 0.5 \times (w_x(k|k-1) + \varepsilon \sigma_x(k|k-1)), \\
y_1 &= \dot{y}(k|k-1) - 0.5 \times (h_y(k|k-1) + \varepsilon \sigma_y(k|k-1)), \\
y_2 &= \dot{y}(k|k-1) + 0.5 \times (h_y(k|k-1) + \varepsilon \sigma_y(k|k-1)), \end{align*}
\]

\( w_x, h_y = w_x, h_y \) are the width and height of the target detected in the previous frame. \( \sigma_x, \sigma_y \) are the standard deviation of the estimation of \( x \) and \( y \), given by the Kalman filter. \( \varepsilon \) is a coefficient. We can also reduce the probability of the false alarm by using the motion estimator.

### IV. SHAPE MATCHING

Serge B. et al proposed a fast contour-based shape matching technique in [11]. Yefeng Z. [13] et al further developed the algorithm to increase robustness with the high computation cost. In this paper, we explored both methods and made adjustments from there for the sake of our demanding requirements on accuracy and time efficiency.
A. Shape Context

Canny edge detection and morphology operation are implemented to extract MAV contour from the region of interest. Contours are then randomly sampled with equal number of points as those on data samples. In order to find the best matching point on the second shape, one has to give the best description on relative position including distance and orientation with reference to other points. A rich local descriptor called shape context was employed. Sets of vectors are derived from one local point with respect to other points on the contours. They express the configuration of the entire shape as the cardinality of vector sets increases. Histogram is assigned to each local point defined as shape context which has to be done before assigning points into each bin. Let $C(p_i, q_j)$ denote the cost of matching these two points, then it can be computed as follows:

$$C_{ij} = C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$  \hspace{1cm} (5)

Not all points will be counted in computing histograms since some of the points will be out of range in terms of distance. Those points that are extremely far apart from local points will be ignored after normalization. They will not fall inside any of the five distance bins.

In order to make it more robust and compact, normalization has to be done before assigning points into each bin. Let $C(p_i, q_j)$ denote the cost of matching these two points, then it can be computed as follows:

$$C_{ij} = C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$  \hspace{1cm} (5)

There are three basic shape transformations i.e. translation, scaling and rotation. Translation and scaling can be maintained easily in log-polar space; however, rotation has to be considered separately instead of simply calculating angles with respect to $+x$ axis by taking $\arctan$ from two point coordinates. Two methods are proposed here:

- Find mass center from extracted contour points and take the direction from local point to mass center as the reference direction.
- Calculate tangent angle at local points and take that as the reference angle.

B. Hungarian Algorithm

Given the set of costs $C_{ij}$ between all pairs of points $p_i$ on the first shape and $q_j$ on the second shape, we want to minimize the total cost of matching.

$$H(\pi) = \sum_{i,j} C(p_i, q_{\pi(i)})$$  \hspace{1cm} (6)

where $\pi$ means one-to-one matching. Since this is an instance of the bipartite graph matching problem, we can use Hungarian algorithm to minimize the cost. Time complexity for this algorithm is $O(n^3)$. The matrix has to be square matrix so extra dummy points can be added into matrix until equal dimensions have been satisfied. In our case, we resample the point sets from two shapes until we get equal number of points from two sets. In order to avoid effects from possible outliers, we assume 25% of sample points are outliers out of resampled equal number of points. Thus extra 25% dummy points have been added into cost matrix with dummy cost to be 0.2. Now the cost matrix becomes as follows:

$$\begin{bmatrix}
C_{00} & C_{01} & \cdots & \text{dummy} \\
C_{10} & C_{11} & \cdots & \text{dummy} \\
\vdots & \vdots & \ddots & \text{dummy} \\
\text{dummy} & \text{dummy} & \cdots & \text{dummy} \\
\text{dummy} & \text{dummy} & \cdots & \text{dummy}
\end{bmatrix}$$  \hspace{1cm} (7)

where original cost matrix is $N$ by $N$. Now 25% dummy points have been added in the cost matrix with dimensions $(N + M)$ by $(N + M)$ where $M$ is the number of possible outliers we assume to be.

C. Relaxation Labeling

From shape context, we can see that the computed cost is negatively related to the chance of similarity between these two points. Thus, we can model this matching problem into a probability problem. Relaxation labeling is one of the probabilistic models dealing with matching problems. One of the drawbacks for simple Hungarian algorithm is that it may lead to wrong matching where two points are far apart but have similar costs. Points on the contours are closely related to its neighbors. In order to preserve the rough structure of shapes, points and its neighbors cannot change freely. If their neighbors have higher probability of matching to the correct result, so does that local point. Neighbors are defined by the length of edges among points. The one with the shorter edge length is counted as neighbors. In our case, we set a local point to have maximum of 30 neighbors. It is possible that those points having fewer neighbors are less likely to find correspondence and more likely to be outliers. Mathematical models can be formulated as follows [13]:

$$\begin{bmatrix}
P_{11} & \cdots & P_{1N} & P_{1,nil} \\
\vdots & \ddots & \vdots & \vdots \\
P_{M1} & \cdots & P_{MN} & P_{M,nil} \\
P_{nil,1} & \cdots & P_{nil,N} & 0
\end{bmatrix}$$  \hspace{1cm} (8)

If point $T_m$ from shape T is matched to point $D_n$ in the deformed shape D, then $P_{mn} = 1$; otherwise $P_{mn} = 0$. In the meanwhile, it has to follow the basic probability principle that the sum of probability from all possible events has to be one.

$$\sum_{n=1}^{M} P_{mn} = 1, \hspace{1cm} m = 1, 2, \ldots, M$$  \hspace{1cm} (9)

$$\sum_{m=1}^{N} P_{mn} = 1, \hspace{1cm} n = 1, 2, \ldots, N$$  \hspace{1cm} (10)

$$P_{mn} \propto e^{-C_{mn}/T_{init}}$$  \hspace{1cm} (11)
The exponential distribution is used to model the probability of the match where \( T_{\text{init}} = 0.1 \) by default to adjust the reliability of the initial probability measures. Thus, the updating rule for each iteration to achieve a global consistent result is

\[
P_{mn} := \frac{P_{mn} S_{mn}}{\sum_{j=1}^{N} P_{mj} S_{mj}},
\]

where \( S_{mn} \) is a supporting function defined as below:

\[
S_{mn} = 4 \sum_{i \in N_m} \sum_{j \in N_n} P_{ij}
\]

\( N_m \) is a set of neighbors of point \( m \) and similarly, \( N_n \) is a set of neighbors of point \( n \). After several iterations of updating and normalization, it shows that the updating process will converge to a local optimal solution.

### D. Affine Transformation

After finding the correct correspondence between point sets \( P \) and \( Q \), we are going to determine an affine transformation such that the image of \( Q \) approaches \( P \) as best as possible in terms of least squares. Let two sets of points \( P \) and \( Q \) be given by \( p_i = (p_{1i}, p_{2i}, ... p_{ni})^T \), \( q_i = (q_{1i}, q_{2i}, ... q_{ni})^T \), and \( i \) refers to total number of points. We are looking for affine transformation matrix \( A \) and \( t \) such that \( S(A,t) \) is minimized

\[
S_{A,t} = \sum_{k=1}^{m} \| p_{ki} - A q_{ki} - t \|^2
\]

It can be proven that matrix \( A \) and \( t \) can be derived as follows:

\[
P = \begin{bmatrix}
\sum_{i=1}^{m} q_x & \sum_{i=1}^{m} q_x q_y & \sum_{i=1}^{m} q_y \\
\sum_{i=1}^{m} q_y q_x & \sum_{i=1}^{m} q_x q_y & \sum_{i=1}^{m} q_x \\
\end{bmatrix},
\]

\[
Q = \begin{bmatrix}
\sum_{i=1}^{m} p_x & \sum_{i=1}^{m} p_x q_y & \sum_{i=1}^{m} p_y \\
\sum_{i=1}^{m} p_y q_x & \sum_{i=1}^{m} p_x q_y & \sum_{i=1}^{m} p_x \\
\end{bmatrix}
\]

\[
P^{-1} \times Q = \begin{bmatrix}
t_{00} & t_{10} \\
t_{01} & t_{11}
\end{bmatrix},
\]

where

\[
A = \begin{bmatrix}
a_{00} & a_{01} \\
a_{10} & a_{11}
\end{bmatrix},
\]

\( t = \begin{bmatrix} t_{00} \\ t_{10} \end{bmatrix}, \)

\( m \) is the number of correspondences.

Sometimes due to large computation costs, approximation by triangulation method can be employed. A large triangle which is extracted from model points can be used to find estimated matrix \( A \) and translation matrix \( t \) as long as the calculated affine residual is within tolerable range. Affine residual can be calculated by finding the median Euclidean distance between the transformed model point sets and the deformed point sets.

### E. TPS Transformation

Apart from the basic affine transformation illustrated above, we can use more advanced transformation methods which can deal with complex and deformed shapes. The aim is to find TPS interpolate \( f(x,y) \) to minimize the bending energy:

\[
I_f = \int \int_{\mathbb{R}^2} \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 dxdy
\]

The objective function has the following form:

\[
f(x,y) = a_1 + a_2 x + a_3 y + \sum_{i=1}^{n} w_i U(||(x_i, y_i) - (x, y)||)
\]

where the kernel function \( U(r) = r^2 \log r^2 \) and \( U(0) = 0 \) by default. It can be proved that in order to find \( a_1, a_2, a_3 \) and \( w_i \) for all correspondences, we can follow steps below:

\[
\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}
\]

\[
A = \begin{bmatrix} w \\ a \end{bmatrix} = L^{-1} C,
\]

where \( P \) contains the homogeneous coordinates of shape \( P \) to be transformed while \( v_i = f(x_i, y_i) \) refers to the point coordinates of the deformed shape. Entry in matrix \( K \) follows \( K_{ij} = U(||(x_i, y_i) - (x_j, y_j)||) \). Matrix \( A \) can be obtained by computing the inverse of the first matrix and post-multiplying by \( C \).

Thus, by knowing all the coefficients, we can obtain objective function \( f(x, y) \). When there is noise in the specified value \( v_i \), we can relax the exact interpolation requirement by means of regularization. This is accomplished by minimizing

\[
H[f] = \sum_{i=1}^{n} (v_i - f(x_i, y_i))^2 + \lambda_f
\]

The regulation parameter \( \lambda \) controls the amount of smoothing. In the calculation of TPS transformation matrix, we can replace the matrix \( K \) by \( K + \lambda I \). If \( (x_i, y_i) \) has been scaled to \( (a x_i, a y_i) \), the result will remain the same if the regulation parameter be \( a^2 \beta \).

### F. Shape Context Distance

After a few iterations, one metrics has to be decided to judge the similarity between warped shapes and prototypes. Here we adopt shape context distance definition as follows:

\[
D_{sc}(P, Q) = \frac{1}{n} \sum_{p \in P} \arg_{q \in Q} \min C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} \arg_{p \in P} \min C(p, T(q))
\]
V. Pose Estimation

The viewing sphere Fig. 1 is endowed with unique aspect views, in other words, the object is rigid in the sense that there are distinctions for each pair of views [1].

The viewing sphere is sampled at regular intervals. For testing purposes, we collected 29 samples in Fig 2 from different viewing orientations including roll, pitch and yaw angles relative to camera positions recorded in a look-up table. We used a motion capture system, VICON system, to accurately measure the relative orientation angles. From shape matching, we are able to match detected MAV with one in the prototypes and estimate their orientations with respect to one another.

Besides, from the prior knowledge of the pose information of aspect views, we are able to obtain estimated object depth information, i.e. the exact distance between two MAV by simple projective geometry as below:

\[
d_{\text{object}} = \frac{d_{\text{model}} \times i_{\text{model}}}{i_{\text{object}}} \tag{24}
\]

Before MAV are launched, the measured depth information of the aspect views and its physical length projected on the image with units in pixels have been pre-stored in a dataset. After the correct model is matched with the detected object, its corresponding information can be retrieved and contributed for depth estimation. Fig. 3 illustrates the notion of projective geometry.

VI. Experimental Results

Both Haar feature detection and Kalman filtering are implemented. One captured frame is shown in Fig. 4.

The detection and tracking results are as satisfying as expected. Although there is relatively large false alarm rate, the hit rate is within acceptable range. The results show that detection for MAV performs quite well under pure clean environments such as grass or sky. The complex background and the shadows of MAV result in high false alarm rate. In order to overcome this problem, we can combine with other...
image processing techniques such as texture or color feature extraction to filter out noisy regions. The processing time per frame with both algorithms implemented is around 0.12 seconds which is desirable for fast decision making of MAV.

Under conditions that MAV objects have been successfully detected, regions of interest can then be used for shape matching. After matching with 29 prototypes (in Fig. 2), the model with minimum cost will be selected as our final target. Orientation information can then be obtained accordingly.

One example shown below illustrates the process and result for one detected object tagged as #1 converging to one of the 29 models. 100 sample points are randomly extracted from contour of detected MAV and they will be used for matching with the given point sets from models. Algorithms we adopted here are relaxation labelling and TPS transformation.

The reason why the scalar plot is piecewise is that 29 images are taken from different pitch and yaw angles. We have also carried out another experiment. 29 collected samples (in Fig. 2) will be tested to match with themselves. The shape context distance is expected to be higher as the viewing angle varies from initial position to different view-sphere angles. The result matrix has been expressed in 3D surface form for better visual effects (Fig. 10).

The algorithms we employed in this testing are Hungarian and Affine transformation. Rotation is done via considering mass center at the first iteration only. Cost matrix shows the relative dissimilarity among one another for 29 samples. Based on the multi-dimensional cost vectors, we adopted K-medoid algorithm to partition and cluster them according to their dissimilarity levels. The final grouping result is shown in Fig. 11.

This grouping result is reasonable in terms of relative
Fig. 9. Scalar plot of computed cost between the detected object tagged as #1 and 29 sample images.

Fig. 10. 3D surface plot of cost matrix from pair-comparison among 29 model images.

Fig. 11. Clustering result out of 29 aspect views after K-medoids.

Fig. 12. Positions of cameras are plotted in 3D space to help visualize the relative orientations between the representative object within each group and the cameras after K-medoids.

orientations when we capture the aspect views of MAV. Based on its grouped result, we take the first object from each group as the representative objects and plot them based on their orientations pre-recorded in look-up table. Results show that they are very good representative of all 29 collected views (Fig. 12).

We can see that only by Hungarian algorithm and Affine transformation, the matching result is satisfactory. Compared with TPS and Relaxation labelling, one of its advantages for UAV case is that the computation cost can be greatly reduced. Hence, processing time in general is around 0.37 sec/frame for hungarian and affine case. This is especially good for real time applications.

A flight experiment has been conducted to evaluate the performance of proposed vision-based pose estimation algorithms. A monocular camera is mounted on each autonomous quadrotor with their positions adjusted to monitor each other. GPS data was recorded in real time to compare with the vision-based estimated depth information. 50 frames have been selected and the testing result is shown in Fig. 13.

Compared with the conventional algorithm where depth is estimated based on the width of the bounding box of the detected target without knowledge of the target orientations whose result refers to the green dash line, our method shows prestigious advantages in terms of accuracy. However, due to many environmental and quadrotors’ physical constraints, there still exist large errors with error percentage 16.4% and the estimated depth signal is jerky with variance 1.01 meters. The tentative causes may be due to the low resolution images obtained from the monocular camera which makes shape matching difficult and the disturbance from moving blades in the captured images compared to the static model images. Currently we have one ongoing plan to improve the accuracy of depth estimation which mainly focuses on image quality enhancements for cases where two quadrotors are far away
from each other during flight formation.

**VII. CONCLUSION**

In this paper, we present an innovative approach for MAV detection and pose estimation using only one monocular camera in flight formation of MAVs. Haar feature detection is the initial step to do quick scanning and detection. Kalman Filter has been employed to greatly improve target hit rate and fasten computation process. Based on the accurately segmented target in the image, different shape matching techniques were explored and tested. The result turned out that Hungarian algorithm is faster and the matching result is acceptable. Affine transformation is implemented in our case since we do not need to deal with complex deformed shapes in MAV cases which may require TPS transformations. At last, we demonstrated a few successful examples using shape matching to estimate different MAV poses by using real flight images. The experimental result demonstrates great improvements in depth estimation accuracy compared with conventional methods. To our best knowledge, this is the state-of-art simplest and fastest vision-based pose estimation algorithm using only one monocular camera with relatively high depth estimation accuracy for highly complex objects like quadrotors.

**REFERENCES**


