Rose 2005 – IEEE International Workshop
on Robotic and Sensors Environments
Ottawa, ON, Canada, Sept. 30 – Oct. 1, 2005

Optical Flow Techniques in Biomimetic UAV Vision

Fouad F. Khalil and Pierre Payeur

Vision, Imaging, Video and Autonomous Systems Research Laboratory
School of Information Technology and Engineering, University of Ottawa
Ottawa, ON, Canada, K1N 6N5
[fkhalil, ppayeur]@site.uottawa.ca

Abstract *– Optical flow techniques using grayscale image sequences are known to provide powerful solutions for motion estimation and shape reconstruction applications. In this paper, recent advances in optical flow computation from color image sequences are compared and refined to provide alternative optical flow computation techniques for biomimetic vision and control systems. Experimentation is performed in the specific context of flying insect-inspired systems to provide unmanned aerial vehicles (UAV) or flying robots with the ability to navigate through near-Earth environment without colliding with obstacles.*

Keywords *– unmanned aerial vehicles, collision avoidance, optical flow, near-Earth environments, flying insect-inspired systems.*

I. INTRODUCTION

Traditional larger size UAVs (unmanned aerial vehicles) that fly at high altitudes employ sensing mechanisms based on GPS or radar technologies. However, such technologies fail when it comes to smaller vehicles with small width wings. These smaller size UAVs or micro UAVs (MAVs), which weigh only a few grams, have to fly closer to the ground and navigate complex terrain. It is hard to add radar to their small payloads. Also, GPS lacks accuracy when it comes to low-level flight navigation. On the other hand, it is not possible to program the autopilot system by feeding it with information about every single obstacle.

Thus, alternatively there is an increasing interest among researchers in optical flow methods inspired from the flying insects as a solution for MAVs vision [1]. Optical flow is in fact the method by which flying insects with compound eyes like dragonflies, honeybees, or fruit flies, perceive the world. It allows such creatures with tiny brains and low resolution eyes to gather enough information quickly to perform highly accurate navigation maneuvers even in complex near-Earth environments. Consequently, researchers are seeking to create a new class of MAVs provided with autonomous collision avoidance capabilities inspired from the maneuvering techniques of these flying insects like hovering, searching, escaping and even chasing.

This paper presents two traditional grayscale optical flow methods [2] which can be used on these MAVs. It also describes the extension of these methods into color [3] to obtain better features.

II. OPTICAL FLOW IN INSECT-INSPIRED SYSTEMS

The phenomenon of "optical flow" or flow perspective was first studied by Gibson [4]. The optical flow is defined in [5] as the speed at which texture moves in an image focal plane as a result of relative motion between the observer and objects in the environment. Optical flow is typically formulated as a vector field over an image, in which the vectors define the velocity that the texture is moving in the image plane. There are clear mathematical relationships between the magnitude of the optical flow in an image sequence and the object position. If the object doubles the speed at which it travels, the optical flow will also double. If an object is brought twice as close, the optical flow will again double.

There is much evidence that flying insects utilize optical flow sensors to maneuver through regions with dense obstacle fields. Optical flow in this case refers to the apparent movement of texture in the visual field relative to the insect's velocity. For example, while in-flight, objects which are in close proximity to the insect have higher optical flow magnitudes. Flying insects avoid imminent collisions by saccading (or turning) away from regions of high optical flow.

Without GPS or radar system for navigation, flying insects perform tasks like collision avoidance, altitude control, and even takeoff and landing. Such insect-inspired navigational methods can therefore serve as a model for micro air vehicle (MAV) flight patterns in near-Earth environments. Barrows and Neely [5] depict how a MAV might experience optical flow while in-flight in a near-Earth environment, and how the optical flow could be used for such small scale navigation and collision avoidance.

Barrows [6] also developed a new class of optical flow micro-sensors based on grayscale optical flow. This algorithm is implemented using mixed-mode and mixed-signal VLSI techniques. Barrows [7] also mentioned the idea of incorporating color in UAV vision but did not propose any specific method. However, with such compact optical flow sensors, efficient and robust navigational strategies suited for MAVs can be developed by mimicking the natural behaviors of insects. Applications of these optical flow micro-sensors can be found in [8], [9], [10], [11], and [12] for different real flight patterns.

In general, while in-flight within a certain plane MAVs experience two basic motions: rotation and translation. The optical flow, *OF*, measured in rad/sec, can be defined as a function of the MAV's velocity of translation, *V*, angular velocity, ω, distance, *D*, from an obstacle, and the angle, θ, between the MAV's direction of travel and the obstacle, as shown in Fig. 1. The optical flow seen by the MAV in the

0-7803-9379-1/05/$20.00 ©2005 IEEE

direction of the obstacle is determined by the rate of change of θ and can be calculated as the sum of the flows produced by translational and rotational movements:

$$OF = \frac{V}{D}\sin\theta - \omega \tag{1}$$

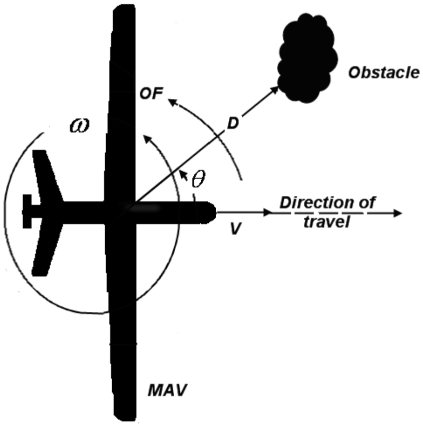

Fig. 1. Optical flow during MAV flight within 1D plane.

In fact, eq. (1) is the source of information provided to the MAV while in-flight about its environment. Accordingly optical flow micro-sensors mounted on the vehicle can be oriented to perceive information about oncoming collisions and altitude. Hence, it is possible to autonomously land a MAV by mimicking insects landing behavior. For example, the honeybees land by keeping the optical flow on the landing surface constant [1, 13]. Therefore, when measuring the optical flow on the landing surface, the obstacle is now the ground and thus $\theta = 90$. To further simplify this task, the rotational component of optical flow, ω, is assumed to be smaller than the translational component. Thus eq. (1) reduces to:

$$OF = \frac{V}{D} \tag{2}$$

In a similar manner to what flying insects do in complex near-Earth environments to avoid collisions, MAVs can also turn away from regions of high optical flow. In this case, optical flow must be detected in front of the vehicle in order to avoid collision. Different collision avoidance strategies based on optical flow inspired from insects are outlined in [1]. Unlike with autonomous landing, where the sensor is oriented at 90 degrees to the direction of travel, the angle θ to the obstacle will be a factor. Assuming the MAV is traveling in a straight path with a relatively constant translational velocity, V, and $\omega \ll |V|$, we have from eq. (1):

$$OF = \frac{V}{D}\sin\theta \tag{3}$$

III. COMPUTING THE OPTICAL FLOW

This section outlines how optical flow can be measured in conventional machine vision techniques [2]. As mentioned before, the optical flow of an image sequence is a set of vector fields, relating each image to the next. Each vector field represents the apparent displacement of each pixel from image to image. If we assume that pixels conserve their intensity, the “brightness conservation equation” can be defined as:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{4}$$

where I is an image sequence, (dx,dy) is the spatial displacement vector for the pixel at coordinates (x,y), and t and dt are respectively the frame time and the temporal displacement between successive images in the sequence.

An obvious solution to eq. (4) is to use template-based search strategies in which a template of a certain size is created around each pixel and the best match is searched for in the next image. The best match is usually found using correlation, sum of absolute difference or sum of squared difference metrics. This process is often referred to as block matching. Such a search strategy is computationally costly and generally doesn’t represent sub-pixel displacements.

Most methods presented in the literature are gradient-based. They solve the differential form of eq. (4), derived by a Taylor expansion. After discarding higher order terms, the optical constraint equation is obtained as follows:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{5}$$

which contains two unknowns, (u,v), the parameters of the optical flow in the image focal plane, for only one equation. However, eq. (5) can be extended to a system of equations in the form of:

$$A\vec{v} = b \tag{6}$$

where $\vec{v} = [u, v]^T$ is the optical flow vector which is to be recovered by solving this over-determined system of equations. Therefore, the problem is ill-posed and extra constraints must be imposed in order to arrive at a solution and achieve the optical flow recovery.

There are many methods used in the literature to impose these extra constraints. These methods are outlined and quantitatively compared in Barron *et al* [2]. The two most widely used methods are namely those of Horn and Schunck [14], and Lucas and Kanade [15]. The Horn and Schunck’s method considers global smoothness of flow across the image, while Lucas and Kanade’s method considers instead a local smoothness in the local neighborhood of the block of pixels which is of interest.

IV. COLOR OPTICAL FLOW

Classical optical flow methods analyzed in [2] as well as the approach proposed by Barrows [6] are all based on grayscale image sequences. Literature on the computation of optical flow from color images still remains limited in spite of the fact that computing optical flow from grayscale images cannot be solved without additional constraints on the nature of the movement.

Color images represent a natural source for such additional information that allows defining an over-determined system of linear equations. Another justification for using color images is that many insects are able to perceive colors and that might be an important factor in achieving biomimetic vision. Therefore, including color in MAV vision provides additional information about the visual field that is not available from intensity images alone. In addition, optical flow recovery from color images can be implemented using direct methods presented in [3], [16] and [17], i.e. without using computationally expensive iterations or search strategies. Finally, the quality of the recovered color optical flow can be assessed and tailored after processing, to provide an effective, efficient and accurate tool for motion estimation.

In grayscale optical flow, the three channels of color data (RGB) which are available to an optical flow algorithm are often combined to form a single intensity image. However, the optical flow constraint, eq. (5), can be extended to form a system of equations, eq. (6), to explicitly consider color information coming from different channels of the image, e.g. red, green and blue. The system becomes:

$$\begin{aligned} \frac{\partial R}{\partial x}u + \frac{\partial R}{\partial y}v + \frac{\partial R}{\partial t} = 0 \\ \frac{\partial G}{\partial x}u + \frac{\partial G}{\partial y}v + \frac{\partial G}{\partial t} = 0 \\ \frac{\partial B}{\partial x}u + \frac{\partial B}{\partial y}v + \frac{\partial B}{\partial t} = 0 \end{aligned} \tag{7}$$

This over-determined set of equations can be solved using least squares techniques.

Another idea proposed in [17] is based on the concept of “color conservation” or color constancy rather than brightness conservation. Accordingly, a single linear system of equations can be defined from color components only while the illumination is allowed to change.

Based on this concept, four color models were investigated in [3] and [18]. These are RGB, normalized RGB, HSV and YUV. The RGB color model decomposes colors into their respective red, green and blue components. Normalized RGB is calculated by normalizing each color by the sum of all three colors at a certain point. The HSV (Hue, Saturation, Value) model expresses the intensity of the image (V) independently of the color (H, S). While YUV decomposes the color as intensity (Y) and color coordinates (U, V)

Barron and Klette [16] apply a color optical flow approach to the particular case of panning and zooming in image sequences. They demonstrate that the use of three color channels provides important improvement in the resulting optical flow over the use of grayscale images.

V. ALTERNATIVE COLOR METHODS

In this paper two alternative color optical flow algorithms inspired from [3] are investigated for an application in the guidance and collision avoidance of MAVs.

A. Fusion of Horn and Schunck

The first algorithm, named Fusion of Horn and Schunck, extends the Horn and Schunck‘s grayscale algorithm to estimate recursively the optical flow of each color plane using least squares, and then fuses the results into one flow field.

In the original grayscale of Horn and Schunck [14], which imposes a global smoothness constraint, the optical flow is assumed to be smooth across the image such that:

$$\|\nabla u\|_2^2 + \|\nabla v\|_2^2 = 0 \tag{8}$$

where ∇ represents a spatial domain differentiation by which the optical flow constraint, eq. (5), can be written alternatively as:

$$\nabla I \cdot \vec{v} = -I_t \tag{9}$$

where:

$$I_x = \frac{\partial I}{\partial x} \qquad I_y = \frac{\partial I}{\partial y} \qquad I_t = \frac{\partial I}{\partial t} \tag{10}$$

Combining eq. (8) with eq. (5) results in the minimization of the following expression [2]:

$$\iint_{\Omega} (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 \left(\|\nabla u\|_2^2 + \|\nabla v\|_2^2\right) dxdy \tag{11}$$

over the domain *Ω* where the smoothness can be controlled by *λ*. Eq. (11) leads to solve iteratively the following two Euler-Lagrange equations:

$$\begin{aligned} u_{k+1} = u_k - \frac{I_x\left(I_x u_k + I_y v_k + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \\ v_{k+1} = v_k - \frac{I_y\left(I_x u_k + I_y v_k + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \end{aligned} \tag{12}$$

where *α* is a smoothness weighting factor.

After estimating the flow at each color channel independently, the proposed extension that results in the Fusion of Horn and Schunck method combines the three recovered flow fields by taking their average. The recovered motion vector could also be selected among the three channels as the one with the smallest estimated error at each point. But this cannot be carried out without a confidence measure that adds extra computational burden on the algorithm.

B. Neighborhood Least Squares

The second proposed color algorithm, named Neighborhood Least Squares, utilizes a simple neighborhood least squares algorithm equivalent to Lucas and Kanade’s but without considering any weighting values.

In the traditional Lucas and Kanade’s grayscale method [15], which implies a local smoothness constraint, the assumption of the constant flow in a local neighborhood is made. In this method weighted least squares are used to solve for $\vec{v}$ in eq. (9) by minimizing:

$$\sum_{x \in R} W^2(x)\left[\nabla I(x,t) \cdot \vec{v} + I_t(x,t)\right]^2 \tag{13}$$

where $W = diag\,[W(x_1),..., W(x_n)]$ and denotes a window function that assigns larger weight to the constraint at the center than at the edge of the domain. ***R*** is a spatial neighborhood. The optimal solution to this problem in the least squares sense is:

$$\vec{v} = (A^T W^2 A)^{-1} A^T W^2 b \tag{14}$$

where A and b are constructed according to eq. (5).

The purpose of the proposed color method is to extend this algorithm to calculate the optical flow from the three color planes, and then have an over-determined set of linear equations formed from a 3x3 window. However, to provide a computationally efficient algorithm, the calculations can also be carried out without considering the weighting matrix, *W*. In this case, when considering the three color planes, values in a 3x3x3 neighborhood around the central pixel are directly incorporated into the $A\vec{v} = b$ system. This over-determined system of equations is then solved using least squares.

VI. ERROR MEASUREMENT

Despite the large volume of research work on optical flow, only two major attempts have been reported in [2] and [19] to evaluate quantitatively the performance of optical flow algorithms on complex image sequences. The difficulty of performing this qualitative evaluation is due to the lack of ground truth data sets, i.e. known correct optical flow for such complex scenes. As a result, the majority of evaluations so far have been mainly qualitative or limited to simple synthetic scenes from which ground truth is easy to extract. Also, few comparative works have been performed on color optical flow estimation.

The error measure considered earlier by Barron *et al*. [2] is the difference in angle between the correct and estimated vectors such that:

$$\theta_{err} = \arccos(\vec{G} \cdot \vec{F}) \tag{15}$$

where θ_{err} is the angular error measure, $\vec{G}$ is the correct optical flow vector, and $\vec{F}$ is the estimated or recovered optical flow vector.

Conversely, Otte and Nagel [19], who opposed the angular error measure of Barron *et al*. for its asymmetry and bias in case of large flow vectors, define the following simple magnitude of difference between correct and estimated flows:

$$M_{err} = \frac{\left| \|\vec{G}\|_2 - \|\vec{F}\|_2 \right|}{\|\vec{G}\|_2} \cdot 100\% \tag{16}$$

where M_{err} is the magnitude error measure.

In this work, in order to evaluate and compare the performance of the algorithms to each other, error analysis is carried out based on statistical quantities (mean and standard deviation) of both the angular and magnitude errors adopted in [2] and [19].

VII. RESULTS AND ANALYSIS

The comparative study and error analysis of the four optical flow algorithms described previously was carried out using a modified verson of the OFCat software simulation package [18]. OFCat is an extensible GUI-driven optical flow computation and analysis tool based on Matlab and Intel's IPL [20] and OpenCV [21] libraries.

The algorithms were implemented on two image sequences, respectively named Rock and Tree, representing scenes of various complexity as found around a flying MAV. Fig. 2 and 3 show the first frame of each sequence. The Rock sequence is a synthetic image sequence [22] for which the ground truth (correct flow) is unknown. In this sequence the optical flow sensor is assumed to aim forward and the MAV moves along its line of sight toward nearby the rock, and hence the focus of expansion (FOE) is near the center of the scene. The FOE is the point from which the optical flow vectors radiate and indicates the MAV's direction of heading. In the Tree sequence case, which is a real image sequence [23], the ground truth is also unknown. In a similar manner to the Rock sequence, the sensor in this sequence is also assumed to be aimed forward and the MAV moves along its line of sight toward the tree with the FOE at the center of the scene.

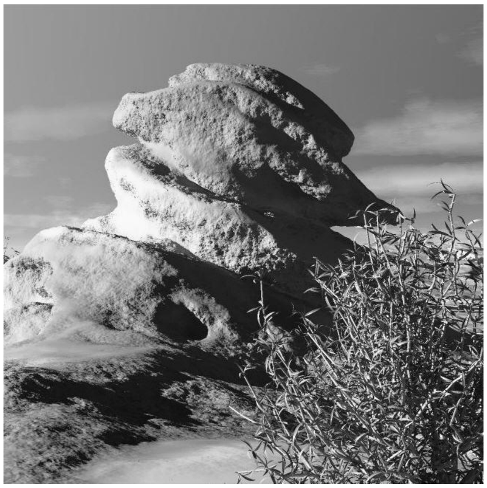

Fig. 2. First frame of the Rock image sequence.

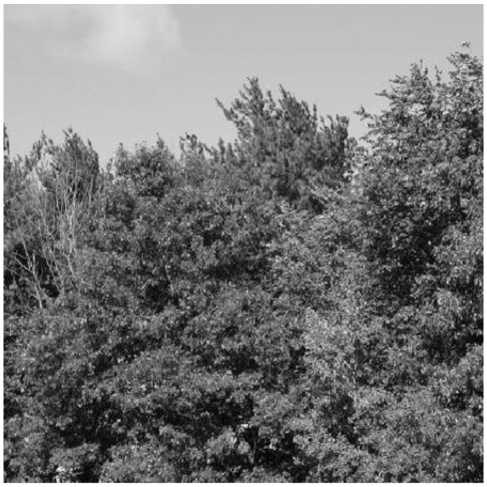

Fig. 3. First frame of the Tree image sequence.

Given that the ground truth for the two sequences is unknown, the performance analysis was carried out by generating a synthetic flow field. For this matter, an image set has been created from a subsample of the first frame by controlling the relative motion of pixels between successive images in the sequence. The resulting set is used for estimating the optical flow and then comparing the recovered flow with the synthetic flow considered as the ground truth.

For each sequence, the recovered optical flow and hence the motion field induced by the corresponding movement is shown in Fig. 4a and 5a respectively. The two sequences were initially encoded with the YUV color model, and the color Neighborhood Least Squares method was used to recover the optical flow. In these two figures, visual observation of the flow field is used to validate the correctness of the recovered field. The optical flow measured in the forward direction of the MAV is related to the existence of any obstacle in front of the sensor. In the Rock sequence (Fig. 4b), the MAV is considered to have a flight path that will cause it to fly nearby the rock (the optical flow field has a large divergence or expansion only on the side of the image containing the obstacle and the FOE is not located inside the obstacle, the rock in this case. In the Tree sequence (Fig. 5b), the MAV approaches the obstacle (the tree), and then turns away to prevent a collision. In this case, the flow field has a large divergence or expansion, and the FOE is located inside the obstacle (the tree) in the visual field.

Tables 1 and 2 compare the performance of the two alternative color optical flow methods (Fusion of Horn and Schunck and Neighborhood Least Squares) with the common grayscale optical flow methods of Horn and Schunck [14], and Lucas and Kanade [15] that were applied on the same sequences after conversion to grayscale images by averaging the three color components. The results presented in the tables are the mean and standard deviation values computed over all pixels of the image for the angular error, and for the magnitude error, as defined in section VI. These results demonstrate that Neighborhood Least Squares performs better than the other tested methods. The Fusion of Horn and Schunck also outperforms its grayscale counterpart, demonstrating the advantage of using color optical flow computation over traditional grayscale image sequences.

Table 3 provides a comparison between the two proposed color optical flow estimation methods applied on the Tree sequence using different color models, respectively, YUV, HSV, RGB, and nRGB. This table shows the same measuring values (mean and standard deviation) calculated for each color model. The YUV color model shows more accurate motion estimation for this case where lighting and scene complexity are relatively uniform.

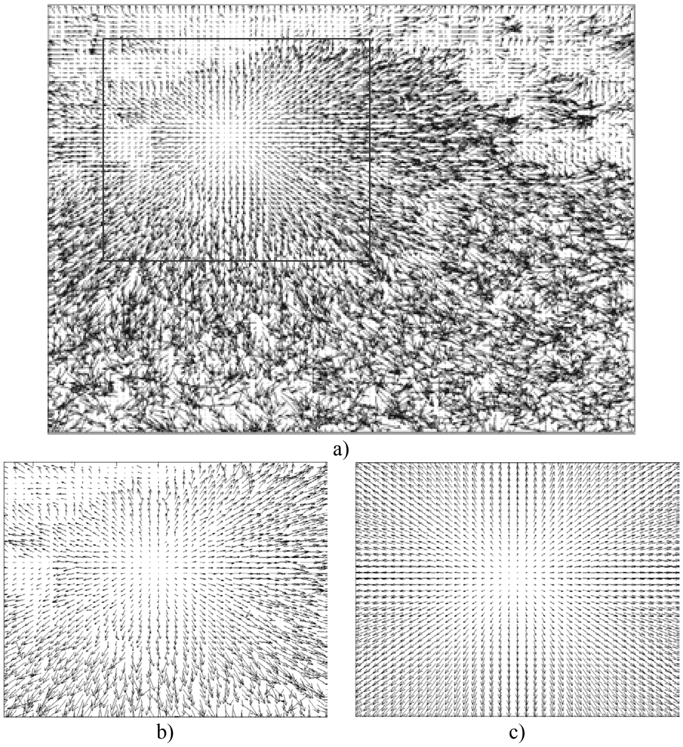

Fig. 4. a) Recovered optical flow for the Rock sequence using Neighborhood Least Squares with b) the magnified region around the FOE and c) the corresponding ground truth optical flow.

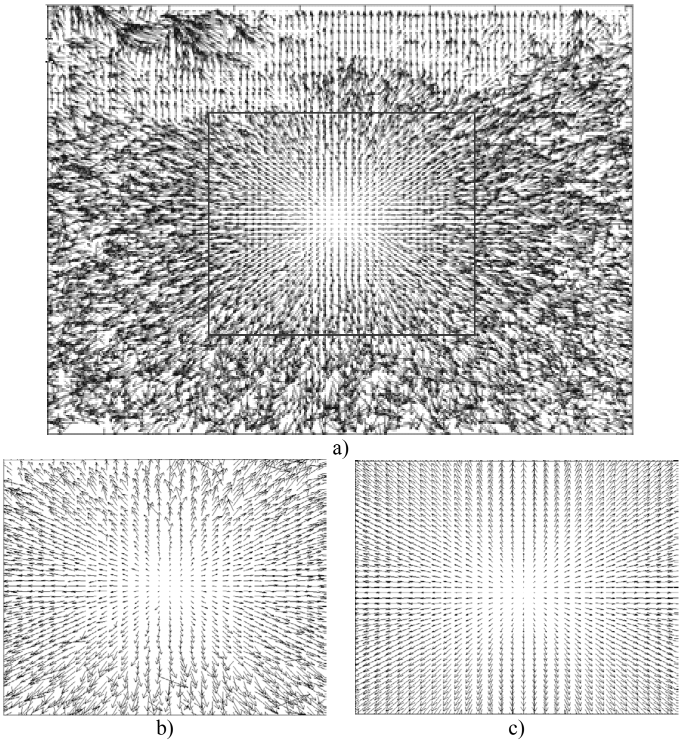

Fig. 5. a) Recovered optical flow for the Tree sequence using Neighborhood Least Squares with b) the magnified region around the FOE and c) the corresponding ground truth optical flow.

Method	Angular Error (degrees)		Magnitude Error (%)	
	Mean	Std Dev	Mean	Std Dev
Neigh. LS (color)	0.615	0.538	2.225	1.901
Fus. H&S (color)	0.729	0.590	2.439	2.106
L&K (grayscale)	1.094	0.683	2.733	2.003
H&S (grayscale)	1.113	0.691	2.918	3.341

Table 1. Optical flow errors on the Rock sequence.

Method	Angular Error (degrees)		Magnitude Error (%)	
	Mean	Std Dev	Mean	Std Dev
Neigh. LS (color)	0.467	0.445	1.611	1.402
Fus. H&S (color)	0.553	0.516	1.793	1.594
L&K (grayscale)	1.062	0.698	2.067	1.444
H&S (grayscale)	1.099	0.716	2.495	3.234

Table 2. Optical flow errors on the Tree sequence.

Method	Mode	Angular Error (degrees)		Magnitude Error (%)	
		Mean	Std Dev	Mean	Std Dev
Neigh. LS	YUV	0.467	0.445	1.611	1.402
Neigh. LS	HSV	0.587	0.543	1.803	1.690
Neigh. LS	RGB	0.779	0.685	2.123	2.209
Neigh. LS	nRGB	1.255	0.179	3.730	1.390
Fus. H&S	YUV	0.553	0.516	1.793	1.594
Fus. H&S	HSV	0.654	5.980	1.963	2.003
Fus. H&S	RGB	0.840	0.695	2.335	2.593
Fus. H&S	nRGB	1.256	0.179	3.731	1.390

Table 3. Errors with different color models on the Tree sequence.

VIII. CONCLUSION

Grayscale optical flow methods have received much attention for recovering optical flow in numerous applications. Color optical flow, on the other hand, has not been investigated as extensively and was not yet considered for UAV vision despite the supplementary information that is available from the three channels of color data.

In this paper, an analytical and comparative study of the traditional optical flow techniques with their extensions to color is carried out. Simulation results show that the two alternative color optical flow methods, namely Fusion of Horn and Schunck and Neighborhood Least Squares, perform better than the corresponding grayscale approaches for estimation of optical flow on complex scenes. This research work demonstrates that the use of these color optical flow techniques could significantly enhance the performance of micro aerial unmanned vehicles (MAVs) for the completion of autonomous tasks and maneuvers like collision avoidance, altitude control, takeoff and landing in near-Earth environments.

REFERENCES

[1] G. Barrows, J.S. Chahl, and M. V. Srinivasan, "Biomimetic Visual Sensing and Flight Control", *The Aeronautical Journal, The Royal Aeronautical Society*, vol. 107, pp. 159-168, London, 2003.

[2] J.L. Barron, D.J Fleet, and S.S. Beauchemin, "Performance of Optical Flow Techniques", *Intl Journal of Computer Vision*, vol. 12, pp. 43-77, 1994.

[3] R.J. Andrews and B.C. Lovell, "Color Optical Flow", *Proc. of the Workshop on Digital Image Computing*, vol. 7, pp. 135-139, Brisbane, 2003.

[4] J.J. Gibson, *The Ecological Approach to Visual Perception*, Boston, Houghton Mifflin, 1979.

[5] G. Barrows and C. Neely, "Mixed Mode VLSI Optic Flow Sensors for In-Flight Control of a Micro Air Vehicle", *SPIE Proceedings, Critical Technologies for the Future of Computing*, vol. 4109, pp. 52-63, 2000.

[6] G. Barrows, *Mixed-Mode VLSI Optic Flow Sensors for Micro Air Vehicles*, Ph.D. Dissertation, University of Maryland, College Park, MD, 1999.

[7] G. Barrows, "Future Visual Microprocessors for Mini/Micro UAV Applications", *Proc. of the Intl Workshop on Cellular Neural Network and their Applications*, Frankfurt, Germany, pp. 498-506, 2002.

[8] W.E. Green and P.Y. Oh, "An Aerial Robot Prototype for Situational Awareness in Closed Quarters", *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 61-66, Las Vegas, Nevada, 2003.

[9] W.E. Green, P.Y. Oh, and G. Barrows, "Flying Insect-Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environment", *Proc. of the IEEE Intl Conf. on Robotics and Automation*, pp. 2347-2352, New Orleans, LA , 2004.

[10] W.E. Green, P.Y. Oh, K. Sevcik, and G. Barrows, "Autonomous Landing for Indoor or Flying Robots Using Optic Flow", *Proc. of ASME Intl Mechanical Engineering Congress*, vol. 2, pp. 1341-1346, Washington D.C., 2003.

[11] P.Y. Oh, and W.E. Green, "CQAR: Closed Quarter Aerial Robot Design for Reconnaissance, Surveillance and Target Acquisition in Urban Areas", *International Journal of Computational Intelligence*, vol. 1, pp. 353-360, 2004.

[12] P.Y. Oh, W.E. Green, and G. Barrows, "Closed Quarter Aerial Robot Prototype to Fly in and Around Buildings", *Proc. of the Intl Conf. on Computer, Communication and Control Technologies*, vol. 5, pp. 302-307, Orlando, Florida, 2003.

[13] M.V. Serinivasan, S.W. Zhang, J.S. Chahl, E. Barth, and S. Venkatesh, "How Honeybees Make Grazing Landings on Flat Surfaces", *Biological Cybernetics*, vol. 83, pp. 171-183, 2000.

[14] B. Horn and B. Schunck, "Determining Optical Flow", *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.

[15] B. Lucas and T. Kanade, "An iterative Image Restoration Technique with an Application to Stereo Vision", *Proc. of DARPA Image Understanding Workshop*, pp. 121-130, 1981.

[16] J.L. Barron and R. Klette, "Quantitative Color Optical Flow", *Proc. of the Intl Conf. on Pattern Recognition*, vol. 4, pp. 251-255, 2002.

[17] P. Golland and A.M. Bruckstein, *Motion from Color*, Technical Report #9513, Centre for Intelligent Systems, Computer Science Department, Technion, Haifa-Israel, 1997.

[18] R.J. Andrews and B.C. Lovell, "OFCat: An Extensible GUI-driven Optical Flow Comparison Tool", *Proc. of the Workshop on Digital Image Computing*, vol. 7, pp. 129-133, Brisbane, 2003.

[19] M. Otte and H.H. Nagel, "Optical Flow Estimation: Advances and Comparisons", *Proc. of the 3rd European Conf. on Computer Vision*, Stockholm, Sweden, pp. 51-60, 1994.

[20] Intel Corporation, *IPL: Intel Image Processing Library*, Reference Manual, Intel Corporation, 2000.

[21] Intel Corporation, *OpenCV: Open Source Computer Vision Library*, Reference Manual, Intel Corporation, 2001.

[22] C. Hormann, POV-Ray rendering software, http://www.povray.org.

[23] J. Rehg, Georgia Insitute of Technology, http://ww.cc.gatech.edu.