AIAA Guidance, Navigation and Control Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-7494

A Comparison of Optical Flow algorithms for Real Time Aircraft Guidance and Navigation

Marco Mammarella [1] Giampiero Campa.[2]
Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506/6106,

Mario L. Fravolini[3]
Department of Electronic and Information Engineering, University of Perugia, 06100 Perugia, Italy,

and

Yu Gu [4], Brad Seanor [5] and Marcello R. Napolitano [6]
Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506/6106.

This paper focuses on the analysis of the performance of several optical flow algorithms for application within aeronautic applications such as obstacle detection and collision avoidance for lightweight unmanned aerial vehicles (UAVs). The study involved the comparison of nine different optical flow algorithms. Some of these algorithms were developed at West Virginia University (WVU); some were included in the Video and Image Processing Blockset® of Matlab® while others were available for download from different sources. The comparative study was performed using both a number of different real-world videos with images rotating or translating at known speeds and a Virtual Reality Environment (VRE) where an aircraft performs complex maneuvers into a detailed virtual world. The comparison of the Optical Flow (OF) algorithms is provided in terms of accuracy of estimation, and an analysis of the computational effort required by the different algorithms was also performed. As expected, algorithms belonging to common conceptual classes turn out to have similar strengths and weaknesses. However, when the computational requirements are taken into account, there is no clear "winning" algorithm, therefore suggesting that ultimately the selection of the "best" algorithm is has to be driven by the particular application.

Nomenclature

C	=	Aerodynamic coefficient
f	=	Focal Length
F	=	Force acting on the aircraft
G	=	Geometric parameters and inertia coefficients
M	=	Moment acting on the aircraft
O	=	Origin of a Reference Frame
P	=	Generic Point
t	=	time
u	=	Horizontal component in Images
$\dot{u}$	=	Horizontal velocity in Images
v	=	Vertical component in Images

[1] Ph.D. Student.
[2] Research Assistant Professor.
[3] Research Assistant Professor.
[4] Research Assistant Professor, AIAA member.
[5] Research Assistant Professor, AIAA member.
[6] Professor.

$\dot{v}$ = Vertical component in Images
V = Velocity
x = x direction in a 3D Reference Frame
y = y direction in a 3D Reference Frame
z = z direction in a 3D Reference Frame

δ = input commands of the aircraft
ω = angular velocity

I. Introduction

Over the last couple of decades, the interest in methods for calculating the optical flow (OF) associated to a sequence of images has been steadily growing, due to the wide range of potential applications. In fact, OF can be used for a variety of purposes, such as motion detection and estimation, collision detection and avoidance, shape reconstruction and object segmentation, etc. In particular, OF-based navigation and control solutions for small size UAVs [1] are becoming appealing since scientific evidence is mounting on the fact that flying insects rely on some form of optical flow to perform quick and highly accurate navigation maneuvers [1]. However, despite some ongoing efforts, significant research issues have to be addressed before wide scale applications of OF-based guidance and navigation control schemes. This is mostly due to the fact that most OF algorithms do not yet provide the measurement density along with the accuracy and the computational speed necessary for real time implementations.

The practical evaluation of the performance of OF algorithms is challenging due the difficulties involved in comparing the different categories of algorithms and the difficulty in the computation of an Ideal Optical Flow in a complex environment. In fact, to date, most of the evaluation studies have relied either on simple sequences generated by progressively applying a distortion to a base image [2], or on simple computer generated image sequences [3]. Unfortunately, real image sequences often present characteristics that are intrinsically hard to model, like motion discontinuities, complex 3D surfaces, camera noise, specular highlights, shadows, transparencies, atmospheric effects, and other sources of disturbances. Applications in complex environment with 6 Degree of Freedom (DoF) are not as well commonly used because of the difficult of the relation between complex 6D motion and projected 2D motion in Camera Reference Frame (CRF). Therefore, it is difficult to assess to what extent the results obtained by comparing OF algorithms on synthetic image sequences can be extended to real and complex virtual environment.

The main contribution of this paper is a detailed comparison of the performance of several OF algorithms. Specifically, 9 different OF algorithms were compared using real image sequences from different experiments involving both rotating and translating motion as well as images from a 6 DoF simulation environment where an aircraft performed a maneuver involving roll, pith, yaw, acceleration and deceleration.

The accuracy of each method was mainly evaluated by computing the ideal Optical Flow. In fact, knowing the rotational and translational velocity it is possible compute the ideal flow for the real image analysis. In the VRE analysis instead, the complete state information of the aircraft is used to find a mathematical characterization of the ideal optical flow generated by the aircraft 6DOF motion with respect to a fixed-altitude plane. Finally, a computational analysis was performed and the computational requirements of each algorithm were highlighted.
The paper is organized as follows. A basic review of OF concepts is introduced in the initial sections. The derivation of the Ideal Optical Flow is presented next. Finally, the experiment set-up and the overall results of the comparison study are illustrated and discussed.

II. Optical Flow

The Optical Flow (OF) is the velocity vector field on the image plane generated by the relative motion of the camera with respect to the objects in the field of view. In general, the methods for OF calculation rely on three fundamental assumptions:

- Brightness Constancy: the local changes in image intensity are caused only by the motion of a certain object with respect to the camera.
- Spatial Coherence: motion is uniform over a small patch of pixels
- Temporal Persistence: the image motion of a surface patch changes gradually over time.

Analytically, if $I(u,v,t)$ is the intensity of a pixel representing a feature that moves of δu, δv during the time δt, then $I(u,v,t) = I(u+\delta u, v+\delta v, t+\delta t)$. The derivative with respect to time leads to the conservation equation:

$$I_u \dot{u} + I_v \dot{v} + I_t = 0 \qquad (1)$$

where I_u and I_v are the spatial derivatives of the image along the u and v image dimensions calculated at a given pixel location, and I_t is the temporal derivative of the image at that location. The considered OF algorithms belong to the following classes:

- "Gradient" Techniques;
  "Phase" Techniques;
  "Matching" Techniques;
  "Feature-based" Techniques.

A. Gradient Techniques

Within (1), for each pixel two scalar unknowns ($\dot{u}$ and $\dot{v}$) are present in a single scalar equation; leading to an analytically under-determined algebraic system. This problem is called the aperture problem. Within the so called "Gradient Techniques", some constraints – usually based on some form of spatial coherence - are added to (1) in order to calculate $\dot{u}$ and $\dot{v}$. Within this effort 4 algorithms pertaining to this category have been analyzed: the "Gradient" method [3], the "Lucas-Kanade" method [4], the "Horn and Shunck" [5] method, and the "Proesmans" [6] method. The "Gradient" algorithm – developed 'ad hoc' at WVU - calculates the OF for each pixel belonging to a predefined grid assuming that $\dot{u}$ and $\dot{v}$ are constant within a certain spatial and temporal neighborhood of the pixel. Therefore, an overdetermined system of equations is assembled and solved – in the minimum square sense – for each considered pixel. Crucially, the system is solved only if its eigenvalues are greater than a given set of thresholds. This allows to discard image areas where derivatives are too close to zero or too similar to each other – e.g. because there is no motion or there is no distinguishable feature – and, at the same time, allowing to speed up the computation by avoiding unnecessary matrix inversions. The Lucas-Kanade" and "Horn and Shunck" implementations are available in recent Matlab versions as Simulink blocks. The "Lucas-Kanade" is very similar to the "Gradient" implementation but it features a weight function to give different weights to the neighborhood in function of their distance to the center. The "Horn and Schunck" algorithm combines equation (1) with a global smoothness term λ to constrain the estimated velocity. Within this algorithm an iterative procedure is used, which is halted when the maximum number of iterations is reached. The "Proesmans" method is in general very similar to "Lucas-Kanade" method. A significant difference arises from the fact that this method takes into account the bias in the direction of the motion due to correlation in the finite difference approximation. The classic advantage of this class of algorithms is their computational speed, while their main disadvantage is that they need to cope with the aperture problem, and that spatial and temporal derivatives are usually very prone to errors.

B. Phase Techniques

Phase techniques are based on the idea that 2D image velocity can be defined as the phase behavior of a band-pass filter output [3]. The idea of using phase information for OF calculation purposes was originally developed by Fleet and Jepson [7]. The algorithm considered within this effort is available from the Matlab Central file exchange site [8] and it is based on the algorithm explained in [9]. This algorithm calculates the OF estimation using the following three sequential steps. First, a spatial filtering is obtained using Gabor Filter and the temporal phase gradient is calculated using the estimation of the velocity components. Second, a component velocity is rejected if the corresponding filter pair's phase information is not linear over a given time span. Third, an interpolation is used to derive the full velocity.

C. Matching Techniques

Matching techniques include the methods in which $\dot{u}$ and $\dot{v}$ are calculated for a given pixel by finding the displacement of a template around the pixel between two consecutive frames. The template matching among two consecutive frames is usually performed by minimizing a predefined function of the difference between the two templates. Within this effort, the "Difference" and the "Correlation" methods have been considered. The "Difference" algorithm uses the sum of the absolute differences (SAD) among templates belonging to consecutive frames to find the best matching templates [3]. The "Correlation" algorithm instead calculates the correlation among templates to perform the matching. Algorithms belonging to this category have shown to be substantially more computationally demanding than the algorithms belonging to the "Gradient" category.

D. Feature-based Techniques

The main concept of "Feature-based" techniques is that it is possible to calculate $\dot{u}$ and $\dot{v}$ through measurements of the displacements of certain image features – as detected by a feature detection algorithm, and associated by a feature matching algorithm - between two consecutive frames. These methods rely on the implicit assumption that

the same image features can consistently be detected and associated over different image frames. In this effort two different feature based techniques have been considered, that is the Harris [10] and the SIFT [11] method. Harris method is a widely used approach for the specific problem of corner detection; in fact, the Harris algorithm allows extracting the position of a corner within with a good reliability even with changes in the light conditions. One of the features of the Harris corner detector is that it needs a point matching algorithm for the task of labeling and tracking the corner(s) belonging to consecutive frames of an image sequence. The associated "Point Matching" algorithm used within this effort was previously developed at WVU [12]. The other feature detection algorithm is known as SIFT (Scale Invariant Feature Transform). SIFT has been developed specifically for recognizing the same features between different images. Features are detected using a filtering approach that identifies stable points in the scale space; next, the features are associated using a descriptor-based approach [11]. Empirical experience has shown that the precision of the methods belonging to this class depend strongly on the performance of the associated matching algorithm.

III. Derivation of the Ideal Optical Flow

The comparison between the different Optical Flow algorithms is based on the calculation of the "Ideal" Optical Flow (or Ideal Flow) generated by the motion of an object in space. Given any point on the image plane, the ideal flow can be calculated from the position and the velocity - with respect to the camera - of the point in the field of view that generates - by projection - the optical flow.
Specifically, a 'pin-hole' mathematical model of the camera [15] is assumed:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{x_c} \begin{bmatrix} y_c \\ z_c \end{bmatrix} \tag{2}$$

where f is the camera focal length, u and v are – as previously described - the horizontal and vertical coordinates of a point in the image plane resulting from the projection of the point ${}^{C}P=[x_c, y_c, z_c]^T$ on such plane.
Note that the left superscript "C" in ${}^{C}P$ indicates that the point is expressed with respect to a camera-fixed reference frame, which is centered in the camera plane and has its x-axis pointing in the direction of view, and its y and z-axis pointing respectively in the directions of u and v of the image plane. Assuming that ${}^{C}P$ is part of a rigid body centered in ${}^{C}O_B$ and moving with respect to the camera reference frame with a linear velocity ${}^{C}V_{B/C}$ and angular velocity ${}^{C}\omega_{B/C}$, differentiating (2) with respect to time, and using standard kinematics relationships to express the derivative of ${}^{C}P$ yields:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = f \begin{bmatrix} -\frac{y_c}{x_c^2} & \frac{1}{x_c} & 0 \\ -\frac{z_c}{x_c^2} & 0 & \frac{1}{x_c} \end{bmatrix} \left[{}^{C}V_{B/C} + {}^{C}\omega_{B/C} \otimes \left({}^{C}P - {}^{C}O_B \right) \right] \tag{3}$$

where $\dot{u}$ and $\dot{v}$ represent the *ideal* optical flow (at the image coordinates u and v) generated by the motion of ${}^{C}P$ and $\otimes$ indicates the three-dimensional cross product.

IV. Experiment set up

Four specific sets of experiments were developed with the objective of comparing the difference of the OF algorithms listed above. The first set is based on the OF produced by a rotating disk; the second set is based on the OF produced by a cart sliding in the horizontal image direction (that is the u direction), and the third set relies on the OF produced by a forward translation of the cart toward the camera. The video was recorded using a "Qware EasyCam WB-001" and its data acquisition software. In the forth set, the algorithms are tested into a simulation environment where an aircraft performs complex maneuvers.

A. 1st Set - Rotating Disk experiment

For this set of experiment, a given picture was glued to a wooden disk, which was in turn attached to a DC motor. A section of 120 frames was then selected for performing the analysis. Two different criteria were selected for comparing the results from the OF algorithms, that is the overall angular velocity error, angular and magnitude errors with respect to the 'ideal' flow. The criteria are briefly discussed below.

- *Overall Angular Velocity Error:*

An estimate of the disk angular velocity is calculated pseudo-inverting (3)with the OF vectors supplied by each algorithm and averaging the results. The estimated angular velocity is then compared with the 'true' measured angular velocity.

- *Angular and Magnitude Errors w.r.t the Ideal Flow:*

For each frame of the video the "ideal" OF was calculated for each image point by using (3), the 'true' disk angular speed and the point position with respect to the disk center. Both the "ideal" flow vector and the flow vector detected from the OF algorithms were then expressed in polar coordinates, resulting in a magnitude and an angle value – with respect to the u axis - for each vector. Finally, the errors in the magnitude and the angle were calculated as the differences between the "ideal" and the "detected" values.

B. 2nd set - Sliding Cart Experiment

This set of experiment consisted in attaching the same picture of the previous experiment to a toy train that was used as a moving cart. Since the camera was positioned on the train side, a pure translational motion of the picture on the *u* axis was recorded. Only a subset of the video frames, specifically the 15 frames in which the train was within the camera field of view, was used for this analysis. As for the previous experiment, the overall velocity error and the angular and magnitude errors with respect to the 'ideal' flow were used as performance metrics:

- *Overall Velocity Error:*

An estimate of the train velocity was calculated directly using the velocities in the *u* and *v* directions provided by the OF algorithms. The estimated velocity was then compared with the 'real' measured train velocity.

- *Angular and Magnitude Errors w.r.t the Ideal Flow:*

For each video frame, the "ideal" OF was calculated in each image point using the true train velocity. Both the "ideal" flow vector and the flow vector detected from the algorithms were then expressed in polar coordinates, resulting in a magnitude and an angle. Finally, the errors in magnitude and in angle were calculated as the difference between the "ideal" and the "detected" values.

C. 3rd set - Forward Translation Experiment

Within this set of experiment, the picture was attached in front of the train with the camera positioned along the longitudinal direction of the train. Therefore, the image of the picture became closer as the train moved forward. A section of 60 frames was selected to perform the analysis. As for the previous experiment, the train velocity was kept constant. The same performance metrics were used to compare the different OF algorithms as in the previous two sets of experiments.

- *Overall Velocity Error:*

An estimate of the train velocity was calculated using the OF vectors supplied by each algorithm. Specifically, for each optical flow vector, a corresponding estimated velocity is calculated by setting to zero both $\omega_{B/C}$ and the third component of $V_{B/C}$ in (3) and pseudo-inverting the formula. The total velocity is then calculated by averaging the velocities calculated from each vector over all the used video frames, and compared with the 'true' recorded train velocity.

- *Angular and Magnitude Errors w.r.t the Ideal Flow:*

For each used video frame, the "ideal" OF was calculated at each image point - by substituting the true train velocity in (3) - and then expressed in polar coordinates. As for the previous experiments, the errors in magnitude and in angle were then calculated as the difference between the "ideal" and the "detected" OF magnitudes and angles.

D. 4th set – Virtual Simulation Experiment

The algorithms are executed within a simulation environment that is linked to a Virtual Reality Toolbox® (VRT) [16] interface. Such interface allows the position and orientation of a flying aircraft in the simulation to drive the position and orientation of an associated visual model of the aircraft in a "virtual world", which was developed by the authors to provide typical airborne scenarios. In particular, within this effort, the scenario consisted of a flight at an initial altitude (700m) over a terrain featuring a repeated picture of a natural landscape, taken from Google Earth ®. The simulation is completed of a detailed non-linear mathematical model of an aircraft developed during previous effort [13] by some of the authors.

A window on this virtual scenario - featuring the view from a virtual camera placed on the aircraft - was made available to the user. Images from such camera were continuously acquired – using functions provided by VRT - and fed to the different optical flow algorithms during the simulation. Specifically, as represented in Figure 1, the

Optical Flow was continuously calculated from each couple of consecutive images, using each of the nine available algorithms.

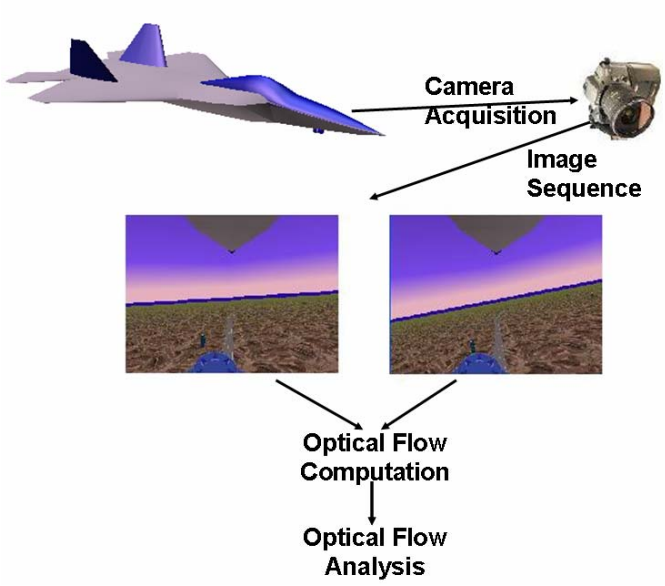

Figure 1: Sequence of operations

A relationship yielding the earth frame coordinates of a generic point ${}^{E}P$ situated on the terrain as a function of the corresponding projected image point [*u, v*] was obtained using the MATLAB® Symbolic Toolbox [17]. Specifically, this relationship was obtained by expressing the generic point ${}^{C}P$ (with respect to the camera frame) as a function of ${}^{E}P$ (with respect to the earth frame). Then, solving the projection equations (2) to find x_e and y_e as a function of all the other variables, including *u*, *v*, the height of the ground expressed in earth frame and the known variables expressing the camera position and orientation with respect to the earth frame. Note that once ${}^{E}P$ is known, its expression with respect to the camera frame ${}^{C}P$ (as a function of *u*, *v*, camera position and orientation) can be found through the application of the appropriate rotation and translation to ${}^{E}P$.

For each image, the ideal OF between that image and the next image was calculated at each image point. For each image point [*u, v*] from which an optical flow vector originates, the camera frame coordinates of the corresponding point ${}^{C}P$ on the terrain is calculated using the formula described in the previous paragraphs, while points that do not belong to the terrain, such as for example points above the horizon, are discarded.

Next, the "ideal" optical flow in the point [*u, v*] is found by substituting ${}^{C}P$ - along with the known translational and angular velocity of the camera with respect to the terrain - in Eq. (3). Expressing in polar coordinates both the ideal and the detected OF vectors - as for the previous experiments - allowed for the calculation of the errors in magnitude and in angle for each detected OF vector. Averaging the errors over all the OF vectors calculated during the simulation yielded the performance metric for the evaluation of the different OF algorithms.

V. Results

The three recorded videos (one for each experiment) were selected to represent the different types of motion that an object can typically undergo in a 3D space. In the video, the glued picture is a chessboard having 0.64 cm wide black-and-white squares. The results are summarized in tables, in each table the algorithms that obtain better performance are marked in bold.

A. Rotating disk experiment

An example of the OF field obtained for the rotating disk experiment - using the SIFT algorithm - is shown in Figure 2.

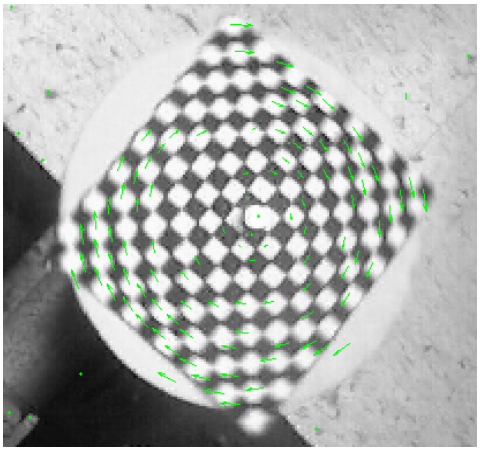

Figure 2: SIFT OF with the rotating disk experiment

The results of the analysis are reported in Table 1. Specifically, the table reports the results obtained by all the nine algorithms when applied to the video.

	ω_{real}	ω_{calc}	*Std*	*Err %*	Ang_{err}	Mag_{err}
Corr	-0.112	-0.057	0.015	49.6	25.8	1.65
Diff	-0.112	-0.044	0.009	60.8	56.4	4.10
Grad	-0.112	-0.012	0.001	89.3	41.7	5.26
Harr	-0.112	-0.084	0.062	25.0	50.5	2.14
Sift	**-0.112**	**-0.109**	**0.004**	**2.9**	**5.9**	**0.51**
Phase	-0.112	-0.035	0.004	69.0	50.4	3.72
Proes	**-0.112**	**-0.061**	**0.009**	**45.3**	**28.2**	**2.78**
L-K	-0.112	-0.017	0.002	85.2	68.1	4.13
H-S	-0.112	-0.037	0.006	67.0	73.7	5.45

Table 1: rotating disk experiment

In the above table, ω_{real} is the 'true' rotational velocity of the disk calculated off-line in radiant per frame. ω_{calc} is instead the average value over 120 frames of the rotational velocity which calculated from the OF field provided by each algorithm. The standard deviation (*Std*) and percentage error between the 'true' and the calculated velocity are also shown in the table. The columns Ang_{err} and Mag_{err} contain the average angular and magnitude errors - measured respectively in degrees and pixels - for each algorithm.

B. Sliding cart experiment

An example of the OF field obtained for the sliding cart experiment - using the Correlation algorithm - is shown in Figure 3, while Table 2 report the results obtained from all the nine algorithms.

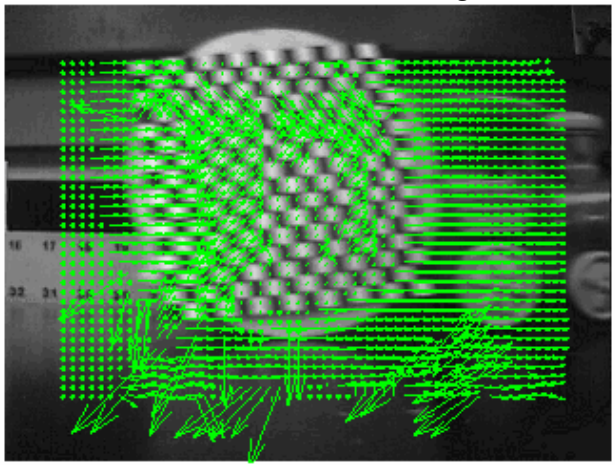

Figure 3: Correlation OF for the sliding cart experiment

	V_v	Real V_u	V_u	Std_u	Ang_{err}	Mag_{err}
Corr	0.73	14.55	5.15	1.05	70.48	11.13
Diff	0.34	14.55	6.18	2.10	55.41	10.03
Grad	0.41	14.55	-0.53	0.11	135.57	15.08
Harris	0.76	14.55	4.59	2.10	64.98	11.52
Sift	**0.14**	**14.55**	**15.44**	**0.46**	**0.93**	**1.14**
Phase	1.21	14.55	-0.95	0.23	123.44	15.57
Proes	2.11	14.55	0.87	1.89	100.72	14.26
L-K	0.55	14.55	-0.95	0.21	130.19	15.56
H-S	0.89	14.55	-0.94	0.18	122.07	15.89

Table 2: sliding cart experiment

In the above tables, $RealV_u$ is the 'true' recorded velocity in the *u* direction. Note that the real velocity along the vertical image axis, that is $RealV_v$, has been considered to be 0. The columns V_u and V_v contain the average – throughout the frames where the sliding picture is visible - of the velocities extracted from the OF field in the *u* and *v* directions, measured in pixel/frame. Finally, Std_u is the standard deviation of the velocity in the *u* direction and Ang_{err} and Mag_{err} contain the average angular and magnitude errors.

C. Forward Translation experiment

An example of the OF field obtained for the forward translation experiment - using the Lucas - Kanade algorithm - is shown in Figure 4.

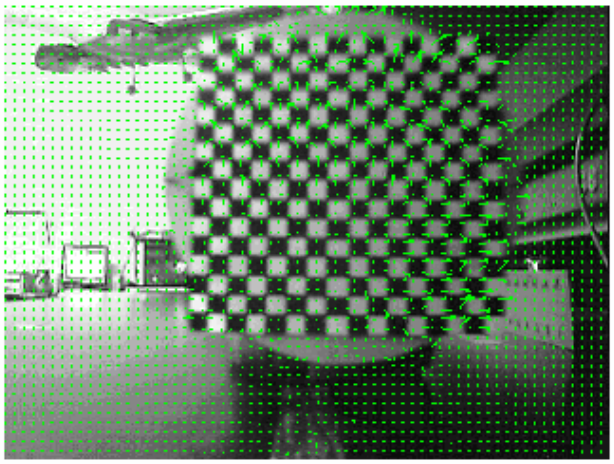

Figure 4: Lucas - Kanade OF for the forward translation experiment

Table 3 reports the results obtained by the nine OF algorithms.

	Real V_x	V_x	Std_x	ERR %	Ang_{err}	Mag_{er}
Corr	-0.168	-0.34	1.42	-103	39.8	2.73
Diff	-0.168	-0.0085	0.48	95.0	75.1	8.50
Grad	-0.168	-0.077	0.02	54.0	35.8	0.77
Harris	**-0.168**	**-0.164**	**0.77**	**2.7**	**63.8**	**3.73**
Sift	**-0.168**	**-0.159**	**0.06**	**5.4**	**27.8**	**0.63**
Phase	**-0.168**	**-0.147**	**0.07**	**12.6**	**41.5**	**0.78**
Proes	**-0.168**	**-0.135**	**0.01**	**19.6**	**37.7**	**0.60**
L-K	-0.168	-0.103	0.02	38.8	43.7	0.82
H-S	-0.168	-0.126	0.04	25.1	57.7	1.43

Table 3: forward translating experiment

In the above tables, *Real* V_x is the 'true' recorded velocity in the x direction. Note that the real velocity along the axis y and z, that is *Real* V_y and *Real* V_z, has been considered to be 0 since the cart motion is parallel to the longitudinal depth axis of the camera. The column V_x contains the average – over the frames where the forward translation is analyzed - of the velocities extracted from the OF field in the x direction, measured in meter/second. Finally, Std_x is the standard deviation of the velocity in the x direction. The columns Ang_{err} and Mag_{err} contain the average angular and magnitude errors - measured respectively in degrees and pixels - for each of the algorithms.

D. Virtual Simulation experiment

Samples of the results of the analysis conducted within this effort are shown in Figure 5 to Figure 14. Figure 5 shows the Ideal Optical Flow calculated as described in the previous sections, note that the Ideal OF can be derived only for the ground. Figure 6 and Figure 7 show the OF calculated with the matching techniques. Figure 8, and Figure 12-12 show the OF calculated with the Gradient techniques. Figure 9 and Figure 10 show the results from the Feature based technique while Figure 11 shows the results from the Phase method.

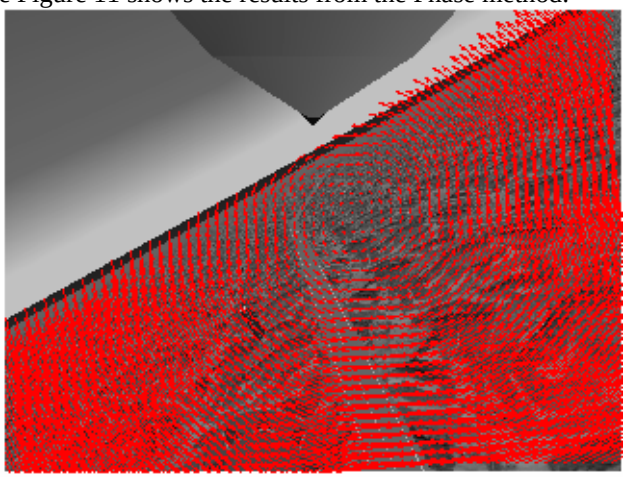

Figure 5: Ideal Optical Flow in the VRE

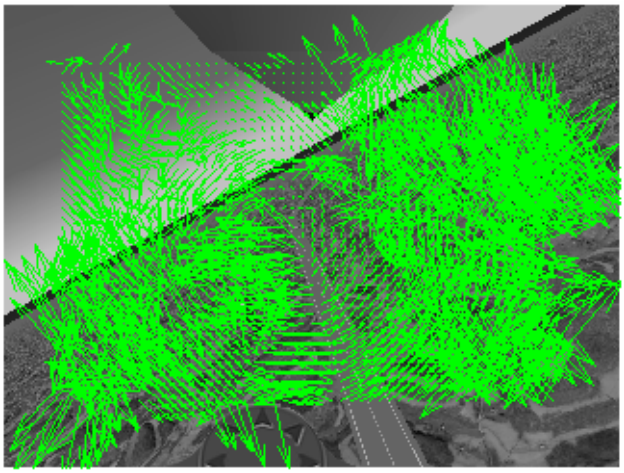

Figure 6: Correlation OF in the VRE

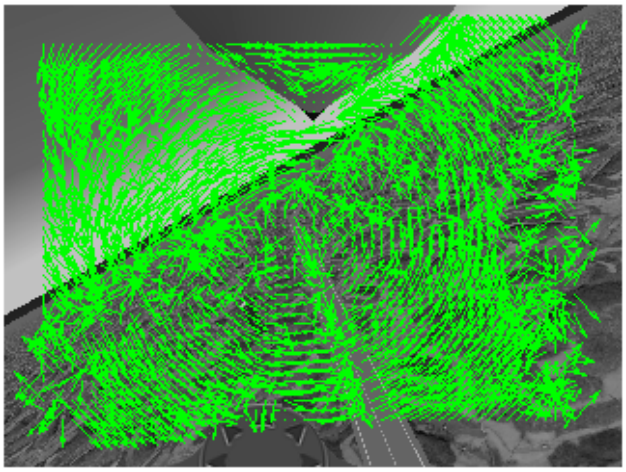

Figure 7: Difference OF in the VRE

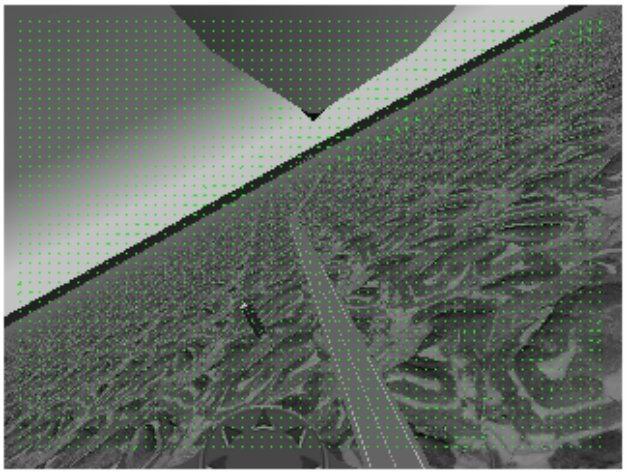

Figure 8: Gradient OF in the VRE

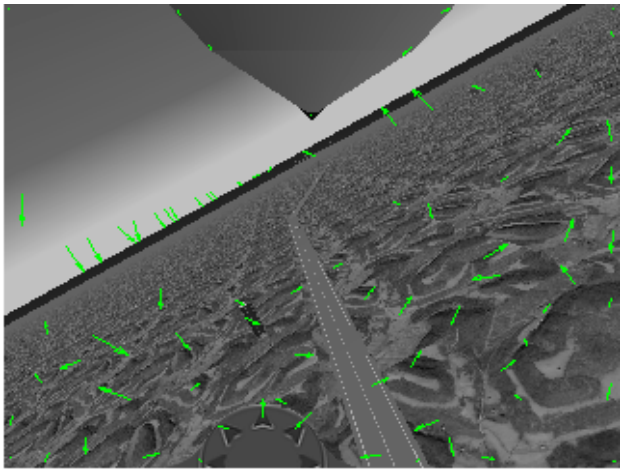

Figure 9: Harris OF in the VRE

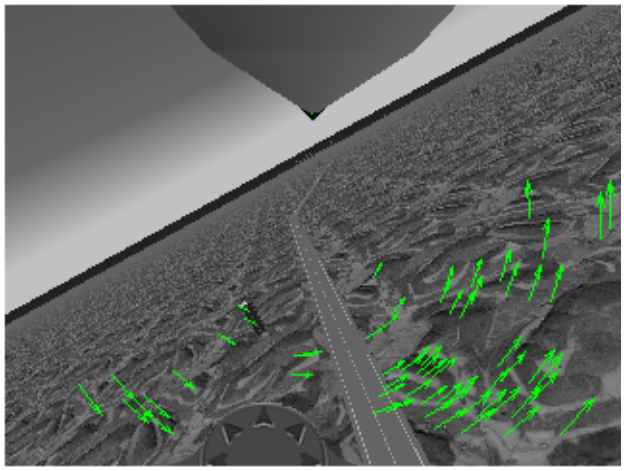

Figure 10: SIFT OF in the VRE

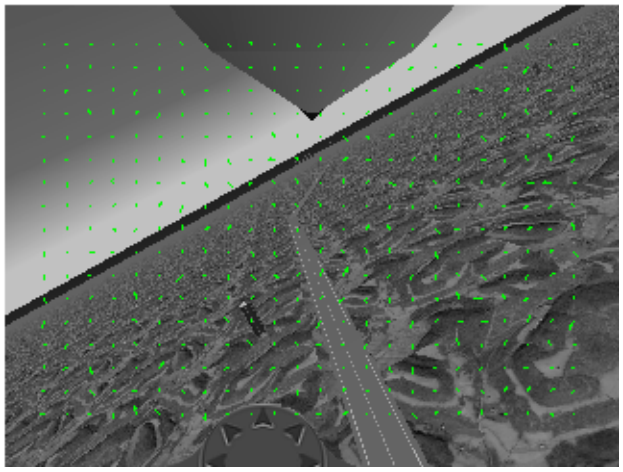

Figure 11: Phase OF in the VRE

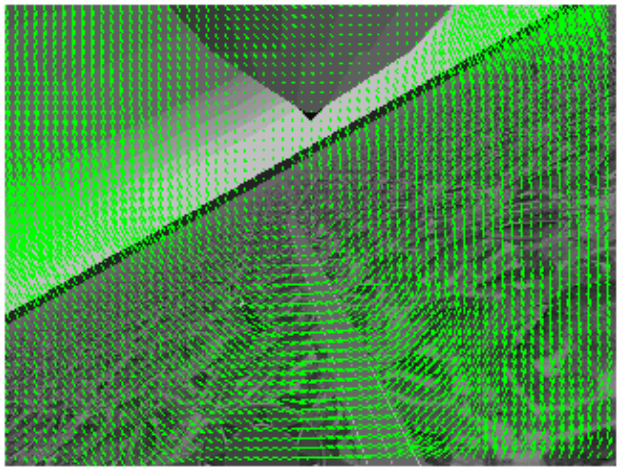

Figure 12: Proesman OF for the experiment in the VRE

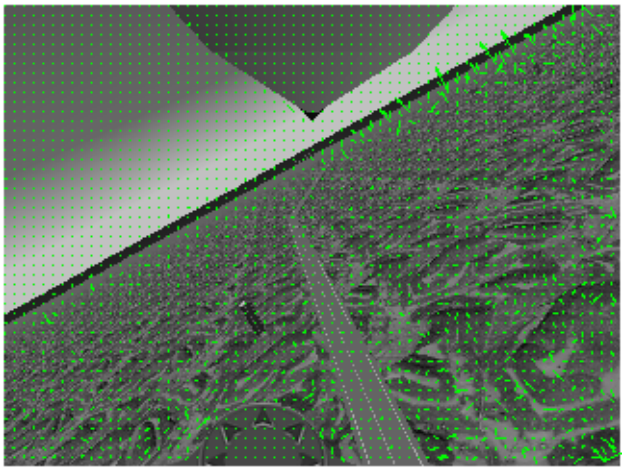

Figure 13: Lucas Kanade OF in the VRE

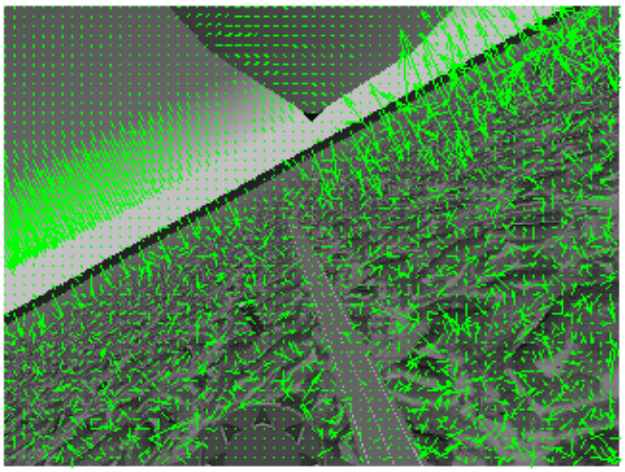

Figure 14: Horn Shunck OF in the VRE

	Mean Ang Error (deg)	*STD Ang Error*	*Mean Mag Error (pix)*	*STD Mag Error*
Corr	**49.62**	**38.68**	**2.19**	**2.98**
Diff	63.18	25.48	4.34	1.22
Grad	74.77	23.99	4.46	3.28
Harris	73.18	19.04	3.14	1.59
SIFT	**61.32**	**41.44**	**4.58**	**5.84**
Phase	70.12	25.35	3.11	2.79
Proes	**57.64**	**33.36**	**3.64**	**2.50**
L- K	72.11	35.13	3.25	3.30
H S	87.61	5.60	3.47	2.58

Table 4: Experiment in the VRE

Table 4 shows the results obtained from a statistic analysis where the algorithms are compared with the Ideal Flow during a 125 seconds simulation with a sampling time of 0.1 seconds. In the table, the algorithms are compared in terms of the mean and standard deviations of the errors in angle and magnitude. A visual analysis revealed that the Difference, SIFT, and Proesman algorithms performed better. However, the statistical analysis in Table 4 showed that the Correlation and Proesman algorithms provided the best results.

E. Computational Requirements Analysis

Algorithm	Corr	Diff	**Grad**	Harris	Sift	Phase	Proes	**L-K**	**H-S**
Time (sec)	9.77	9.36	**0.237**	0.615	1.45	2.95	1.30	**0.122**	**0.142**

Table 5: Time analysis

Table 5 summarizes the results of the time analysis. Each value in Table 5 is the average computational time (sec) for each frame using the different algorithms. This analysis was performed on a 120-frames video using a Pentium 4 dual processor 3.4 GHz with 2GB of RAM memory.

From an analysis of the results it can be noted that SIFT is the algorithm providing the best performance in terms of accuracy of the OF estimations. With the exceptions of the Gradient, Lucas Kanade and Horn and Shunck algorithms, the results of the time analysis are not encouraging for the purpose of real time applications using embedded hardware. However, a leaner and computationally optimized implementation of the SIFT, Phase, and Difference algorithms is envisioned to be feasible.

VI. Conclusions

This paper described the results of a detailed comparative analysis for nine OF algorithms belonging to different classes. The algorithms were implemented and tested using real videos covering the different types of motion in a 3D space and Virtual Reality Environment where an aircraft performs different maneuvers. The statistical analysis on the real videos and on the Virtual Reality Environment indicate that the SIFT feature detection, and Proesman are the algorithms that seem to perform better in terms of accuracy of the estimates. On the other side, simpler algorithms more suitable for real time applications do not appear to provide a consistent level of accuracy for the OF estimations.

Acknowledgments

Partial support for the authors has been provided by Augusta Systems Inc, Morgantown, WV.

References

1. Barrows, G., and Neely, C.: 'Mixed-mode VLSI optic flow sensors for in-flight control of a micro air vehicle', *Proceedings. SPIE*, 2000, 4109, pp. 52–63.
2. Khalil F. F, and Payeur P. "Optical Flow Techniques in Biomimetic UAV Vision", *IEEE International Workshop on Robotic and Sensors Environments*, 2005, pp. 14-19.
3. Galvin, B., McCane, B., Novins, K., Mason, D. and Mills, S. "Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms". In *Proceedings of the 9th British Machine Vision Conference 1998*, pp. 195-204, 98.
4. Barron J.L., Fleet, D.J and Beauchemin, S.S. "Performance of Optical Flow Techniques", *Intl Journal of Computer Vision*, vol. 12, pp. 43-77,1994.
5. Lucas, B., Kanade, T., "An iterative Image Restoration Technique with an Application to Stereo Vision", *Proc. of DARPA Image Understanding Workshop*, p. 121-130, 1981.
6. Horn B.K.P., Schunck B.G. "Determining optical flow" *AI,*. 17. pp.185-204, 1981.
7. Proesmans, M., Van Gool, L., Pauwels E. and Oosterlinck A., "Determination of optical flow and its Discontinuities using non-linear diffusion". In *Proceedings of the 3rd European Conference on Computer Vision,* Vol. 2, pages 295-304, 1994.
8. Fleet D.J., and Jepson A.D. "Computation of component image velocity from local phase information" Int. J. Comp. Vision 5, pp. 77-104, 1990.
9. Gautama T., Phase-based Optical Flow, Matlab Central http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectType=file&objectId=2422 (2002).
10. Gautama T., Van Hulle M.M., "A Phase-based Approach to the Estimation of the Optical". *IEEE Transaction on neural network. Vol.13 No.5.* 2002.
11. Harris, C and Stephens, M, "A Combined Corner and Edge Detector", *Proc. 4th Alvey Vision Conference*, pp. 147-151, 1988.
12. Lowe D.G., "Object Recognition from Local Scale - Invariant Features", *Proc. of the International Conference on Computer Vision,* 1999.

13. Campa, G., Mammarella, M., Napolitano, M. R., Fravolini, M. L., Pollini, L., Stolarik, B., "A comparison of Pose Estimation algorithms for Machine Vision based Aerial Refueling for UAV", *Mediterranean Control Conference 2006*, 2006.
14. Campa, G., Gu, Y., Seanor, B., Napolitano, M. R., Pollini, L., Fravolini, M. L., "Design And Flight Testing Of Nonlinear Formation Control Laws", *Control Engineering Practice*, pp 1077-1092, Vol. 15, Issue 9, 2007.
15. Hutchinson S., Hager G., Corke P., "A tutorial on visual servo control", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, 1996, pp. 651-670.
16. *The Virtual Reality Toolbox User's Guide*, 2001-2007, HUMUSOFT and The MathWorks Inc.
17. Symbolic Math Toolbox User's Guide, 1993-2007, The MathWorks Inc.