OFCat: An extensible GUI-driven optical flow comparison tool

Robert J. Andrews and Brian C. Lovell
School of Information Technology and Electrical Engineering
Intelligent Real-Time Imaging and Sensing
University of Queensland

E-mail: s341199@student.uq.edu.au, lovell@itee.uq.edu.au

Abstract

OFCat is an extensible GUI-driven optical flow computation and analysis tool. The user can add their own image sequences, filtering, differentiating and optical flow techniques. OFCat can process both grayscale and color images. It implements a number of low-pass filtering and differentiating techniques. Many traditional optical flow extraction techniques are available, in addition to some novel color-based methods.

Analysis of the recovered flow is performed using ground-truth analysis, image reconstruction and sparsity vs error analysis. Synthetic ground-truth can be created and used to create a test image sequence from a single image.

1 Introduction

Optical flow has many applications in computer vision and image-sequence processing. Among other areas, it has been used for robot navigation, face tracking and recognition, three-dimensional reconstruction and general analysis of motion.

Barron [1] and others wrote a review paper on optical flow in 1994 and provided code for other researchers to compare their results to. The tool for the analysis and comparison of optical flow described in this paper (OFCat) is intended to be a useful addition to the computer vision researcher's toolbox. It allows use of custom image sequences, lowpass filters, differentiating kernels and optical flow methods. Many techniques and algorithms are implemented in each of these categories. Some novel color-based optical flow techniques are available to the researcher using OFCat, as well as some previously theorised color techniques [5]. In addition, an error analysis section is provided which allows visualisation of the flow, analysis versus ground truth, image reconstruction error and density vs error threshold metrics. Both color and grayscale images are supported, with similar and different methods available for each.

Due to the script processing nature of Matlab® the speed of algorithms can often be increased by compilation. A script is provided for compilation of the algorithms that benefit most in the application. This script assumes a Matlab®-compatible compiler is installed and that Matlab® is appropriately configured.

In addition to automatic C-translation and compilation, it is possible to use MEX (Matlab® shared library file) wrappers to incorporate external C (or C++) code. This possibility is exploited in the application by incorporating functions which use optimized functions from the Intel® IPL® [3] (Image Processing Library) and Intel® OpenCV® [4]. These libraries take advantage of the enhanced multimedia extensions available on Intel Pentium-based chips. Hence, some functions are available for Intel processers only.

When assessing the accuracy of a system, it is necessary to investigate each constituent independently. The recovery of optical flow using gradient-based methods generally involves three steps:

- Low-pass filtering;
- Estimation of first and often second-order derivatives; and
- Recovering optical flow from the derivatives and other metrics

OFCat allows the user to quickly assess the benefits and drawbacks of various algorithms in each of these three categories, with the possibility of adding their own algorithms.

2 Filtering

2.1 Lowpass filtering

Several non-linear de-noising filters have been implemented, listed in table 1. All of these filters are available in color.

Filter
Median Filtering (3x3)
Median Filtering (5x5)
Wiener (Adaptive) filtering (3x3)
Wiener (Adaptive) filtering (5x5)

Table 1. Implemented non-linear de-noising filters

Filter	Color	IPL®
Box (Linear Avg) Filter (3x3)	Yes	Yes
Box (Linear Avg) Filter (5x5)	Yes	Yes
Gauss Filter (3x3) $\sigma = 0.85$	Yes	Yes
Gauss Filter (5x5) $\sigma = 1$	Yes	Yes
Gauss Filter (3x3) $\sigma \neq 0.85$	No	No
Gauss Filter (5x5) $\sigma \neq 1$	No	No

Table 2. Implemented linear de-noising filters

Table 2 lists the linear filters available to the user. In addition to these, the Matlab®'s FDATool (Filter Design and Analysis Tool) can be used to design IIR or FIR filters which can be easily imported into OFCat.

As Gaussian filters are quite popular among vision systems, an additional GUI is provided for the choice of sigma, the shape of the Gaussian.

2.2 Differentiating filters

Spatio-temporal derivatives are necessary for all gradient-based optical flow methods. OFCat computes temporal derivatives separately from spatial derivatives. Three temporal derivatives are implemented; one-sided finite difference, central finite difference and four-point central difference. The corresponding temporal supports are 2, 3 and 5.

Default spatial derivatives available to the user are listed in table 3. Five filters are denoted as Optimal. These are functions utilising filters from Intel®'s IPL®, which more closely approximate the ideal filter as the length of the filter increases [3].

Manduci provides an in-depth theoretical treatment of spatio-temporal filtering and its effect on the accuracy of optical flow methods in his paper [7].

3 Optical Flow methods

OFCat's default optical flow methods can be categorised as working with gray-scale data (one set of partial derivatives) or color data (three sets of partial derivatives). Each category is described below.

Filter	Color	IPL®
Sobel (3x3)	No	Yes
Sobel (5x5)	No	Yes
Prewitt (3x3)	No	Yes
Prewitt (5x5)	No	Yes
Matlab®'s gradient function	Yes	No
Matlab®'s difference function	Yes	No
Optimal 5x5	Yes	Yes
Optimal 7x7	Yes	Yes
⋮	⋮	⋮
Optimal 13x13	Yes	Yes

Table 3. Implemented spatial-differential filters

3.1 Gray-scale optical-flow

Some common methods of recovering optical flow have been implemented, these include many block-matching algorithms and many of the traditional gradient-based methods reviewed in Barron et. al.'s paper [1]. These are: Horn and Shunck, Lucas and Kanade, Nagel, Uras et. al., Uras et. al. with Barron's gaussian curvature and Anandan. Block matching metrics available to the user are: Normalized, Standard, Zero-mean and Zero-mean normalized versions of Correlation, Sum of absolute differences and Sum of squared differences. Most of the gradient-based methods have default parameter values which are modifiable via dialogs.

This large selection of optical flow techniques enables the researcher to quickly evaluate the performance of various techniques on their image sequence. Alternatively, if they develop a new algorithm, they can quickly and easily compare it's performance with existing techniques.

3.2 Color optical flow

The area of color, gradient-based optical flow has been largely overlooked by researchers. The three planes of data available to an optical flow algorithm are usually combined to form one intensity image. Golland [5] presented two methods which utilise this extra information; these and other color-based methods are presented for the user's appraisal.

The optical flow constraint equation is

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (1)$$

We can extend this to a system of equations, in the form $Ax = b$, with each row consisting of data calculated from different planes of the image, e.g. Red, Green and Blue.

This becomes

$$
\begin{aligned}
\frac{\partial I_R}{\partial x}u + \frac{\partial I_R}{\partial y}v + \frac{\partial I_R}{\partial t} &= 0 \\
\frac{\partial I_G}{\partial x}u + \frac{\partial I_G}{\partial y}v + \frac{\partial I_G}{\partial t} &= 0 \\
\frac{\partial I_B}{\partial x}u + \frac{\partial I_B}{\partial y}v + \frac{\partial I_B}{\partial t} &= 0
\end{aligned}
\tag{2}
$$

There are a few options available for the solution to this equation (2), the two that have been the major focus of color optical flow methods implemented are

- Least squares
- Simple gaussian elimination

If we disregard one plane of the image, we have two equations and two unknowns. This is the basis of Golland's "color constancy" equation, disregarding brightness and retaining the hue and saturation values of the image. Alternatively, we can retain the brightness plane (when we use the HSV or YUV color models) and incorporate one of the color planes.

Three different color models are available to the user. These are HSV, RGB and normalized RGB.

In addition to those mentioned above, two novel methods are included.

- Neighborhood least squares, taking a 3×3 neighborhood in each plane and using all of these measurements to create a system of linear equations;
- Application of a standard grayscale optical flow method (e.g. Horn and Shunck [6]) to each plane, then use the intrinsic error measure of each recovered flow to combine them into one flow field.

Figure 1 displays the GUI used for launching color optical flow algorithms. The corresponding grayscale GUI is very similar, though offers different optical flow methods.

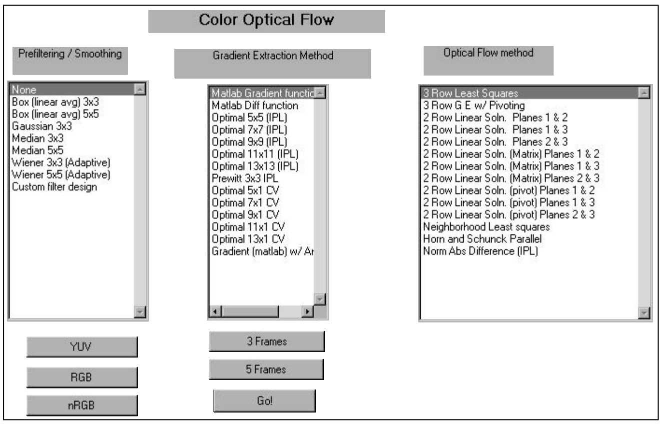

Figure 1. GUI used to select methods for filtering, deriving and estimating optical flow

4 Error Analysis

Four methods of analysing the estimated optical flow are provided. These are detailed in the next four sections.

4.1 Flow visualisation

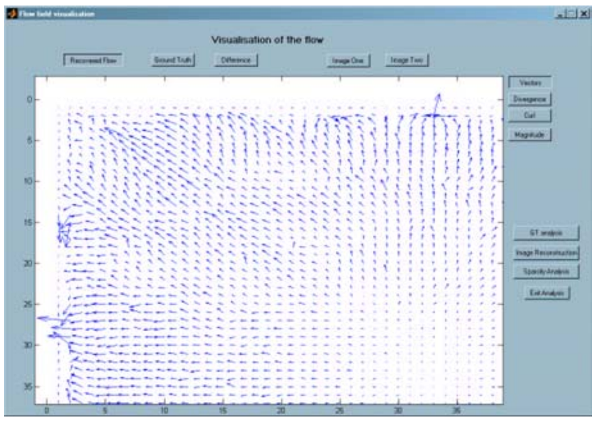

Figure 2. Flow visualisation

Visualisation of the flow is performed via the dialog shown in Figure 2. The recovered flow, ground-truth flow and the difference of both can be viewed here. Additionally, the curl, divergence and magnitude of any of these vector fields can be displayed.

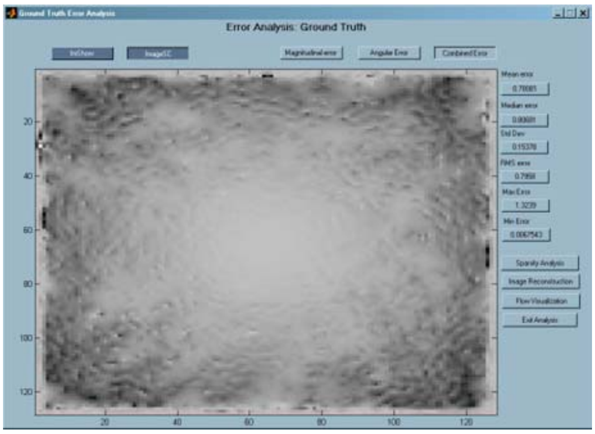

Figure 3. Ground Truth Analysis

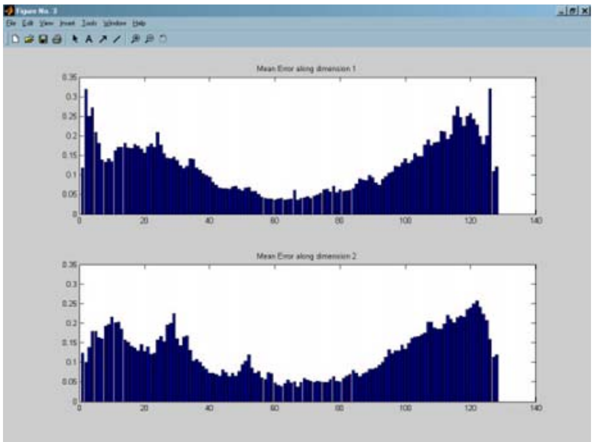

Figure 4. Mean error along each dimension

4.2 Ground Truth Analysis

When ground truth is available, the interface shown in Figure 3 enables the user to analyse the magnitudinal, angular and combined error both visually and statistically. The mean, median, standard deviation, RMS, maximum and minimum errors are calculated and displayed in their respective buttons. Each of the buttons opens a new figure, displaying two plots. The plots are each statistic, respectively, taken across each dimension (horizontally and vertically). An example of these plots is shown in Figure 4.

4.3 Sparsity Analysis

The dialog shown in Figure 5 illustrates the image reconstruction error of optical flow methods by calculating the density of flow fields thresholded at chosen errors. The resulting function is inspected with linear interpolation to estimate the image reconstruction error at certain levels of

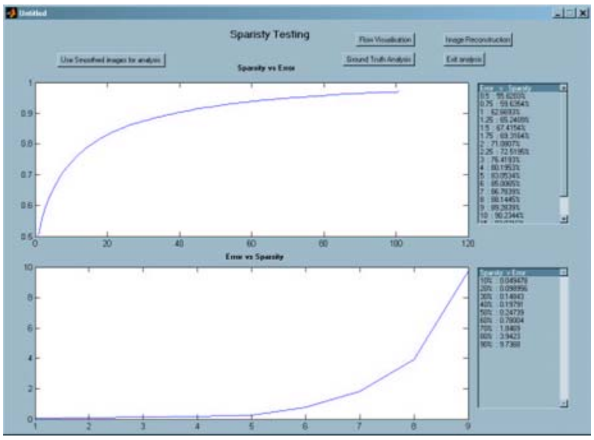

Figure 5. Sparsity Analysis

Figure 6. Image Reconstruction Interface

density. The user can also choose to view the error function when image reconstruction takes the smoothed images for input. The contrast between using smoothed and unsmoothed images illustrates the effect of smoothing on the process of calculating optical flow.

4.4 Image Reconstruction

This interface (Figure 6) allows reconstruction of the second image in the sequence from the first image and the optical flow as per Barron and Lin [2]. The user is able to view the first and second real images, the first and second smoothed images, the reconstructed image, the difference between the reconstructed and second image and the natural logarithm of the difference between reconstructed and second original image.

The natural logarithm of the difference is useful for visualisation of the error as most of the error is of a smaller order of magnitude than the largest errors. Using the natural

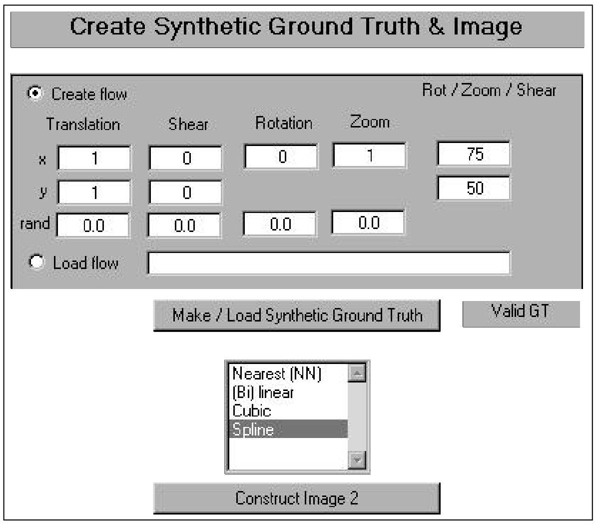

Figure 7. Synthetic Ground Truth Creation Interface

logarithm enables the user to observe the sites perturbed by both large and moderate errors.

The mean of the difference between the reconstructed and correct image is also calculated and displayed.

Thresholding of large vectors (a magnitude greater than > 5 pixels) and thresholding on a chosen image reconstruction or intrinsic error level is supported. The mean returned after thresholding is the mean of only the accepted reconstructed sites. The density of the reconstructed image is also shown. This indicates how many sites were deemed unacceptable and consequently disregarded.

4.5 Creating synthetic error

When ground truth is not available for a sequence, it is still possible to perform ground truth analysis if we create a synthetic sequence based on the first frame of the original. Synthetic planar ground truth can be described by a global translation, shearing in the horizontal and vertical planes, rotation and zoom. If the user has their own synthetic ground-truth flow field saved in a .MAT file, OFCat can load and use this instead.

Figure 7 shows the dialog enabling creation of a synthetic image sequence and ground truth from parameter values. After the synthetic ground truth has been calculated, it is used to warp the first image of the image sequence. This process is repeated until we have a sequence of five frames. Both color and grayscale images are supported.

This sequence then becomes the input to the filtering / optical flow method. As ground-truth is now available, methods of ground-truth analysis can be applied.

5 Conclusion and Future Work

OFCat is a visual, extensible, optical flow analysis tool, intended for students learning about optical flow or for researchers implementing their own methods. Adding new algorithms for filtering, differentiating or computing optical flow to OFCat is simple. OFCat provides quantitative and qualitative error analysis, making it easy for researchers to compare their new method(s) with traditional ones.

Two extra color models will be implemented in the near future, these are the YUV and CIE (UCS) color systems.

OFCat is available to for download at
http://itee.uq.edu.au/ ~iris/ComputerVision/OFCat/
The user must have Matlab® and Intel®'s IPL® installed. Scripts are included for recompilation of the optimized functions.

6 Acknowledgements

The authors would like to thank David McKinnon and Nic Geard for their assistance in creating this document.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Systems and experiment performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.
[2] J. L. Barron and T. Lin. Image reconstruction error for optical flow. 1995.
[3] I. Corporation. Intel ipl reference manual. Technical report, Intel Corporation, Feb 2000.
[4] I. Corporation. Intel opencv reference manual. Technical report, Intel Corporation, Feb 2000.
[5] P. Golland and A. M. Bruckstein. Motion from color. Technical report, Computer Science Dept, Technion, I.I.T., Haifa, Israel, 1997.
[6] B. Horn and B. Shunck. Determining optical flow. *Artificial Intelligence*, (17):185–203, 1981.
[7] R. Manduci. Improving the accuracy of differential-based optical flow algorithms. UCB//CSD- 93-776, University of California at Berkeley, 1994.