Performance of Temporal Filters for Optical Flow Estimation in Mobile Robot Corridor Centring and Visual Odometry

Chris McCarthy
Department of Computer Science and Software Engineering
The University of Melbourne, Parkville, Vic 3010.
cdmcc@cs.mu.oz.au

Nick Barnes
Autonomous Systems and Sensing Technologies Programme
National ICT Australia, Locked Bag 8001
Canberra, ACT 2601.
nick.barnes@nicta.com.au

Abstract

We present a comparison of three temporal filters used in the estimation of optical flow for mobile robot navigation. Previous comparisons of optical flow and associated techniques have compared performance in terms of accuracy and/or efficiency, and typically in isolation. These comparisons are inadequate for addressing applicability to continuous, real-time operation as part of a robot control loop. We emphasise the need for comparisons that consider the context of a system, and that are confirmed by in-system results. To this end, we give results for on and off-board trials of two biologically inspired behaviours: corridor centring and visual odometry. Our results show the best performing filter for use in the control loop is a recursive temporal filter, out performing the traditionally used Gaussian filter. Results for a large Gaussian filter indicate that long latencies may significantly impede performance for real-time tasks in the control loop.

1 Introduction

For a number of years there has been interest in the use of optical flow for vision-based mobile robot navigation, aiming to achieve robust performance for navigational tasks. There is compelling evidence in biological vision, of the use of optical flow in perception and navigation in animals. Studies of vision in flying insects have highlighted visual motion as an important cue for navigational behaviours such as: obstacle avoidance, graze landings, centred flight in corridors and the estimation of distance travelled (see [Srinivasan and Zhang, 2000] for a review). This has inspired new approaches to mobile robot navigation using optical flow.

Biologically-inspired visual behaviours such as corridor centring [Coombs and Roberts, 1993; Santos-Victor and Sandini, 1995], obstacle avoidance [Coombs *et al.*, 1998], and docking [Santos-Victor and Sandini, 1997] have all been demonstrated using visual motion for closed loop control of a mobile robot. The use of optical flow for estimating distance travelled (visual odometry) has also been implemented [Weber *et al.*, 1996].

While reported results have been encouraging, mobile robot research has not broadly adopted this paradigm. This is possibly due to a perceived lack of robustness of these techniques. The paradigm also lacks a defined systematic approach to the implementation and integration of such behaviours into a general navigation framework. Contributing to these issues is the choice of method for optical flow estimation. Despite the reliance on optical flow of such behaviours, the choice of method for estimating flow remains difficult. The literature gives no clear indication of any definitive choice when considering mobile robot navigation. There is an abundance of optical flow techniques available, with varying levels of accuracy, robustness and efficiency. The choice of algorithm for robot navigation or any real-time system operating in the real world, is an important question as research further explores this paradigm of robot control.

Methods for estimating optical flow span a number of categories (see [Beauchemin and Barron, 1995] for a review of these). Previous comparisons suggest gradient-based optical flow methods generally perform well when considering accuracy and efficiency trade-offs, and have gained the most attention in the literature. Even within this class of flow methods, accuracy and efficiency levels vary significantly. Gradient-based methods, defined by their use of spatio-temporal intensity derivatives, typi-

cally require pre-smoothing to estimate flow accurately. Image gradient estimates are highly sensitive to noise and have been shown to perform poorly in high contrast regions of an image [Kearney and Thompson, 1987] (e.g. object boundaries in the scene). For this reason, spatio-temporal filters are used to pre-smooth images before derivatives are estimated. While increasing accuracy, temporal filters pose issues for real-time use in control of a robot. Temporal filters differ in required levels of temporal support (buffered frames) and yield varying levels of accuracy, latencies and computation times. The performance of gradient-based flow algorithms weighs heavily on the choice of temporal filter. This highlights a need for the inclusion of temporal filters when comparing optical flow methods for mobile robot navigation.

Comparisons to date have primarily assessed optical flow techniques and temporal filters on accuracy and/or efficiency, and only in isolation. Since Barron *et al.* [1994] published the first major performance comparison, there have been several attempts to reflect more closely, real-world scenarios and the constraints of real-time use when evaluating techniques. McCane *et al.* [2001] use synthetic image sequences of higher complexity as well as real images with ground truth optical flow on which to quantitatively compare accuracy under real-world conditions. Otte and Nagel [1995] present a quantitative comparison of accuracy for real images obtained from a smoothly translating robot arm. Bober and Kittler [1994] give accuracy measures when Gaussian noise and multiple motions are introduced in synthetic images. Liu *et al.* [1996] address the issue of real-time use through an examination of accuracy/efficiency trade-offs for optimal performance on synthetic and real image sequences. In the case of gradient-based methods, we have not seen any thorough comparison of temporal filters when considering the real-time application of gradient-based optical flow methods. Furthermore, optical flow is generally not compared with consideration of the task to be performed, particularly for operation within the control loop of a mobile robot. We consider in-system issues important when evaluating optical flow methods and temporal filters.

Quantitative comparisons of accuracy and efficiency alone, do not provide sufficient information on which to base a choice for mobile robot navigation. Real image sequences obtained under highly controlled conditions are inadequate for understanding the performance of the flow technique under real-world conditions, particularly when operating on-board a mobile robot. Such comparisons are useful for benchmarking, but inadequately support any systematic choice of flow technique or temporal filter for use in real-time navigational behaviours.

In this paper we present preliminary work in the comparison of optical flow for mobile robot navigation. We compare three temporal filters applied with Lucas and Kanade's gradient-based optical flow method [1981] for mobile robot navigation. Lucas and Kanade was chosen based on its strong performance in previous comparisons of accuracy and efficiency [Barron *et al.*, 1994; Liu *et al.*, 1996]. The temporal filtering techniques for comparison are: Gaussian filtering with central differencing, Simoncelli's multi-dimension pre-smoothing and derivative filters [Simoncelli, 1994], and Fleet and Langley's recursive temporal filter [1995]. We focus only on behaviours involving continuous motion such as corridor centring and visual odometry. Section 2 presents an overview and theoretical comparison for each of the techniques under consideration. Section 3 sets out the methodology for comparing these techniques for the two behaviours mentioned above. Section 4 presents results obtained from off-board trials conducted on a constructed sequence of real images with ground truth flow fields. Section 5 presents on-board results, comparing filters when embedded in the robot control loop. Section 6 gives our conclusions.

2 Overview and Theoretical Comparison

In this section we briefly describe the temporal filters examined in this comparison, and provide an initial theoretical comparison of each method for mobile robot navigation. Traditionally, Gaussian filtering has been used for pre-processing in gradient-based flow estimation. Typically, kernel sizes are large to achieve high accuracy, giving rise to concerns about performance in real-time systems. Of interest are the effects that a large filter has on real-time performance and whether better performances can be achieved using alternative temporal filters of smaller size and delay. We therefore include two possible alternatives for this comparison, which are described below. We also include a brief description of Lucas and Kanade's gradient-based optical flow method which has been used in all comparisons conducted for this paper. The reader is referred to the cited references for a more thorough description of all these techniques.

2.1 Lucas and Kanade Flow Estimation

Optical flow estimation assumes that intensity across the image remain constant over time. This is formally represented by the gradient constraint equation:

$$I_x(\mathbf{x},t)u + I_y(\mathbf{x},t)v + I_t(\mathbf{x},t) = 0, \qquad (1)$$

where $\mathbf{x}$=(x,y), $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, and I_x, I_y, and I_t represent partial intensity derivatives of the image $I(\mathbf{x},t)$. Optical flow at each pixel is represented by the component of motion in the horizontal and vertical direction:

$\mathbf{v}=(u,v)$. The existence of two unknowns and one constraint forms an ill-posed problem, and therefore further constraints are needed to solve for $\mathbf{v}$.

Lucas and Kanade [1981] apply a model of constant velocity as a second constraint, on small local neighbourhoods of the image. The model is applied through a weighted, least squares fit of local first-order constraints. This is achieved by minimising:

$$\sum_{\mathbf{x}\in\omega} W(\mathbf{x},t)(\nabla I(\mathbf{x},t)\cdot\mathbf{v})+I_t(\mathbf{x},t)))^2, \qquad (2)$$

where $W(\mathbf{x},t)$ denotes a window function and ω is the spatial neighbourhood. Improved accuracy of the flow field can be achieved by thresholding eigenvalues of the least-squares matrix associated with (2), however we did not apply this in any of the comparisons described here.

2.2 Gaussian Filtering

Gaussian image pre-smoothing is applied in convolution using a finite number of samples of the Gaussian distribution function as the convolution kernel. Derivatives are then estimated by convolving a differencing kernel over 1-D neighbourhoods of the smoothed image in the direction of the gradient being estimated. Typically, this is done with a size five kernel, allowing two pixels on either side of the central pixel to be included in the differencing. This is known as four point central differencing(4pcd). Temporal derivatives require convolution over multiple frames and so images are buffered. The total number of buffered images required is determined by the size of the derivative kernel and the level of smoothing, given by the standard deviation of the Gaussian. Barron *et al.* [1994] showed that gradient-based methods achieve significantly improved accuracy when a Gaussian filter of 1.5 standard deviation is applied before estimating derivatives. For this comparison, we include two Gaussian filters for use with 4pcd: a large Gaussian with 1.5 standard deviation (Gaussian 1.5), and a smaller Gaussian 0.5 filter.

2.3 Simoncelli's Matched-Pair Filters

Simoncelli [1994] proposed a filter design for obtaining accurate multi-dimensional derivative estimates using a small linear-phased low-pass filter and derivative filter. Unlike traditional approaches, the low-pass and derivative filters are related by their simultaneous design and applied as a matched pair through convolution with input images. To obtain accurate derivative estimates, an intermediate interpolation of the noise and discrete input signal is required. Simoncelli's low-pass filter is obtained by sampling the function performing this interpolation. A derivative filter is then obtained by sampling the derivative of this function. This yields a matched-pair of filters sharing a direct relationship through their design. Reported results [Simoncelli, 1994] show that using 5-tap low and high pass filters (kernel size five), superior derivative estimation is achieved to that obtained using a Gaussian 1.5 filter. The implementation used for this paper employs a pre-smoothing step (size three filter) before applying the 5-tap matched-pair filters. We refer to this entire technique as the Simoncelli filter.

2.4 Recursive Temporal Filter

Fleet and Langley [1995] proposed a causal temporal smoothing and differentiation filter that is applied recursively on each incoming frame. The filter design alleviates problems of time delay encountered with standard temporal filters by implicitly carrying forward past frames through recursive applications of the filter. This reduces storage requirements and latency (typically two or three frames). Images are filtered via a cascaded implementation of an order n filter, where n is the number of cascades used (and consequently, the temporal delay). A time constant, τ^{-1}, gives the duration of temporal support. Fleet and Langley [1995] report a minimal loss in angular accuracy when incorporating an order three filter (n=3, τ^{-1}=1.25) with Lucas and Kanade. Over two synthetic image sequences, accuracy was only slightly worse than results obtained using a Gaussian 1.5 filter. We include this version of the filter in this comparison.

2.5 Theoretical Comparisons

Here we compare the above-mentioned temporal filters for real-time use in mobile robot navigation on the basis of: accuracy, efficiency, robustness and responsiveness. We aim to highlight differences that may lead to contrasting results when evaluating the in-system performance of each filter.

Accuracy

The accuracy of gradient-based flow methods is sensitive to spatio-temporal derivative estimates. Based on this, the Simoncelli filter would be expected to yield the most quantitatively accurate flow. Reported derivative estimates in [Simoncelli, 1994] show an improvement on Gaussian 1.5 filtering.

Reported results for the recursive filter [Fleet and Langley, 1995] indicate that angular error for flow estimates obtained from synthetic image sequences, are slightly larger than Gaussian 1.5 filtering for the same sequences. Qualitative assessment of flow field patterns generated from real image sequences indicate the recursive filter produces flow fields resembling those obtained using a Gaussian 1.5 filter and Lucas and Kanade.

For a Gaussian 0.5 filter, average angular error will increase due to raised levels of noise. Assessing whether the reduced accuracy is within an acceptable level is a

Filter	time (ms)	support (frames)	latency (frames)
Gauss 0.5	116	9	4
Gauss 1.5	170	15	7
Recursive	110	3	3
Simoncelli	106	7	3

Table 1: Efficiency data for temporal filters

system-dependent trade-off, highlighting a need for in-system evaluation. Based on reported results, all other filters appear to produce errors within acceptable levels, giving no clear indication of their relative in-system performances. To further differentiate these filters, other issues for real-time, in-system use must be considered.

Efficiency

For a 192x144 pixel image sequence, Table 1 shows computation times[1], storage requirements and latencies for all filters. Computation time includes time taken to perform filtering and derivative estimation, storage requirements represent the number of frames that are required to be explicitly stored, and latency indicates response delay in frames for each filter.

A Gaussian 1.5 requires the explicit storage of 15 frames to compute flow. Concerns arise for real-time use when considering the computational costs and seven frame delay of the Gaussian 1.5 filter. Temporal delay and storage requirements are improved significantly for Gaussian 0.5 filters, giving a four frame latency. With superior accuracy to the Gaussian 1.5 filter, the Simoncelli filter requires the explicit storage of just seven frames, and has a delay of just three frames. The recursive filter also has a latency of three frames (depending on chosen parameters), and requires the explicit storage of only three frames.

The Simoncelli and recursive filters both compare well for latency. The recursive filter however requires less than half the frame storage of the Simoncelli filter. The Gaussian 1.5 filter rates poorly in all aspects of efficiency. Reducing it to a Gaussian 0.5 filter improves efficiency at the cost of accuracy. As mentioned above, in-system comparisons will assess the effects of this trade-off.

Robustness

The relatively large temporal support of the Gaussian 1.5 suggests it should be robust to noise in detected visual motion. This is an important consideration for mobile robot navigation. Noise induced by the robot's motion through small bumps and fluctuations in direction and velocity have a significant effect on the estimated flow. Sensitivity to changing conditions is desirable, however, it is also important to filter small fluctuations due to noise. This avoids the introduction of unnecessary directional and velocity changes in robot ego-motion, causing oscillations and control instability.

The Gaussian 1.5 filter should inhibit small fluctuations in apparent motion, thereby providing better temporal cohesion than smaller sized filters, such as the Simoncelli filter. The recursive filter, while only requiring a small explicit temporal support, carries forward past frames implicitly giving it support from all past frames. This should provide temporal cohesion in consecutive flow estimates, providing a comparable (possibly better) level of robustness to noise to that of the Gaussian 1.5.

Responsiveness

Where significant environmental changes occur, responsiveness in the estimated flow is important for a timely and proportional motor response. This is dependent upon temporal smoothing, and the extent to which temporal cohesion is enforced.

There is a clear trade-off of robustness to noise and responsiveness to genuine environmental change perceived in the flow field. Of concern for the recursive filter is the extent to which it inhibits change due to the weighting of influence on past frames only. The recursive filter, unlike the Gaussian and Simoncelli filters, has no look-ahead, and relies only upon past frames to perform all temporal smoothing and derivative estimation. This may inhibit the filter's immediate response to image motion changes. The Gaussian 1.5 filter has a seven frame look-ahead, and therefore should have reasonable responsiveness to changes in apparent visual motion for the central frame under consideration. This look-ahead has little value for real-time use due to the seven frame delay in motor-response to the central frame.

Less temporal support should reduce temporal cohesion to a shorter time period, thereby increasing responsiveness to change. Temporal delay would also be reduced, allowing the control scheme to base motor commands on more current data. This suggests that the Simoncelli and Gaussian 0.5 filters should be the most responsive filters. However, this may not be the case if noise levels become large. Excessive noise may result in a lack of responsiveness, causing a reduction or loss of signal due to noise overwhelming the flow field. Reported accuracy for the Simoncelli filter suggests it should not suffer from this problem. The Gaussian 0.5 is at greater risk given the expected increase in noise levels when used.

3 Methodology

The assessment of optical flow performance in a real-time system requires more than a comparison of accuracy or efficiency in isolation. While both factors are impor-

[1]Times taken on an Intel x86 866 MHz machine.

tant, an accurate assessment of performance can only be obtained when considering the context of the tasks being performed. To this end, we set out a methododolgy for the comparison of optical flow methods and temporal filters, for mobile robot navigation where motion is continuous and predominantly constant. Descriptions are given for two selected navigational behaviours implemented for this comparison (corridor centring, and visual odometry). These were chosen because they represent a contrast of uses for optical flow in robot navigation. We highlight important properties of flow estimation that serve as performance indicators to these behaviours. These form the basis of comparisons described in later sections.

3.1 Corridor Centring

Corridor centring using optical flow draws upon observations in flying honeybees. The balancing of visual motion in both eyes, generated by the bees' ego-motion provides an emergent centring behaviour in corridors[Srinivasan and Zhang, 2000]. For robot navigation, this can be achieved by differencing the average magnitudes of flow in the left and right thirds of the image from a single, forward looking camera:

$$\theta = \tau_l - \tau_r, \tag{3}$$

where τ_l and τ_r are the average flow magnitudes in the left and right peripheral views respectively. θ can be directly fed into a control scheme for directional control.

Given constant motion and the expected constancy of a straight corridor, the flow field response should exhibit consistent average flow magnitude. The robot should be free of short period directional oscillation resulting from noise introduced through the robot's ego-motion. Frequent and current flow updates are needed to maintain behaviour stability. Long period directional oscillation through reduced responsiveness is the likely side effect of such latencies. This can be assessed off-board through the examination of temporal cohesion in consecutive flow estimates where ego-motion is approximately constant. On-board trials can be qualitatively assessed for consistency in robot directional control for a straight corridor where environmental changes are minimal.

3.2 Visual Odometry

Distance travelled can be quantitatively estimated by examining accumulated image motion over time, referred to as visual odometry. This can be used for map building, navigating with a map or for correcting shaft encoder odometry. For a single, forward-looking camera, distance travelled can be estimated by accumulating average flow magnitudes in the peripheral regions of the image over time, where at a given discrete time t, the visual odometer, d_t, is given by [Weber *et al.*, 1996]:

$$d_t = \sum^{t} \frac{4}{[\frac{1}{\tau_l} + \frac{1}{\tau_r}]}, \tag{4}$$

where corridor width is represented in the denominator, reducing sensitivity to lateral motion.

Visual odometry is sensitive to environmental change, and is not expected to reproduce odometry readings in different environments for the same distance. It should, however, reproduce the same approximate distance measure for multiple runs in the same environment. For comparison of methods, average distances travelled and variances can be examined for multiple on-board trials, providing quantitative measures of repeatability. However, on-board trials are subject to other factors such as oscillatory directional control, lateral drift, and environmental changes. These are important in-system considerations which can be accounted for by examining the visual odometer in isolation.

Off-board comparisons over real image sequences allow quantitative comparison against ground truth. Accuracy can then be assessed by examining the growth of accumulated visual motion over time. If the distance measure is accurate and repeatable, we expect it to differ by only a scale factor, s, from a ground truth visual odometer. This scale factor should remain approximately constant over time such that:

$$s = \frac{g_t}{d_t} = \frac{g_{t-1}}{d_{t-1}} = .. = \frac{g_1}{d_1}, \tag{5}$$

where d_t is the current value of the visual odometer and g_t is the corresponding ground truth visual odometer value.

4 Off-board Comparisons and Results

Off-board experiments were conducted using a real, heavily textured image sequence cropped to 100x100 pixels as shown in Figure 1. The image sequence shows the motion of a wall moving in a near parallel direction to the optical axis of the camera. The velocity of the wall's motion is approximately 5mm per frame, however the wall's apparent visual motion with respect to the camera is subject to small fluctuations throughout the sequence.

To assist quantitative comparison, ground truth flow fields were generated for images in the sequence. Ground truth was obtained by calibrating the camera and employing a projective warping technique as described in [McCane *et al.*, 2001]. For increased robustness, the model fitting technique RANSAC [Fischler and Bolles, 1981] was used to exclude outlying, manually acquired image feature correlations. We present the experiments and results below.

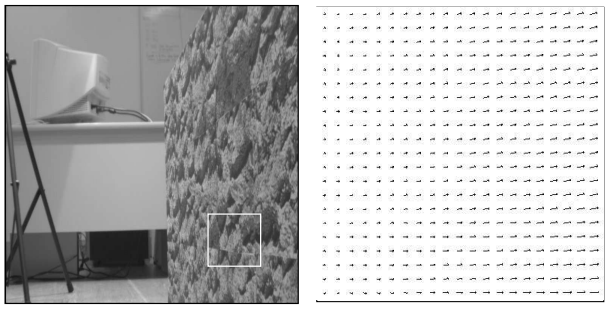

Figure 1: Full frame from side wall sequence and ground truth flow field for the 100x100 pixel (boxed) region used in off-board comparisons.

4.1 Corridor Centring

Flow magnitude consistency was identified as a significant performance indicator for the stability of corridor centring where robot ego-motion is reasonably constant. The side wall image sequence described above was used, allowing the comparison of temporal filters in a well-formed, but realistic scenario. Fluctuations in the wall's apparent image motion allowed for a comparison of consistency under imprecise conditions.

For each frame in the sequence, average flow magnitudes were computed from estimated flow fields using Lucas and Kanade and each of the temporal filters. These results are presented in Figure 2, showing the consistency of average flow magnitude for all temporal filters and ground truth, across the side wall image sequence. In Figure 2, no latency has been accounted for (i.e. the graphs plot average flow magnitudes for the frame for which flow was estimated).

Figure 2 shows all filters producing average flow magnitudes that vary over time. This is consistent with the estimated ground truth which indicates that apparent visual motion between frames is not uniform, despite the approximately constant motion of the wall when the sequence was constructed. The Gaussian 0.5 curve exhibits the worst consistency over the image sequence, showing relatively sharp fluctuations in response to each frame. The other three temporal filters exhibit greater temporal cohesion, showing smoother curves. The Simoncelli filter performs slightly worse with sharper fluctuations between some frames. This is most evident between frames 19 and 20, as well a frames 28 to 30.

The recursive and Gaussian 1.5 filters show approximately equal consistency. The two filters, however, can be further differentiated when real-time considerations are introduced.

Figure 3 shows the same data as Figure 2, however, this time with temporal delay for all of the filters represented. This graph shows the real-time responses of each filter across the sequence (i.e. the actual response

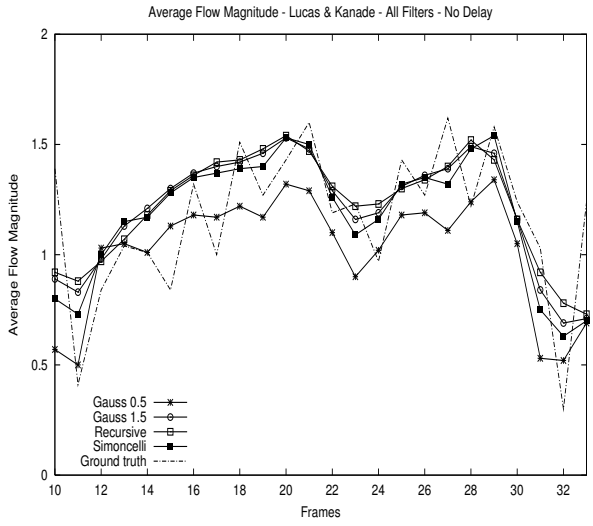

Figure 2: Average flow magnitudes for all temporal filters without latencies.

that would result at a given frame). The recursive filter produces an equal level of consistency with a latency of just three frames, as opposed to the seven frame delay of the Gaussian 1.5 filter. The Simoncelli filter, despite being slightly less consistent over the sequence, appears a more attractive option than the Gaussian 1.5 filter when temporal delay is considered.

4.2 Visual Odometry

The comparison of visual odometry for each temporal filter was conducted using the same side wall sequence as described above. At each frame, the accumulating visual odometer was updated using (4).

Figure 4 shows accumulated visual motion experienced over time for each temporal filter. This graph shows the Gaussian 1.5, recursive and Simoncelli filters all exhibiting growth closely resembling ground truth. The Gaussian 0.5 filter accumulates less motion over time, suggesting large flow magnitudes are undetected by the smaller Gaussian filter.

Table 2 shows average scale factor errors ($av(\mathbf{s})$) and standard deviations (σ) of scale factor errors for each filter when compared with ground truth across the sequence. Scale factor errors were calculated using the value of the ground truth visual odometer at each corresponding odometer update.

According to (5), $\mathbf{s}$ should ideally remain constant over time for the estimate of distance travelled to be accurate. All filters exhibit similar levels of deviation. The last column in Table 2 shows variances of scale factor error when calculated from the straight line approximation to the ground truth visual odometer ($\mathbf{s_{av}}$). The recursive filter shows marginally less variance in $\mathbf{s}$ and $\mathbf{s_{av}}$.

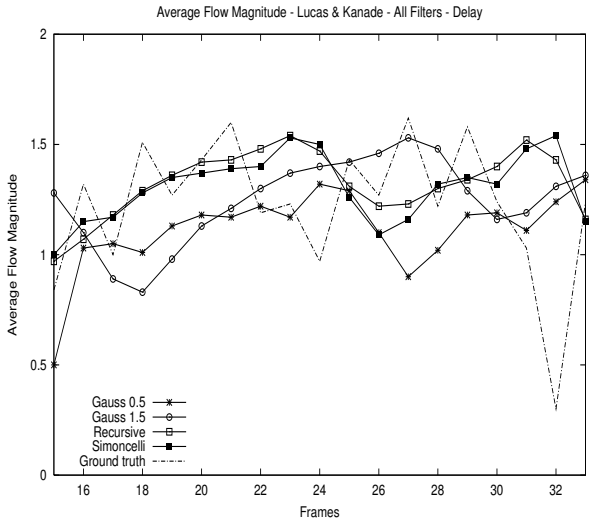

Figure 3: Average flow magnitude for all temporal filters with latencies.

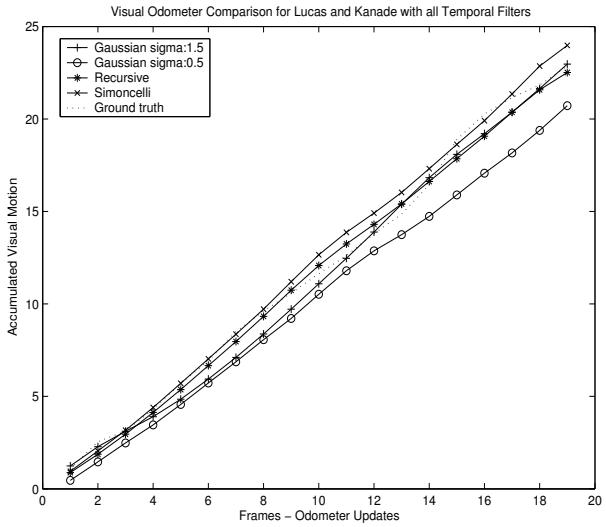

Figure 4: Accumulated Visual Motion for all filters.

Table 2: Visual odometry error analysis.

Filter	$av(\mathbf{s})$	$\sigma(\mathbf{s})$	$\sigma(\mathbf{s_{av}})$
Gauss 1.5	1.06	0.10	0.12
Gauss 0.5	0.87	0.14	0.14
Recursive	1.07	0.09	0.11
Simoncelli	1.03	0.11	0.13

The Gaussian 0.5 exhibits the most deviation over the sequence on both metrics.

4.3 Discussion

Off-board flow consistency results suggest the recursive and Simoncelli filters should perform well for corridor centring. The Gaussian 1.5 filter, despite its relatively good consistency, is likely to be impeded by temporal delay. The extent to which this delay effects its performance in the control loop will be seen in on-board trials. The Simoncelli filter, despite exhibiting slightly less consistent flow magnitude over the sequence, appears a more attractive option when latency is considered. Results for the Gaussian 0.5 filter suggest it would not perform as well when used in the control loop for corridor centring.

Visual odometry results suggest no clear distinctions between filters, however the recursive filter did perform the best overall. The Gaussian 0.5 filter exhibited the most variance in scale factor error, suggesting it is the least accurate of the filters. Given the large implicit temporal support of the recursive filter, and its reduced latency, it is expected that this filter would produce the most accurate and robust distance estimate. This off-board comparison does not take into account filter latencies, thus not showing the effects of this on distance estimation.

5 On-board Comparisons and Results

In this section we present results for comparisons conducted on-board a mobile robot. Implementations of all temporal filters and Lucas and Kanade's gradient-based optical flow method were integrated into the robot control software, running on an Intel x86 866MHz PC with radio link to a mobile robot. An on-board camera, facing in the direction of forward motion was tethered to the PC. Frames from the camera were sub-sampled to 192x144 pixels, with a frame rate of 12.5 frames/sec. Robot locations were plotted using a calibrated overhead camera, and tracking software running on a separate machine. We present results of both comparisons below.

5.1 Corridor Centring

Trials were conducted for each temporal filter using two corridor scenarios: a straight corridor approximately 2.5 meters in length and a slightly longer curved corridor. The width for both corridors was kept approximately constant at 0.6 meters. Only directional control was used, with forward velocity kept constant at $0.15m/s$ for all trials. This speed was chosen empirically, being considered slow enough to adequately assess consistency for all filters and not exceed maximum detectable flow.

A simple proportional control scheme was used for directional control, where for each filter, the proportional gain K_p, was empirically chosen from multiple trials of both the straight and curved corridor. The control scheme was deliberately made simple to assess flow estimations on the basis of control performance. We acknowledge that more sophisticated control schemes can be employed to improve performance, however, we wish

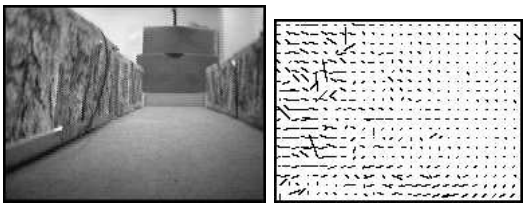

Figure 5: Sample on-board frame and flowfield.

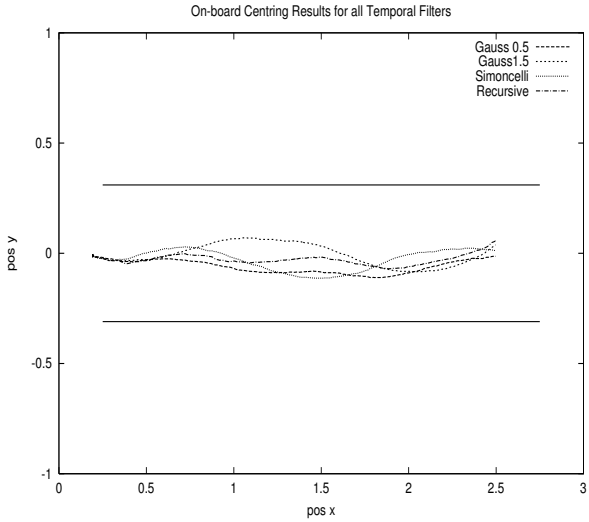

Figure 6: Best straight corridor results for all filters.

to observe the full effects of the filters without dampening from derivative control. Figure 5 shows a sample frame and estimated flow field from the robot in the straight corridor. Figure 6 and Figure 7 show the best performances for each filter in the straight and curved corridor scenarios respectively.

5.2 Visual Odometry

On-board visual odometry trials were conducted using the straight corridor and centring behaviour used in the previous comparison. For each filter, four trials were conducted. The robot started from the same position each trial, and moved down the corridor until the accumulated visual motion exceeded a preset threshold. Table 3 shows the preset maximum visual distance to travel, the average distance travelled along the corridor and standard deviations (in meters) for each filter.

5.3 Discussion

Corridor centring results show that for both corridors, the strongest performing filter was the recursive filter, yielding low amplitude oscillation in the straight corridor, and a near centred path in the curved corridor. The Gaussian 0.5 filter appears to perform well in the

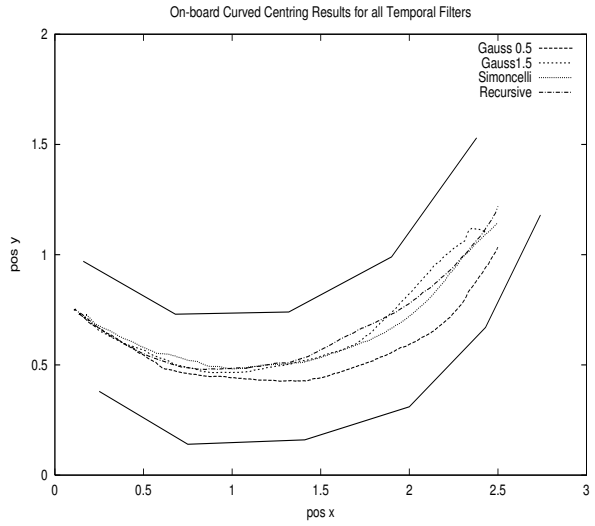

Figure 7: Best curved corridor results for all filters.

straight corridor, but its curved corridor performance showing a lack of responsiveness, suggesting the success of its straight corridor performance is due to this lack of responsiveness. Of the four Gaussian 0.5 trials conducted in the curved corridor, it failed three times. Increasing K_p did not improve this. The Gaussian 1.5 oscillates significantly in the straight corridor. In other Gaussian 1.5 trials conducted, the oscillation was typically much greater than this. In the curved corridor, it had a relatively high fail rate of three fails from six trials. Comparing this with the Simoncelli and recursive filters where no fails were observed, suggests that the larger delay of the Gaussian 1.5 filter is a likely cause for its instability.

On-board visual odometry trials exhibited more variation in results than off-board comparisons showed. From Table 5.2, the strongest performance was clearly given by the recursive filter, with a standard deviation of just 3 cm. The next best was the Simoncelli filter with a recorded 6cm standard deviation. The more stable centring control, and high update frequency of the recursive filter is the likely reason for its superior performance. All filters out performed the Gaussian 1.5, most notably the Gaussian 0.5. The larger temporal delay of the Gaussian 1.5 appears to have effected its performance, most

Table 3: On-board visual odometry results.

Filter	updates	avg dist (m)	std dev (m)
Gauss 0.5	38	2.17	0.07
Gauss 1.5	35	2.20	0.10
Simoncelli	38	2.12	0.06
Recursive	45	2.31	0.03

likely due to large oscillation in the corridor. performance. The strong performance of less accurate filters indicates a trade-off of computation speed and accuracy in real-time performance. This highlights the importance of in-system comparisons for these techniques.

6 Conclusion

In this paper, we have presented results for the comparison of temporal filters for gradient-based optical flow estimation in mobile robot navigation. We have emphasised the need for comparisons of vision techniques that consider the context of a system. Results were presented for on and off-board trials of two navigational behaviours: corridor centring and visual odometry.

Over all comparisons conducted, the strongest performances were achieved using the recursive filter. Strong performances in corridor centring were also achieved using the Simoncelli filter. Despite the greater temporal support given to the Gaussian filters, results showed the recursive and Simoncelli filters consistently out performed both Gaussian filters.

The recursive filter appears to be the strongest performing filter, particularly when integrated into the control loop. Short latency and large implicit temporal support appear to be the reasons for this. This is an encouraging result for the use of gradient-based optical flow in real-time, real-world conditions, where a relatively fast filter with short delay appears to out perform the larger Gaussian filter. Off-board results suggested the Gaussian 1.5 filter could suffer from latency issues, which appeared to be the case in on-board centring trials. The Gaussian 1.5 oscillated significantly more than others for the straight corridor case, a likely result of its large temporal delay. Latencies appeared to also impede the Gaussian 1.5 filters performance in visual odometry trials.

Acknowledgements

The authors would like to thank Professor John Barron, at the University of Western Ontario, Canada, for providing the implementation of Simoncelli's, matched-pair 5-tap filters.

References

[Barron *et al.*, 1994] J L. Barron, D J. Fleet and S S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[Beauchemin and Barron, 1995] S S. Beauchemin and J L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, September 1995.

[Bober and Kittler, 1994] M. Bober and J. Kittler. Robust motion Analysis. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 947–952, June 1994.

[Coombs *et al.*, 1998] D. Coombs, M. Herman, T. Hong and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. *IEEE Transactions on Robotics and Automation*, 14(1):49–59, February 1998.

[Coombs and Roberts, 1993] D. Coombs and K. Roberts. Centering behavior using peripheral vision. In *1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 440–445, New York, NY, USA, June 1993.

[Fischler and Bolles, 1981] M A. Fischler and R C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[Fleet and Langley, 1995] D J. Fleet and K. Langley. Recursive filters for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,17(1):61–67,January 1995.

[Kearney and Thompson, 1987] J K. Kearney and W B. Thompson. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Pattern Analysis and Machine Intelligence*, 9(2):229–244, March 1987.

[Liu *et al.*, 1996] H. Liu, T H. Hong, M. Herman and R. Chellappa. Accuracy vs. efficiency trade-offs in optical flow algorithms. *Computer Vision and Image Understanding*, 271–286, 1998.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of DARPA Image Understanding Workshop*, pages 121–130, 1984.

[McCane *et al.*, 2001] B. McCane, K. Novins, D Crannitch and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, October 2001.

[Otte and Nagel, 1995] M. Otte and H. Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-Interlaced image sequences. *Artificial Intelligence*, 78(1):5–43, November 1995.

[Santos-Victor and Sandini, 1995] J. Santos-Victor and G. Sandini. Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, 14(2):159–177, March 1995.

[Santos-Victor and Sandini, 1997] J. Santos-Victor and G. Sandini. Visual Behaviours for Docking *Computer Vision and Image Understanding*, 67(3):223–238, September 1997.

[Simoncelli, 1994] E P. Simoncelli. Design of multi-dimensional derivative filters. *Proceedings of 1st International Conference on Image Processing*,pages 790–794, Austin TX USA, November 1994. IEEE Signal Processing Society.

[Srinivasan and Zhang, 2000] M V. Srinivasan and S. Zhang. Visual navigation in flying insects. *International Review of Neurobiology*, 44:67–92, 2000.

[Weber *et al.*, 1996] K. Weber, S. Venkatesh and M V. Srinivasan. Insect inspired behaviours for the autonomous control of mobile robots. *Proceedings of the 13th International Conference on Pattern Recognition*, 1:156–160, Vienna, Austria, August 1996.