

Comparison of Temporal Filters for Optical Flow Estimation in Continuous Mobile Robot Navigation

Chris McCarthy¹ and Nick Barnes²

¹ Dept. Computer Science and Software Engineering
The University of Melbourne
cdmcc@cs.mu.oz.au
<http://www.cs.mu.oz.au>

² Autonomous Systems and Sensing Technologies Programme
National ICT Australia
nick.barnes@nicta.com.au
<http://www.nicta.com.au>

Abstract. We present our complete study involving comparisons of three spatio-temporal used in the estimation of optical flow for continuous mobile robot navigation. Previous comparisons of optical flow and associated techniques have compared performance in terms of accuracy and/or efficiency, and only in isolation. These comparisons are inadequate for addressing applicability to continuous, real-time operation as part of a robot control loop. In recent work [11], we presented a comparison of optical flow techniques for corridor navigation through two biologically inspired behaviours: corridor centring and visual odometry. In that study, flow was predominantly constant. In this paper we give new results from comparisons for flow-divergence based docking, where flow is non-constant. Results for traditionally used Gaussian filters indicate that long latencies significantly impede performance for real-time tasks in the control loop.

1 Introduction

Biologically-inspired visual behaviours such as corridor centring [5], visual odometry[17] and docking[14] have all been demonstrated using visual motion for closed loop control of a mobile robot. Despite encouraging results, mobile robot research has not broadly adopted this paradigm. A perceived lack of robustness, and the absence of any defined systematic approach to the implementation of such behaviours are likely reasons for this. The choice of optical flow method is perhaps the most important case in point. While visual motion is central to these behaviours, no systematic choice of optical flow technique currently exists.

Past comparisons have primarily assessed flow methods on accuracy and/or efficiency, and only in isolation [2,8,10]. These studies have not considered performance when embedded in a system, performing real-time tasks. Quantitative comparisons of accuracy and efficiency alone do not provide sufficient information on which to base a choice for mobile robot navigation. We emphasise the importance of in-system evaluation when comparing optical flow techniques for such an application.

In recent work [11], we conducted a comparison of optical flow techniques for corridor navigation. It was found that the choice of spatio-temporal filter applied

with gradient-based methods significantly effected in-system performance. Filters are used to reduce image signal noise before estimating image gradients. Temporal filters differ in required temporal support, latency and accuracy. It is important to consider temporal filters when evaluating flow methods for robot navigation.

In this paper we present our complete study in the comparison of temporal filters for robot navigation, including new results obtained for docking. We focus only on behaviours involving continuous motion, thus excluding such behaviours as hovering where motion changes are necessarily sharp and discontinuous. Corridor navigation comparisons examine performances when flow is constant or near constant. Docking comparisons examine performances when flow is not constant, but exhibiting increasing levels of divergence. The temporal filters for comparison are: Gaussian filtering with central differencing, Simoncelli’s matched-pair derivative filters [15], and Fleet and Langley’s recursive temporal filter [7]. These are applied with Lucas and Kanade’s gradient-based optical flow method [9], chosen on the basis of strong performances in [11]. We give an overview and theoretical comparison of these techniques before setting out our methodology for comparison. We then present results from on and off-board comparisons and then our conclusions.

2 Theoretical Overview

In this section we introduce all techniques and provide theoretical comparisons of the filters for robot navigation. Refer to the cited references for their full details.

Optical Flow Method: Lucas and Kanade[9] This method applies a model of constant velocity on small local neighbourhoods (ω) of the image by minimising:

$$\sum_{\mathbf{x} \in \omega} W^2(\mathbf{x}) ((\nabla I(\mathbf{x}, t) \cdot \mathbf{v}) + I_t(\mathbf{x}, t))^2, \quad (1)$$

where $W(\mathbf{x})$ denotes a window function. Thresholding eigenvalues of the least-squares matrix can improve accuracy, however, this was not applied.

Gaussian Filtering Gaussian filtering (std dev 1.5) and central differencing are the traditionally used filters for optical flow estimation. An isotropic Gaussian filter is applied in convolution for spatio-temporal pre-smoothing. A central differencing kernel (typically size 5) is then applied to estimate derivatives. We include two Gaussian filters with standard deviations 1.5 (Gaussian 1.5) and 0.5 (Gaussian 0.5). Central differencing (size 5) is then applied.

Simoncelli’s Matched-Pair Filters[15] Simoncelli proposed a filter design for obtaining accurate multi-dimensional derivative estimates using a small low-pass filter and derivative filter. These are related by their simultaneous design and applied as a matched pair through convolution. The implementation used here employs a size three pre-filter before applying the 5-tap matched-pair filters.

Recursive Temporal Filter[7] Fleet and Langley proposed a recursively applied causal temporal filter. Images are filtered via a cascaded implementation of an order n filter, where n is the number of cascades used. A time constant, τ^{-1} , gives the duration of temporal support. We use an order three filter ($n=3$, $\tau^{-1}=1.25$).

2.1 Theoretical Comparisons

To assist discussion of experimental results, theoretical comparisons are given below.

Accuracy: The Simoncelli filter by its design, is the strongest filter for angular flow accuracy [15]. Gaussian 0.5 accuracy will be low due to increased noise levels. The recursive filter is known to be less accurate on synthetic image sequences than the Gaussian 1.5 filter[7]. Of interest is how accuracy effects on-board performance.

Efficiency: For a 192x144 pixel image sequence, Table 1 shows computation times, storage requirements and latencies for all filters. Of the temporal filters, the Gaussian 1.5 requires the largest explicit frame support and frame delay. Frame delay is likely to influence in-system results significantly.

Robustness: Sensitivity to changing conditions is important, however, robustness to small fluctuations due to robot ego-motion is also desirable. The relatively large temporal support of the Gaussian 1.5, and the implicit support given to the recursive filter, suggests both should be robust to such noise. Simoncelli and Gaussian 0.5 may exhibit higher sensitivity due to their reduced temporal support. Increased noise levels with the Gaussian 0.5 are likely to further impede robustness.

Responsiveness: The recursive filter's large implicit temporal support may inhibit responsiveness to changes. Large frame delay with the Gaussian 1.5 will also impede responsiveness. Simoncelli and Gaussian 0.5 should exhibit high responsiveness given reduced frame support and therefore increased weighting on the current frame.

3 Methodology

In this section we present three navigational behaviours implemented for this comparison: corridor centring, visual odometry and docking. We outline the methodology and performance indicators used in our comparison of temporal filters.

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. Times taken on an Intel x86 866 MHz machine

3.1 Corridor Centring

Corridor centring, inspired by honeybee behaviour [16], is achieved by differencing average flow in the outer thirds of an image from a forward facing camera:

$$\theta = \tau_l - \tau_r, \quad (2)$$

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

Given constant motion and 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. Off-board comparisons over an image sequence depicting near constant motion of the camera can examine temporal cohesion. On-board trials in a static corridor can demonstrate the level of stability in robot directional control.

3.2 Visual Odometry

Distance travelled can be estimated by accumulating flow in the peripheral regions of the image. At a discrete time t , the visual odometer, d_t , is given by [17]:

$$d_t = \sum^t \frac{4}{\left[\frac{1}{\tau_l} + \frac{1}{\tau_r}\right]}. \quad (3)$$

Odometry estimates will vary in different environments for the same distance. In the same environment, however, the estimate should be repeatable. To compare methods, variance in average distances travelled can be examined for multiple on-board trials in the same environment. On-board trials are subject to oscillatory directional control, lateral drift, and environmental changes. To account for such in-system influences, off-board performances can also be examined.

A real image sequence with ground truth allows a quantitative comparison. If the distance measure d_t is repeatable, we expect it to differ by only a scale factor, s , from a ground truth visual odometer g_t . Under constant motion, this scale factor should remain approximately constant over time such that:

$$s = \frac{g_t}{d_t} = \frac{g_{t-1}}{d_{t-1}} = \dots = \frac{g_1}{d_1}. \quad (4)$$

3.3 Docking

Flow divergence is the measure of image expansion, given by:

$$div = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \quad (5)$$

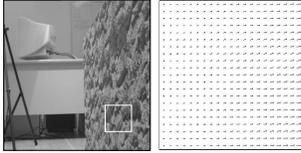


Fig. 1. Sample frame (boxed) and flow field from side wall sequence.

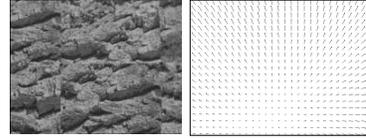


Fig. 2. Sample frame and flow field from looming wall sequence.

where u and v are components of a flow vector in the x and y directions respectively. It has been used extensively in the literature for obtaining time-to-contact estimates [12, 4, 1]. Docking can be achieved by reducing forward velocity in inverse proportion to increasing flow divergence. To maintain constant flow divergence on approach, forward speed is proportionally reduced.

Typically, divergence is calculated in one or more patches applied at the same image location over time. This assumes the looming surface is fronto-parallel, where in such a case divergence is constant across the image. During docking, minor rotations of the robot will lead to a lateral shift of the focus of expansion (FOE), which would considerably disturb the calculation of divergence if not accounted for. We therefore place divergence patches on a set radius originating from the FOE. The patches are above the FOE, at 45 degrees on either side. An approximately fronto-parallel docking surface is still assumed, however FOE tracking allows for small shifts giving more accurate divergence estimates.

Temporal filters must provide adequate responsiveness to rapidly increasing flow divergence as the wall approaches. Robustness to noise is also important. This can be assessed in off-board comparisons using a looming wall image sequence. On-board docking trials will assess system responsiveness, stability and reliability.

4 Off-board Comparisons and Results

For off-board comparisons, two real, heavily textured image sequences were constructed as shown in Figure 1 and 2. Figure 1 shows a side wall image sequence, depicting the motion of a wall moving 5mm per frame in a near parallel direction to the optical axis of the camera. Figure 2 shows a looming wall sequence. It was constructed by moving a camera 15mm per frame towards an approximately fronto-parallel wall. In both sequences, the visual motion with respect to the camera is subject to small frame-to-frame fluctuations. Ground truth flow fields were generated for all images in both sequences using a calibrated camera and a projective warping technique described in [10]. The model fitting technique RANSAC [6] was used to exclude outlying image correlation points. Comparisons are presented below.

4.1 Corridor Centring

Figure 3 shows average flow magnitudes obtained for each filter across the sequence. No latencies are accounted for, allowing a direct comparison of flow magnitude con-

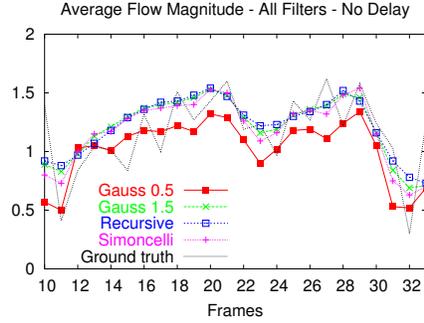


Fig. 3. Average flow magnitudes.

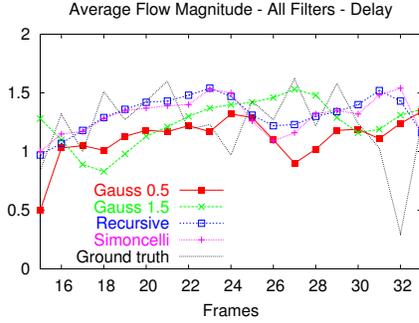


Fig. 4. Average flow magnitudes with latencies.

sistency across the sequence. The recursive and Gaussian 1.5 filters show near identical results. The Simoncelli filter performs slightly worse with sharper fluctuations most evident between frames 19 and 20, and frames 28 to 30. The Gaussian 0.5 exhibited the sharpest fluctuation between frames. Further distinctions are made when their respective frame delays are considered as shown in Figure 4. The larger temporal delay of the Gaussian 1.5 response is clearly evident.

4.2 Visual Odometry

Table 2 shows average scale factor errors ($av(s)$, where s is defined in (4)) and its variance (σ) for each filter against ground truth. Scale factor errors were calculated using the value of the ground truth visual odometer at each corresponding odometer update. Given constant motion, s should ideally remain constant over time.

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 (s_{av}). The recursive filter shows marginally less variance in s and s_{av} . The Gaussian 0.5 exhibits the most deviation on both metrics.

4.3 Docking

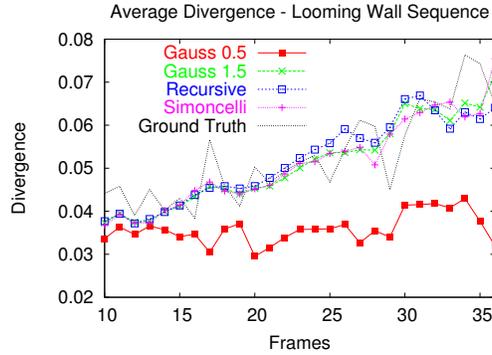
For off-board docking comparisons, the looming wall sequence was used. Divergence was calculated in two 40x40 pixel patches, centred on a 25 pixel radius from the FOE. Divergence was calculated for each pixel in both patches, then averaged to produce a final output value. Figure 5 shows divergences calculated at each frame, for all filters. All filters, except Gaussian 0.5, produce similar divergence growth.

4.4 Discussion

Across the side wall sequence, all filters except the Gaussian 0.5, exhibited similar consistency. Frame latency is likely to effect Gaussian 1.5 in-system performance.

Table 2. Odometry error analysis.

Filter	$av(s)$	$\sigma(s)$	$\sigma(s_{av})$
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

**Fig. 5.** Divergence over looming sequence.

Large temporal support, however, appears advantageous under near constant motion.

Visual odometry results give little distinction between filters. All filters appear likely to be reasonable. Some advantage may exist in having large temporal support (i.e. recursive and Gaussian 1.5), thereby reducing sensitivity to noise during motion.

In divergence comparisons, results indicated no lack of responsiveness to divergence increases in the Gaussian 1.5 or recursive filter, despite their large temporal support. With the exception of Gaussian 0.5, differences only become apparent from frame 28 onwards. Consistency appears to diminish in all filters at this point, particularly with the Simoncelli filter. At this point it is likely that flow is beginning to exceed measurable levels.

5 On-board Comparisons and Results

All techniques were integrated into the robot control software, running on an Intel x86 866MHz PC with radio link to a mobile robot. A forward-facing on-board camera was tethered to the PC. Frames were sub-sampled to 192x144 pixels, at a rate of 12.5 frames/sec. Robot tracking was achieved using a calibrated overhead camera. In all comparisons, surfaces were textured as in off-board comparisons.

5.1 Corridor Centring

Trials were conducted for each filter using a straight corridor, 2.5 metres in length, and 0.6 metres wide. Forward velocity was kept constant at $0.15m/s$ and rotation governed by a proportional control scheme. A proportional gain K_p , was chosen for each filter based on multiple trials in a straight and curved corridor. Due to varying filter latencies, a single K_p for all filters was deemed inadequate to accurately gauge relative performances. Figure 6 shows a sample on-board frame and flow field.

Table 3 shows average deviations from corridor center (centring error). The recursive filter achieves the lowest centring error. The Gaussian 1.5 achieves the

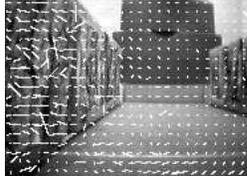


Fig. 6. Sample on-board frame and flow.

Filter	flow updates	centring error	std dev (stop pos)
Gauss 0.5	38	4cm	7cm
Gauss 1.5	35	6cm	10cm
Simoncelli	38	5cm	6cm
Recursive	45	3cm	3cm

Table 3. Centring and odometry results.

highest error. Notably, both Gaussian filters recorded collisions during trials. Figure 7 shows typical path plots for all filters.

5.2 Visual Odometry

On-board visual odometry trials were conducted using the same straight corridor and centring behaviour. For each filter, five 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. The final column of Table 3 shows the variance of stopping positions recorded for each filter. The recursive filter shows significantly lower variance than all others.

5.3 Docking

In docking trials, the robot approached an approximately fronto-parallel wall, attempting to decelerate and safely dock with the wall. Travel distance was 1 metre at an initial speed of 0.4 m/s. Forward velocity control was achieved using:

$$v_t = v_{t-1} + K_p(div_{ref} - div_t), \quad (6)$$

where v_t is the current velocity, K_p is a proportional gain, div_t is the current flow divergence and div_{ref} is the desired divergence which was set to 0.03 for all trials. For each filter, K_p was chosen to be the smallest value for which safe docking could be achieved four times consecutively.

Figure 8 shows typical velocities recorded with all but the Gaussian 0.5 filter for which docking was never achieved. A wide range of K_p values were used to verify this. Stopping distances were found to be similar for all filters. The Gaussian 1.5 exhibits a delayed response compared with other filters. The Simoncelli filter exhibited rapid deceleration, with brief periods of near constant velocity. Notably, no K_p value could be found for which the Simoncelli filter succeeded four times consecutively¹. The recursive filter shows early response and consistent deceleration.

5.4 Discussion

On-board results suggest the recursive filter is best suited to use in the control loop for continuous motion navigation. This appears a result of large temporal support and low frame delay.

¹ $K_p = 0.04$ achieved docking three times consecutively and so is shown in Figure 8.

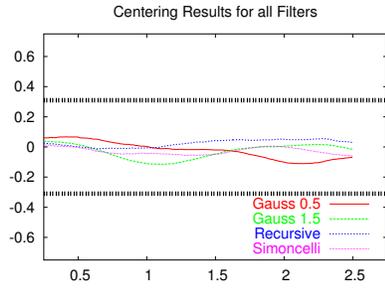


Fig. 7. Best straight corridor results.

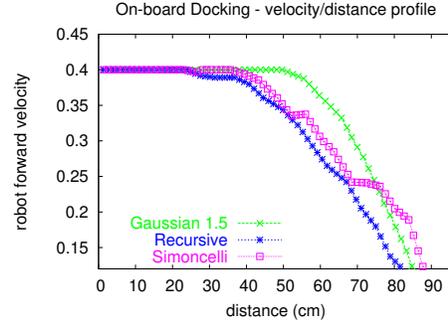


Fig. 8. On-board docking results.

In centring trials, the Gaussian 1.5 filters large frame delay seen in off-board results, appears to cause long phase oscillatory motion when centring. The Gaussian 0.5 centring error results appear reasonable, however several failed attempts were observed. Systematic noise and unresponsiveness appear the main causes of this.

Visual odometry trials exhibited more variation in results than off-board comparisons showed. The recursive filters stability in centring control, and high update frequency are likely reasons for its superior performance. The lack of stability in centring appears the likely reason for the poor performance of the Gaussian 1.5.

Divergence-based docking was reliably achieved using the recursive and Gaussian 1.5 filters. The Simoncelli filter exhibits less stable deceleration. This is possibly due to flow exceeding measurable levels as suggested in off-board results. Reduced temporal support will also heighten sensitivity to noise from large, unmeasurable flow. Future work will investigate this issue further. The Simoncelli filter maybe better suited to navigation under non-continuous motion, such as in hovering. Frame latency in the Gaussian 1.5 appears to have caused delay in divergence response. Where more controlled docking is required, it is likely the Gaussian 1.5 would perform worse than the recursive filter.

6 Conclusion

In this paper, we have presented on and off-board comparisons of temporal filters for gradient-based optical flow estimation in continuous mobile robot navigation. We have emphasised the need for in-system comparisons of vision techniques.

Over all comparisons conducted, the strongest performances were achieved using the recursive filter. Low frame delay and large implicit temporal support appear to be the main reasons for this. This is an encouraging result for the use of gradient-based optical flow in real-time, real-world conditions. Strong performances in corridor centring were also achieved using the Simoncelli filter. The Simoncelli filter appears better suited to navigation under non-continuous motion, such as hovering. Off and on-board results suggested the traditionally used Gaussian 1.5 filter is impeded by high frame delay, causing instability in centring and delayed response in docking.

Acknowledgement

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

References

1. N. Ancona and T. Poggio, "Optical flow from 1D Correlation: Application to a simple time-to-crash detector," *International Journal of Computer Vision*, 14(2):131–146, 1995.
2. J L. Barron, D J. Fleet and S S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, 12(1):43–77, 1994.
3. M. Bober and J. Kittler, "Robust motion Analysis," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 947–952, 1994.
4. 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, 1998.
5. D. Coombs and K. Roberts, "Centering behavior using peripheral vision," *1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 440–445, New York, NY, USA, 1993.
6. 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, 1981.
7. D J. Fleet and K. Langley, "Recursive filters for optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):61–67, 1995.
8. H. Liu, T H. Hong, M. Herman and R. Chellappa, "Accuracy vs. efficiency trade-offs in optical flow algorithms," *Computer Vision and Image Understanding*, pp 271–286, 1998.
9. B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision," *Proceedings of DARPA Image Understanding Workshop*, pp 121–130, 1984.
10. B. McCane, K. Novins, D Crannitch and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, 84(1):126–143, 2001.
11. C. McCarthy and N. Barnes, "Performance of optical flow techniques for indoor navigation with a mobile robot," *Proceedings of IEEE International Conference on Robotics and Automation*, pp 5093–5098, 2004.
12. R. Nelson and Y. Aloimonos, "Obstacle avoidance using flow field divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.
13. 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, 1995.
14. J. Santos-Victor and G. Sandini, "Visual Behaviours for Docking," *Computer Vision and Image Understanding*, 67(3):223–238, 1997.
15. E P. Simoncelli, "Design of multi-dimensional derivative filters," *Proceedings of 1st International Conference on Image Processing*, pp 790–794, Austin TX USA, 1994.
16. M V. Srinivasan and S. Zhang, "Visual navigation in flying insects," *International Review of Neurobiology*, pp 44:67–92, 2000.
17. 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*, pp 1:156–160, Vienna, Austria, 1996.