Progress Report: Adaptive Hybrid Tracking and Registration for Mobile Outdoor Augmented Reality

Samuel Grant Dawson Williams

September, 2012

Abstract

Outdoor tracking and registration are important enabling technologies for mobile augmented reality. Sensor fusion and image processing can be used to improve global tracking and registration for low-cost mobile devices with limited power and sensor accuracy. Prior research has confirmed the benefits of this approach with high-end hardware, however the methods previously used are not ideal for current consumer mobile devices. We propose the development of a hybrid tracking and registration algorithm that combines multiple sensors and image processing to improve on existing work in both performance and accuracy. Our research will target consumer level mobile devices and sensors. The design will also incorporate the ability to adapt its performance to the environmental conditions and user's task in an effort to balance usability with resource consumption. To evaluate the system, we will develop a set of synthetic quantitative tests to measure the improvements in accuracy, and also trial the system in a real-world application.

1 Introduction

Augmented reality (AR) is technology that allows virtual content (such as text, pictures, 3D models and sounds) to be blended with images of the real world[1]. Augmented reality systems have been an exciting research area for over 40 years. Recently the technology has been deployed on mobile devices[2, 3] and used in outdoor applications[4].

Typical outdoor augmented reality systems such as Layar[1], Wikitude[2] or AndroidAR rely on the global positioning system (GPS), compass and accelerometer information to provide position and orientation information. However, in practice these sensors often have a large degree of error and can be easily affected by local environmental phenomenon.

This project will explore two goals within the limitations of current consumer mobile devices: 1/ improving both global and local tracking using sensor fusion and optical flow, and 2/ accurate registration of content using computer vision techniques.

[1] http://www.layar.com
[2] http://www.wikitude.org

2 Background

Augmented reality systems deal with two fundamental technical challenges: 1/ tracking the camera's position and orientation in the real world and 2/ registering virtual object geometry with images taken from the camera.

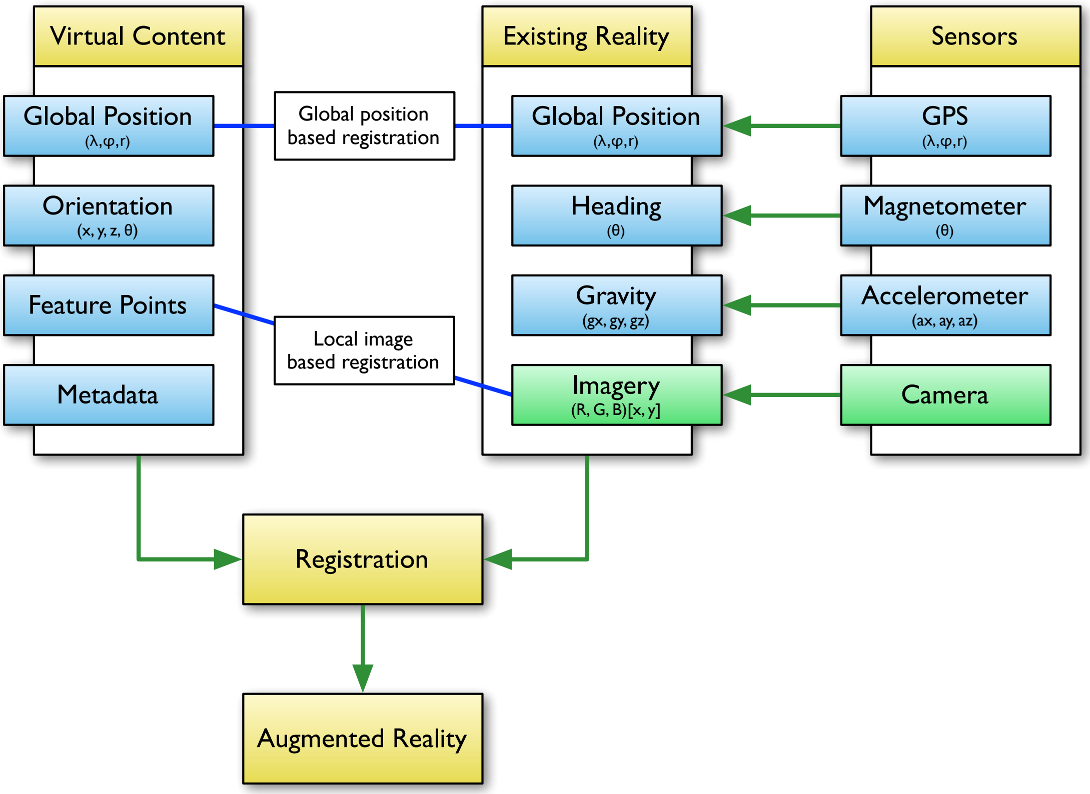

Figure 1: An overview of a typical augmented reality system, where virtual content and existing reality are combined together by establishing a mapping between the two forms of information.

The quality of outdoor augmented reality applications depends on the accurate tracking of position and orientation. A combination of physical sensors and visual data can be used to provide this information, however the organisation of the data processing and modelling will affect the performance and accuracy of the position and orientation measurements. In modern mobile devices, GPS (Global Positioning System) is the most common option for measuring global position along with a magnetic compass and accelerometer for orientation.

A significant amount of research has already gone into solving many of these problems, including how to deal with sensor drift using visual information[5], how to improve orientation estimation using natural feature tracking[6]. However, the majority of this research has been done using specialist hardware and with specific applications in mind.

Our research focuses on adapting existing approaches and designing new algorithms to improve outdoor augmented reality on modern mobile devices. In particular we will focus on developing an efficient version of global optical flow using motion estimates, named transform flow. This method will improve on existing feature point detection and extraction and combine many of the fundamental advancements made in a wide range prior research on sensor fusion and hybrid tracking.

For a thorough review of existing literature please see the original research proposal.

2.1 Expected Contributions

My thesis research is driven by the following key outcomes:

- A hybrid tracking algorithm utilizing sensor fusion and optical flow.
- A tracking algorithm that can be adapted to particular usage scenarios to improve performance and accuracy.
- A registration algorithm based on natural feature tracking and planar alignment.
- Implementation of the algorithm in a software library for the iPhone platform.
- A demonstration mobile outdoor AR application based on the software library.
- A written thesis which describes all the research work completed.

3 Work Completed

3.1 Ground Plane Detection (Late 2011)

One of the first ideas we explored involved tracking feature points on the ground plane. Due to the speed and accuracy of the accelerometer and gyroscope, calculating the gravity vector is fast and easy to do. Using this information, a bird's eye visual representation of the ground plane can be extracted. Using this transform, feature points can be tracked on the ground plane and mapped to changes in bearing and position using existing algorithms such as optical flow[7].

We implemented a basic ground plane extraction algorithm, but after investigating this method, we concluded that this approach wasn't viable in general. In many cases the accuracy of the camera wouldn't be good enough to track feature points (e.g. if the camera was pointing more than 45° up, the resolution of the ground plane is reduced significantly) and non-planar features caused significant artefacts in the final image.

This original implementation was running in real time and was hard to test. Therefore, we decided to focus on tools data acquisition and analysis.

3.2 Data Acquisition (Late 2011 - Early 2012)

There are few tools designed specifically for the capture and visualisation of of outdoor augmented reality sensor data and image frames. The most common platform, OpenCV[3] has a wide range

[3] http://opencv.willowgarage.com

of algorithms and several tools for visualisation but nothing specifically designed for visualising outdoor augmented reality data-sets.

In order to support the development and evaluation of new algorithms we decided to create data capture and analysis tools. These tools allow existing and new approaches to be tested in a controlled environment thus ensuring reproducability of results. Specifically, we decided these tools would support our quantitative evaluation including comparisons with ground truth data (e.g. measuring the absolute error against the expected results) and comparisons between algorithms (e.g. measuring the relative error and performance).

The first tool we designed was a mobile data acquisition system (see figure 2) which captures directly from an iPhone's sensors, including the gyroscope, accelerometer, gravity and video frames. The data sequence is recorded as a series of PNG images along with a structured CSV text file that records relevant sensor data. The use of a mobile platform ensures that we work within the capabilites of current mobile devices, which is an important goal of this research.

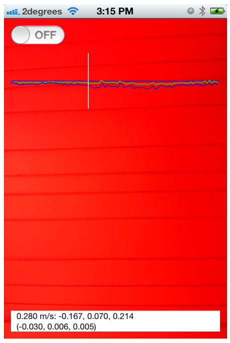

Figure 2: The data acquisition application running on an iPhone 4.

We captured several sample sequences and can easily capture additional sequences as required. The data-sets include different types of motion including horizontal and vertical rotations, lateral translations and forward translations and twisting.

The actual application builds on existing ARBrowser technology (developed in late 2010 - 2011) and can visualise sensor data using a real-time continuous graph. The graphs were helpful for sensor calibration and analysis as we could plot both accelerometer and gyroscope data in real-time.

3.3 Data Analysis (Early 2012 - Mid 2012)

After developing the data capture tool, we moved on to the analysis and algorithm design process. We designed a desktop application for analysing the captured data-sets and applying various different algorithms (see figure 3) and visualising the results. The tool was developed using an open-source graphics platform utilising OpenGL for rendering and OpenCV for analysis.

The data-sets we captured using the data acquisition tool can be visualised using sensor data to calculate frame alignment. We essentially use the gyroscope and accelerometer to calculate the 3D transform from one frame to the next. In addition to using sensor data, we have explored several image-based algorithms to improve the alignment, and we can visualise various processes such as keypoint detection, correspondence and alignment using markers overlayed in 3D.

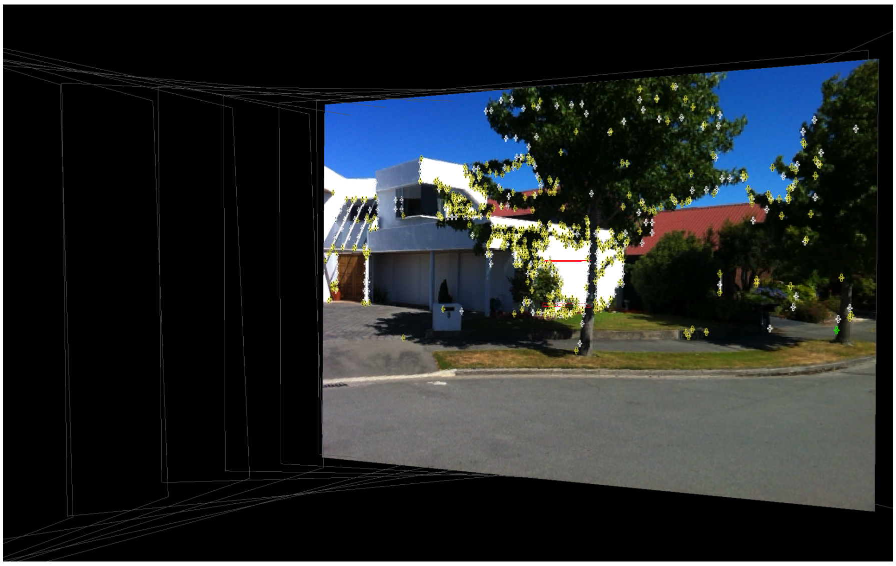

Figure 3: The data visualisation application running on a laptop computer.

In addition to analysing existing algorithms, we are using this tool as the primary test bed for the development of our ẗransform flowálgorithm. This tool allows us to visualise and test our approach repeatedly against the same data-sets which reduce the burden of debugging and evaluation.

In particular, this tool allows us to have an intuitive appreciation for the way a particular algorithm is working with a specific data-set, which in turn guides our decisions relating to the overall structure and approach we are taking with the transform flow algorithm. Through visual inspection of data processing and analysis we have developed a good understanding of the stengths and weaknesses of different approaches.

3.4 Transform Flow Implementation (Early 2012 - Late 2012)

Using the offline data analysis tools, work has begun developing a hybrid tracking algorithm. Our current approach extracts keypoints by using scanlines to detect gradiants, in some ways similar to a Harris Corner Detector[8], but improves on typical feature registration by using the gravity vector for consistency across frames.

Using sensor data, a motion estimate between sequential frames can be calculated. The motion estimate can include both changes in position and rotation and is internally represented as a 4x4 matrix. Using camera intrinsics along with motion estimates allows us to formulate a per-pixel correspondence between image frames, however due to sensor inaccuracies, visual features are not always aligned from one frame to the next. We seek to improve this pixel alignment by aligning visual features between frames using the motion estimate as a predictive guide as to where features may have shifted.

In order to find the actual transform that has occured, we use a series of scan-lines to detect edges in the image. These edges are then matched in the subsequent image by scanning along the 2D per-pixel transform as calculated from the motion estimate. The offset in pixels is used along with the initial motion estimate to produce an improved frame transform that ultimately results in better alignment of the two images.

The current keypoint algorithm uses the sum-of-squares error metric for determining alignment of visual features, but other more advanced approaches may be desirable at a cost of increasted processing time. Our approach also minimises the amount of data that needs to be copied per frame as opposed to other approaches that have large keypoint data-structures.

3.4.1 Feature Detection

Our alignment algorithm works based on the assumption that in typical outdoor augmented reality the change between frames is minimal and that it will be possible to detect at least several common edges from one frame to the next, as is typical in urban environments. We improve efficiency over existing visual feature alignment algorithms by reducing the amount of image data analysis required and integrating motion estimates into the feature correspondence search.

Typical feature alignment algorithms track corners from one frame to the next. In a typical implementation, the initial feature detection is used to initialise the database of feature points, and these are then tracked from frame to frame using a more efficient optical flow based approach. Optical flow cannot create new feature points in the database or recover points once they leave the field of view so it is typical to have the feature detection running in the background[9].

Similar approaches have been used in outdoor augmented reality, however our approach avoids the cost of feature detection by tracking edges which are comparatively easy to find. Edges, in practice, suffer from a number of problems when tracked from one frame to the next. However, we avoid these problems by using sensor data to estimate the change in position and rather than matching corresponding feature points from subsequent frames, we use a search algorithm to calculate the best alignment based on the motion estimate.

The primary result of this approach is that by depending on sensor data we can minimise the search space required to track subsequent frames significantly.

3.4.2 Performance

We have done some basic preliminary testing comparing our algorithm Transform Flowẅith Lucas-Kanade Optical Flow[7, 10, 11] using ORB keypoints[12] (see figure 1). We found that the algorithms corrected horizontally within ±1 pixel but that the cost of transform flow was significantly less.

Method	Y-Correction	Feature Detection	Correction
Transform Flow	-1.79167	0.003017s	0.000876s
LK Optical Flow + ORB	-1.72948	0.00733s	0.035261s

Table 1: These results are from initial testing in a synthetic environment.

In an offline testing environment there are no hard requirements on wall-clock time, however we would tend to prefer an algorithm that incrementally improves its result when allocated additional processing time, rather than an all-or-nothing solution. Therefore in our implementation we plan use an incremental approach which will sample only as many feature points as required to converge on a good solution.

Further analysis has revealed that there is a potential to reduce the overhead of the feature point matching algorithm used in transform flow as framerate increases. This is due to the fact that individual edges will transition fewer pixels for the same motion between frames at a higher framerate. This naturally means that the efficiency of the algorithm is tied directly to the speed of the motion that occurs and the number of feature points being tracked.

3.5 ST Project (Mid 2012)

As part of this research, I travelled to Singapore for three weeks to work on an augmented reality navigation application (see 4 for an example screenshot). Due to project requirements we were targetting iPhone hardware. We combined existing research from navigation and outdoor augmented reality to develop a dynamic navigation system that responds to the user's position and orientation.

The main navigation component relies heavily on GPS for positioning and the compass+gyroscope for orientation. This work provided useful insights into how existing technology and navigation systems operate and how they could be potentially improved using a hybrid computer vision based algorithms.

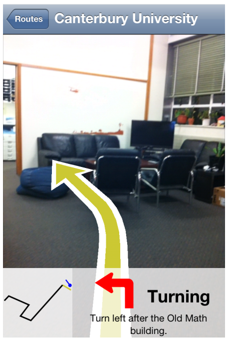

Figure 4: A screenshot from the Urban Navigation project developed as part of the ST Project.

3.6 Overview

The augmented reality arrows application has been designed to dynamically assist the user to navigate through a series of waypoints. The waypoints are generated offline or via map routing systems and record latitude/longitude coordinates in a specific sequence. The application presents location information in three ways: a large dynamic arrow that can bend to represent turns in the path, a mini-map with an orientation marker and a textural description (e.g. road name and distance from waypoint) which can also include iconic representation of the waypoint as appropriate.

The main concern for this type of visualisation is the ability to present information to the user in a continuous domain, such that a change in position or orientation produces a proportional change in the visualisation. In some specific cases of navigation (e.g. car navigation), it is possible to assume the user is always on a road heading in a specific direction. However pedestrian navigation is much more demanding as typically users may wander off course or choose a more convenient route. Thus, making assumptions about the user position is potentially challenging but can be simplified if the method of navigation is restricted.

The first design used globally registered 3D arrows positioned periodically in the space between the waypoints. This design assumed accurate global registration and as such the arrows could be easily visualised by computing the position of the arrows relative to the user. All visible path segments would be displayed which could lead to visual clutter and if the user was travelling at speed, arrows would move by the user very quickly which could be disconcerting. Finally, the accuracy of global registration was a problem: typically the user would not be located directly on the path segment and thus the visualisation would not align up with the user's actual movement.

To improve upon these issues, we decided to use a single locally registered arrow. Initially we positioned the arrow perpendicular to gravity. However, we found that in this case the arrow may not be visible if the user isn't holding their device correctly. Because of this, we used a 45° fixed incline relative to the screen so that the arrow would always be visible. In addition the arrow points the user towards the next waypoint at all times, so regardless of the users current position, meaningful and continuous navigation information is presented. When the user is in range of a turning point, the arrow will bend in the direction of the turn to assist the user in turning onto the next path segment.

3.7 Bending Arrows

The bending arrow visualisation responds dynamically to the user's position and bearing relative to the next waypoint. A state machine was implemented to process changes in location and produce relevant data for the visualisation. There are two major phases to consider: when the user is travelling between waypoints and when the user is travelling through a waypoint (turning).

Travelling between waypoints is trivial in general and it is assumed that the user will travel sequentially from one waypoint to the next. In the case of significant course deviation the current path segment will be recomputed based on position; in certain applications it may be useful to take into account the direction of travel (e.g. if you have parallel path segments in opposite directions in the case of a 180° turn).

The turning state is activated when the user enters a sphere centred at the waypoint with a specified turning radius. To exit the turning state and thus transition to the next segment the user must travel out of the circle and towards the next path segment. The practical consequence is that we expect users to travel through specified waypoints and that significant deviation is not possible. If the user does miss the waypoint they will eventually cause the state machine to recalibrate based on the closest path segment.

For vehicle based navigation this may be a good approach, but for pedestrian navigation it may be too rigid. To relax these constraints, a proportional scheme using an ellipse may be more appropriate where the aspect ratio is proportional to the angle of the turn. In addition, while in typical vehicle navigation a 180° turn may only be possible at a certain point, for a pedestrian, such a waypoint has little meaning (they can reverse direction at any time). In this case, some types of arrow visualisations may be best calculated taking into addition the users orientation and direction of travel.

The actual arrow visualisation currently implemented depends primarily on two variables: the incoming bearing and the outgoing bearing. While travelling between waypoints, both the incoming and outgoing bearing are set to the heading towards the current waypoint. When turning, the incoming bearing is calculated as the heading from the previous waypoint to the user location and the outgoing bearing is calculated as the heading from the user location to the next waypoint (see figure 5).

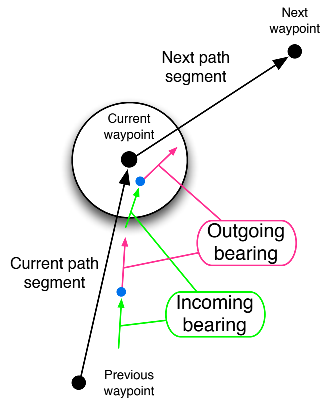

Figure 5: Bearing calculations both between waypoints and passing through a waypoint.

To minimise the abrupt change in bearing that would occur as the user transitions into a turn, we model the turning ratio as a factor going from $-1 \rightarrow +1$ (see figure 6) which is used to interpolate the change in bearing as the user moves through the turn. The simplest way to calculate the turning ratio is to plot the user's location in a circle centered at the waypoint of a fixed radius.

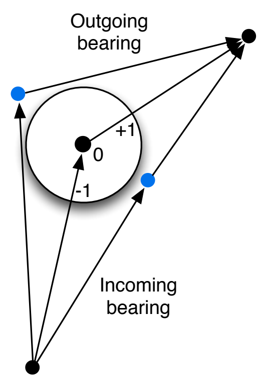

Figure 6: Bearing calculations both between waypoints and passing through a waypoint.

In addition to the above interpolation, we incorporate the orientation of the user into this calculation so that the bending calculation is responsive to the turning behaviour of the user. This ensures that as the user moves through the turn that the bending arrow responds rapidly to the user's change in bearing. This method has been robust to errors in user position while mapping the transition across a consistent domain.

3.8 ColAR (Mid 2012 - Ongoing)

As part of a commercial company grown out of HIT Lab NZ research, we have been implementing a typical natural feature tracking algorithm for planar based registration. While the goals of this work are significantly different from typical outdoor augmented reality, the algorithms and their use overlap in some areas and it has been helpful to look at existing natural feature tracking algorithms and their efficient implementation.

The main application ColAR is designed to overlay 3D content on 2D planar pages in colouring books. The initial registration step analyses feature point correspondence between a known 2D planar image and the input from the camera (see figure 7). Not all features match correctly, so RANSAC is used to refine the set of features by assuming planar correspondence.

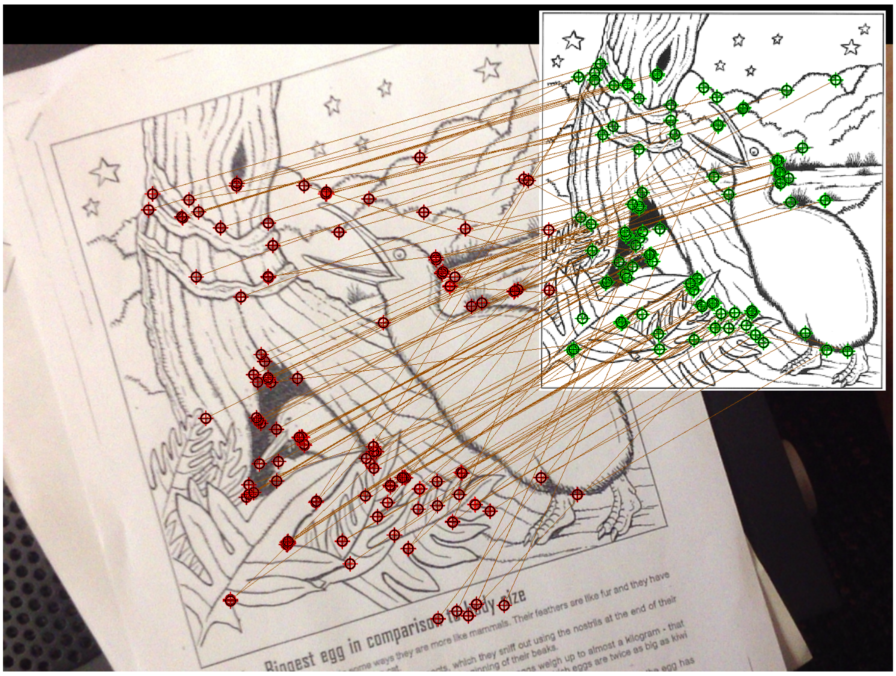

Figure 7: Feature point correspondence using ORB binary feature points and FLANN matching.

Corresponding feature points are used to extract 3D translation and rotation, which allows virutal content to be aligned with planar surfaces in the real world.

Natural feature detection and description algorithms are typically expensive. Our implementation of ColAR uses a similar approach to OPIRA[13], but uses ORB feature point detection and descriptors[12]. This approach uses a two step process where initial registration is done using natural feature recognition and subsequent frame-by-frame registration uses optical flow to track feature points efficiently.

4 Challenges

The biggest challenges relating to the development of our new algorithms are due to the nature of the data we have available and the inherent error in these measurements. Sensor fusion alone gives reasonable results which have fairly consistent error conditions (e.g. drift, positional errors). Incorporating visual information to improve upon these calculations means that we are trying to reduce the overall error of the measurements being made. However in many cases, combining different algorithms can have an overall detrimental effect on the worst case error which we want to avoid.

As an example, using optical flow to increase the speed and accuracy of changes in orientation may be computationally expensive while doing nothing to reduce the absolute error in the initial compass measurements. In typical sensor fusion, a low-pass filter would combine compass and gyroscope data[5]. Over time, the results should converge to a minimum error while still being responsive to changes in physical disposition. However, instantaneous changes in sensor data (including visual data) will be unable to correct incorrect absolute measurements and additional filters may reduce the overall effectiveness of the total combination.

Another challenge involves the analysis of 3D motion. Translation and rotation are difficult to separate and incorporating image-based processing in some circumstances increases the level of uncertainty. For example, using optical flow it is possible to detect translations and rotations based on global field vectors, but in certain circumstances visual information can be misleading e.g. a moving car in the field of view might make it appear as if there has been a global translation.

5 Work Scheduled

The current algorithm is designed specifically to proove the viability of the approach and is not a general purpose implementation at this time. Several options exist for future work including expanding our current implementation to incorporate more features or to focus on fine tuning the existing implementation to solve some of the fundamental issues. In addition, several lateral branches of research have become apparent and thus could be good candidates for future work.

Our next step will be driven by the need to solve the key research problems which involve feature tracking and optical alignment. In particular, bearing measurements incorporate both global and local sensor measurements (compass and gyroscope). Computer vision techniques such as the one we've already implemented using the vertical line feature points can also provide very accurate rotational information. We will refine the existing implementation and then combine the three data-streams together using a statisitical model. This will allow us to evaluate our approach against existing solutions in terms of accuracy and efficiency, while still touching on all the important research goals.

Once we have confirmed the practical viability of the transform flow algorithm, we can continue to develop the algorithm for full rotational and positional tracking.

5.1 Refining Transform Flow

Initially we need to improve the full frame motion analysis by analysing multiple feature points and using a statistical model to combine these data-points with sensor data. Existing filters such as Kalman filter and Particle filter may proove useful in this area along with potentially simpler combinatory filters (e.g. low pass filters).

Improved optical alignment based on better feature detection (e.g. as per natural feature tracking) is another area that could significantly improve the quality of the prediction. We need to explore this area and determine if it is viable and look at the performance of such an approach.

5.2 Local - Global Registration

Changes in bearing can be accurately measured using the gyroscope, but if the original bearing calculation is incorrect, the updated bearing will be equally incorrect. We have referred to this problem as the local - global correspondence problem. One appraoch to reduce the error over time involves modelling error measurements and correlating the data over time[14].

We would like to look at the complexity of such and approach and whether it has a significant positive effect on the practical application of the transform flow algorithm.

5.3 Hybrid Tracking

Combining existing sensor fusion with image processing presents a unique opportunity to analyse motion beyond either approach alone. We are investigating the practical nature of this hypothesis and how we can utilise it to improve existing sensor based approaches which are typically very poor at measuring translation accurately.

One such option would be a per-frame motion checksum. The checksum algorithm could be designed to efficiently detect visual or sensor motion. If motion is detected, further processing could be done to calculate and refine the motion. If no motion is detected, no such processing would need to take place. Relative sensors, such as the gyroscope, that exhibit drift are potentially a good candidate for this type of correction[4].

When using optical flow alone, it can be difficult to measure the difference between rotation and lateral translation. It might be possible to improve on optical flow based analysis by incorporating the gyroscope to isolate translation from rotation. If optical flow indicates a left or right translation in the visual information, the gyroscope could be used to confirm whether this was a rotation or translation.

5.4 Feature Descriptors

The current implementation finds edges and uses these for alignment. We do not pay attention to local morphological structure, however it has become apparent that tracking more than just a feature point may improve the general reliability of tracking. Tracking a set of connected edges, or an otherwords a line, may significantly improve the reliability over purely feature point tracking (e.g. sub-pixel accuracy). Our existing implementation could be extended to incorporate this type of tracking and we could evaulate whether it has an overall beneficial effect.

A second option in this area is to try to compute a local edge-normal vector which could be used to infer the quality of the edge with relation to the motion estimate detected. Edges which are parallel to the detected motion are less useful than those that are perpendicular. This analysis could improve the efficiency of the overal algorithm by discarding feature points that wouldn't add to the overall end result.

Another area of interest involves refining feature points based on a local maximising search. Once an edge is found, we could scan along both directions of the edge to find local maxima (e.g. corners). While this approach might not be useful for transform flow, it might give very consistent frame-to-frame feature correspondence which in itself would be a useful result.

6 Summary

At this time, we will focus on the transform flow algorithm and developing a practical working implementation. Additional areas of research have been identified which we have documented and will review again in the future. Several practical and commerical projects have been implemented which have given us an excellent understanding of the types of problems we are trying to address. We will continue forward with our research and aim to have the master's thesis completed early next year.

7 Work Schedule

- Refine feature detection for use in finding and tracking vertical lines.
- Design a statistical model for frame-to-frame rotation analysis using feature tracking output.
- Combine statistical model with data from compass and gyroscope.
- Compare performance and correctness with ground truth data and Lukas-Kanade optical flow.
- Publish the computer vision software tools online.
- Combine proposal and progress report into initial draft thesis.
- Write a detailed summary of how the algorithm works and how it was implemented.
- Write a detailed report on how the algorithm performed.

8 Publication Schedule

- 3DUI Poster Competition; December 8th, 2012 deadline[4].
- ISMAR; April 2013 deadline, Conference November 2013: Conference paper regarding results, possible collaboration with ST project team.
- IVCNZ; September 2013 deadline, Conference in December 2013: Conference paper regarding results.
- IEEE; Virtual Reality 2013, Conference in March 2013. Conference paper regarding research and results.
- Condense the final thesis into a journal paper.

[4] http://www.3dui.org/cfp-posters

References

[1] Ronald Azuma, Bruce Hoff, Howard Neely Iii, and Ron Sarfaty. A motion-stabilized outdoor augmented reality system. In *Proceedings of the IEEE Virtual Reality*, VR '99, pages 252–, Washington, DC, USA, 1999. IEEE Computer Society. Available from: http://dl.acm.org/citation.cfm?id=554230.835680.

[2] Mathias Mohring, Christian Lessig, and Oliver Bimber. Video see-through ar on consumer cell-phones. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 252–253, Washington, DC, USA, 2004. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/ISMAR.2004.63.

[3] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 125–134, Washington, DC, USA, 2008. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/ISMAR.2008.4637338.

[4] Bolan Jiang, Ulrich Neumann, and Suya You. A robust hybrid tracking system for outdoor augmented reality. In *Proceedings of the IEEE Virtual Reality 2004*, VR '04, pages 3–, Washington, DC, USA, 2004. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/VR.2004.5.

[5] Gerhard Schall, Daniel Wagner, Gerhard Reitmayr, Elise Taichmann, Manfred Wieser, Dieter Schmalstieg, and Bernhard Hofmann-Wellenhof. Global pose estimation using multi-sensor fusion for outdoor augmented reality. In *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '09, pages 153–162, Washington, DC, USA, 2009. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/ISMAR.2009.5336489.

[6] Daniel Wagner, Alessandro Mulloni, Tobias Langlotz, and Dieter Schmalstieg. Real-time panoramic mapping and tracking on mobile phones. In *Proceedings of the 2010 IEEE Virtual Reality Conference*, VR '10, pages 211–218, Washington, DC, USA, 2010. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/VR.2010.5444786.

[7] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. Available from: http://dl.acm.org/citation.cfm?id=1623264.1623280.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[9] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/ISMAR.2007.4538852.

[10] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[11] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Ithaca, NY, USA, 1993.

[12] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society. Available from: http://dx.doi.org/10.1109/ICCV.2011.6126544.

[13] Adrian Clarke. *OPIRA: The Optical-flow Perspective Invariant Registration Augmentation and other improvements for Natural Feature Registration.* PhD thesis, University of Canterbury, 2009.

[14] Tom Botterill, Steven Mills, and Richard Green. Bag-of-words-driven single camera simultaneous localisation and mapping. *Journal of Field Robotics*, 2010. Available from: http://www.hilandtom.com/tombotterill/Botterill-Mills-Green-2010-BoWSLAM.pdf.