•

AP 068

VR|AR Headset with Kalman Filter Position Tracking and Stereo Vision Depth Sensing

Team members ： Chong Kwang Liang, Lim Jia Zhi, Yong Kah Men
Organization ： Universiti Tunku Abdul Rahman (UTAR)
Instructor ： Dr Khaw Mei Kum, Dr Lo Fook Loong

I. High-level Project Description

Virtual Reality (VR) and Augmented Reality (AR) are predicted to be the important trends in the near future with more and more new applications emerging as VR/AR becoming increasingly present in daily lives. VR/AR will step towards mainstream adoption of new technology and change our interfaces to digital devices forever.

Virtual Reality is where the user’s senses is presented with a computer generated virtual environment. Virtual Reality is typically experienced through head-mounted display (HMD) which display two streams of slightly different video feed for each eye to simulate the user’s senses and create an illusion of reality. Augmented Reality is another form of VR which combine virtual information with real time camera feed, by overlaying graphics onto the view of real world.

One of the main goals of this project is to implement a lightweight, low power consumption sensor hub to be integrated with the head mounted display to realize a cordless battery head mounted display to improve on the user experience.

1.1 Position Tracking

Position tracking plays a vital role in VR/AR application in order to simulate an immersive virtual environment. Positional tracking raw data from IMU unit on headset is fed into FPGA for processing using the Kalman filter algorithm. Kalman filter is an algorithm that uses a series of data collected to estimate a more accurate output. As the filtering algorithm is computationally intensive, FPGA is able to optimize the computation through parallelism in order to minimize the processing latency to make the experience as immersive as possible.

1.2 Eye Tracking

Eye tracking is the measurement of eye activity such as eye positions tracking and eye movement tracking to sense the point of gaze. Eye tracking enables interaction and control of the device more naturally by using only eyes as an input device. This allows a more natural way of performing action than with mouse or touchpad. The eye tracker make use of image processing algorithms to interpret the image stream captured by the hardware and calculate the eyes position and movement. FPGA can help to improve the latency and power consumption in the computation of image processing algorithm which can take advantage of the parallelism nature of FPGA.

1.3 Stereo-Vision Depth Sensing

Stereo-vision is the extraction of 3D information from images by extracting information from two 2D images into 3D depth map. This can substantially improve accuracy of surrounding condition estimation which is essential in AR. Dual vision depth sensing require pipeline of processes which are noise filtering, distortion correction, stereo matching, outlier detection, subpixel depth interpolation. The processes can be accelerated by implementing parallel processing in FPGA, allowing fast generation of dense depth map. The depth map will then processed by processor for surrounding estimation, object recognition and tracking.

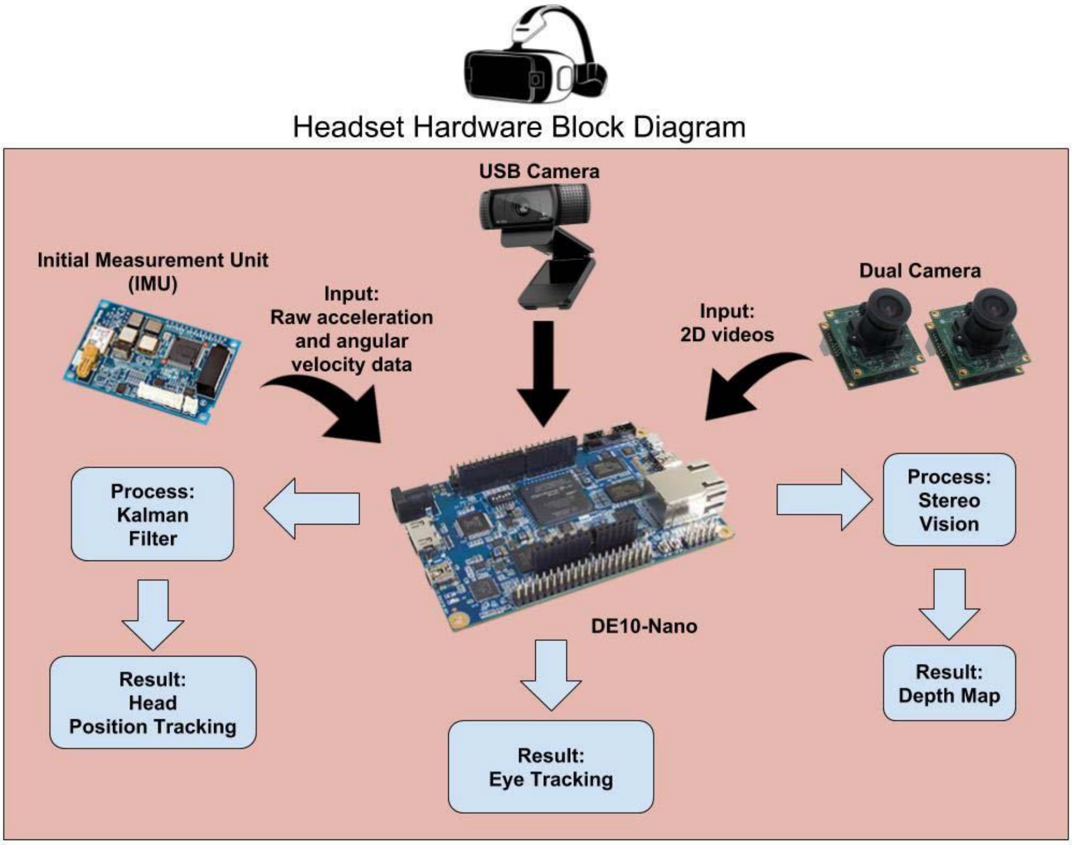
1.4 Application and Targeted users

The main targeted users of this project is the maker crowd, VR/AR developer and also students. The project objectives is to mainly provide a sensor hub to be integrated on the VR/AR headset and also to provide an API to facilitate the reading

of read raw data.

II. Block Diagram

Figure below shows the overall hardware block diagram of our system which consists of three main parts, which are head position tracking, stereo-vision depth sensing and eye tracking.



III. Intel FPGA Virtues in Your Project

By combining the flexibility and high level of logic integration of gate arrays and the benefits of the convenience, ease-of-use and shorter design time, Intel FPGAs have become a mainstream technology for digital system design, especially through the use of FPGAs as reconfigurable computing machines.

Kalman filter is a complex algorithm where complex matrix computation is required.

Intel hardware architecture FPGAs will have advantages in resources usage and throughput compared to conventional implementation of this algorithm. With low latency requirement for position tracking in VR/AR, a fast processing speed is required where Intel FPGA able to achieve it.

The repetitive operations of image processing in the application of stereo vision and eye tracking can causes delay in output, as real-time processing is too intensive for a low-power CPU. FPGA allows parallel image processing in generating real-time output while consuming less power.

IV. Functional Description

4.1 Stereo vision

Stereo-Vision processing first will run on FPGA for preprocessing and processor for processing. Camera video feed is stream in into FPGA first for distortion correcting and filtering. Then, the corrected images will be feed into stereo matching circuit, combining information from two corrected 2D images into 3D depth map. Finally, the depth map will be processed by hard processor system for object recognition and tracking.

Stereo vision extracts 3D depth information from digital images captured from two cameras. The two camera are placed horizontally some distance apart from one another.

Two different vantage points of a scene is obtained in an arrangement similar to human binocular vision. By comparing these two images, the depth information can be obtained in the form of a disparity map. The relative depth position of objects can then be calculated when the distance between the two cameras and the focal length of the cameras are known. The steps of stereo vision are as follow:

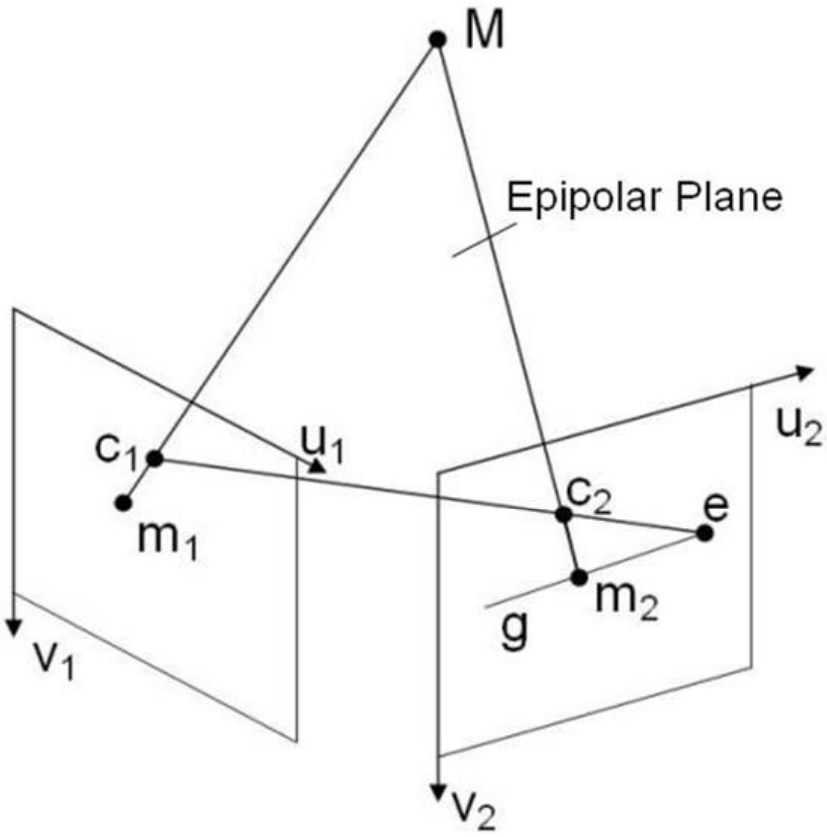
1. Calibrate Camera and Rectify Image
   - Offline camera calibration by using a simple planar grid pattern of known size to align images to compensate hardware misalignment of camera mounting. Moreover, image input from camera are rectified to remove distortion of camera image and making pairs of conjugate epipolar lines in two images to become collinear and parallel to one image axes for local matching.

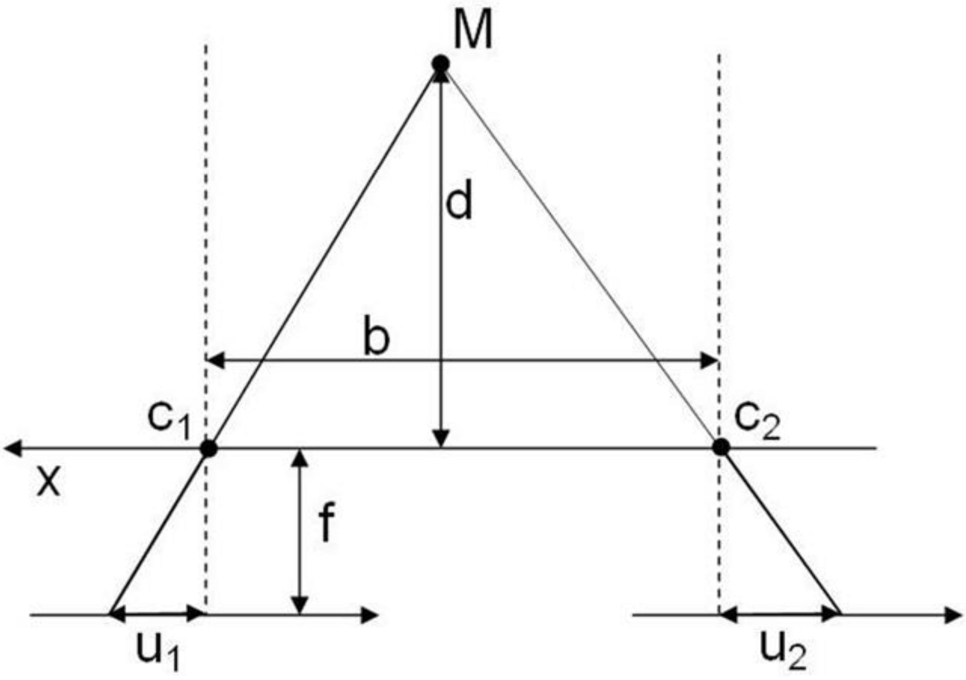
2. Grey-scale conversion

3. Stereo Matching
   - Local matching algorithm is used to obtain disparity information as it has low complexity, and suitable for real-time stereo vision computation. Fixed window Sum of Absolute Difference (SAD) algorithm is used for cost aggregation. SAD computes intensity differences for each pixel (i, j) in a window with the size of vx and vy as the equation:
   - Then, winner-take-all(WTA) strategy is used to find the disparity with highest aggregated cost. Local matching algorithm is known to be easily affected by the uneven illumination and occlusion. Hence, occluded pixels near object borders will be removed by left-right consistency check.

4. Disparity Calculation
   - From the disparity (D) computed, distance will be calculated from the Equation

$$d = b\frac{f}{D}$$





- D = u2 - u1
- d = distance
- b = baseline (distance between optical centres of cameras)

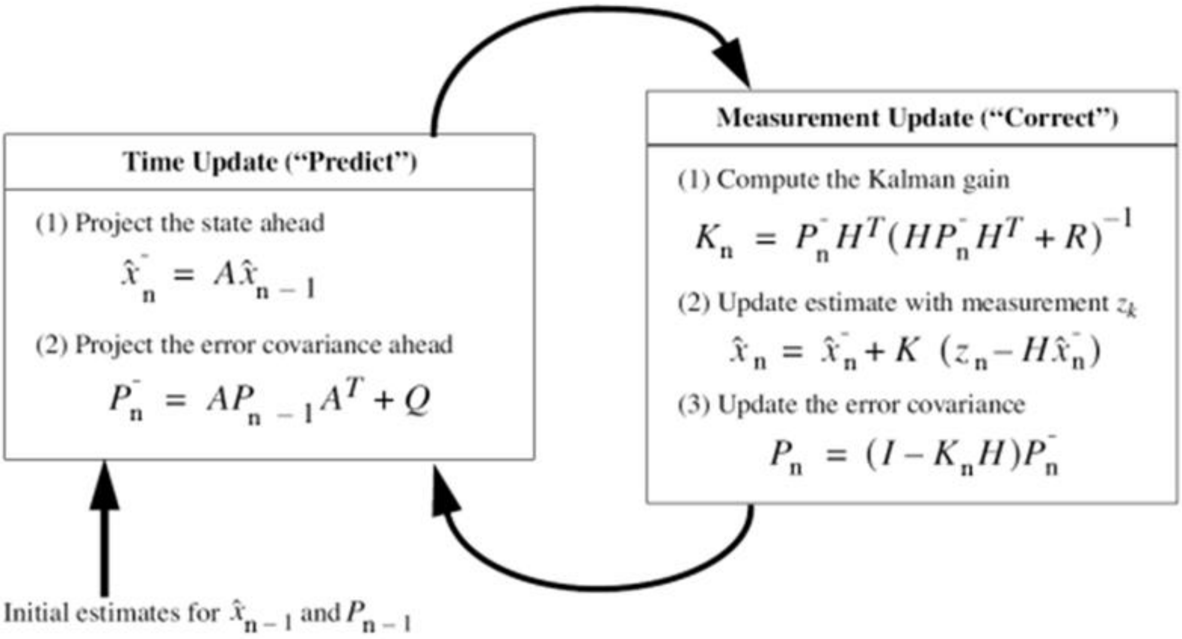
- D = disparity
- c = optical center
- f = focal length
- g = epipolar line
- Point M is projected as m1 in left image plane, the corresponding point can
- also be projected as m2 in right image.

4.2 Kalman Filter

A Kalman Filter is a statistical algorithm used to compute an accurate estimate of a signal contaminated by noise. It uses measurements observed over time to reduce noise and produce estimates of the true values, and estimate other parameters related to the signal model. The locator uses multiple input signals which when alone, can each be used to determine location of a device. Depending on different circumstances, some inputs are more contaminated with noise than others. The Kalman Filter uses all the available information from all of the input signals to minimize the noise and determine the best estimate for the location of the device.

The Kalman filter operates by producing a statistically optimal estimate of the system state based upon the measurement(s). To do this it will need to know the noise of the input to the filter called the measurement noise, but also the noise of the system itself called the process noise. To do this the noise has to be Gaussian distributed and have a mean of zero, luckily for us most random noise have this characteristic.

The Kalman filter is essentially a feedback estimation system. There are two steps that the Kalman filter takes. The first is to calculate a prediction of the state. Next, feedback is given to the filter in the form of measurements, which are of course noisy and not perfect. Therefore, the filter is split into two parts: the time predictions and the measurement updates. The Figure below shows complete Kalman Filter's equation diagram.



It is a powerful and widely applicable system, however it does have some requirements which can make it difficult to implement. It is a recursive filter, which means that any system implementing it requires the inputs to wait for the outputs, which can take a lot of computing time on a traditional sequential computing system. Also, it is not unusual for there to be a large block of inputs to the system. The system is able to model the internal state of large and complex system, therefore one expects there to be a large number of variables involved in the processing. This process may be very slow on a traditional computing system.

By utilizing the FPGA environment, the inherent parallel computation ability of the

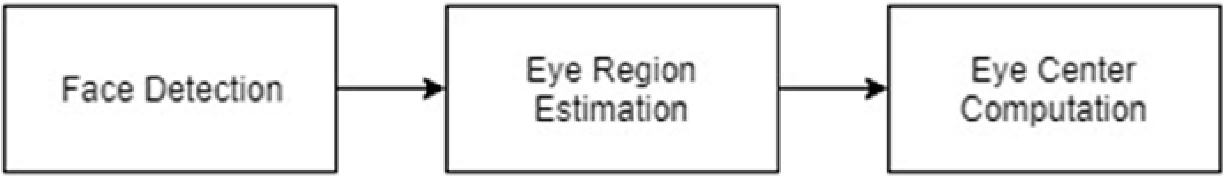
Kalman filter is exposed and exploited for maximum computing performance. The way to leverage this ability is to take advantage of the many computing resources available on an FPGA. FPGAs come with blocks of hardware resources such as DSP units, embedded memories, complex clocking structures, etc, which can all be used to further the main overall goal: to make the system as fast and seamless as possible. Plus, by utilizing an FPGA system, there is no need to send the hardware description to a foundry to have to produced in silicon. The FPGA can be configured by the end user as needed.

4.3 Eye and Gaze tracking

Eye tracking measures the motion of an eye and he point of gaze. An eye tracker webcam is employed in measuring eye positions and eye movement. The eye tracker is not able to provide an absolute gaze direction, but only an approximate changes in gaze direction.

Multi-stage image processing approach is chosen as no additional hardware equipment such as IR LED or IR camera used in a typical infrared eye tracking system is needed.

A multi-stage approach is taken for the centre localisation. A face detector is first applied as the eye is position is always at the upper region of the face, the rough eye region can be extracted. Precise eye center is estimated based on a simple image gradient-based eye center algorithm by Fabian Timm where a simple objective function based on dot product is derived and the centre of the pupil can be found at where the objective function yields a strong maximum. Then, a simple calibration is performed by looking in each direction to calculate the threshold of each gaze direction before using the eye tracker.



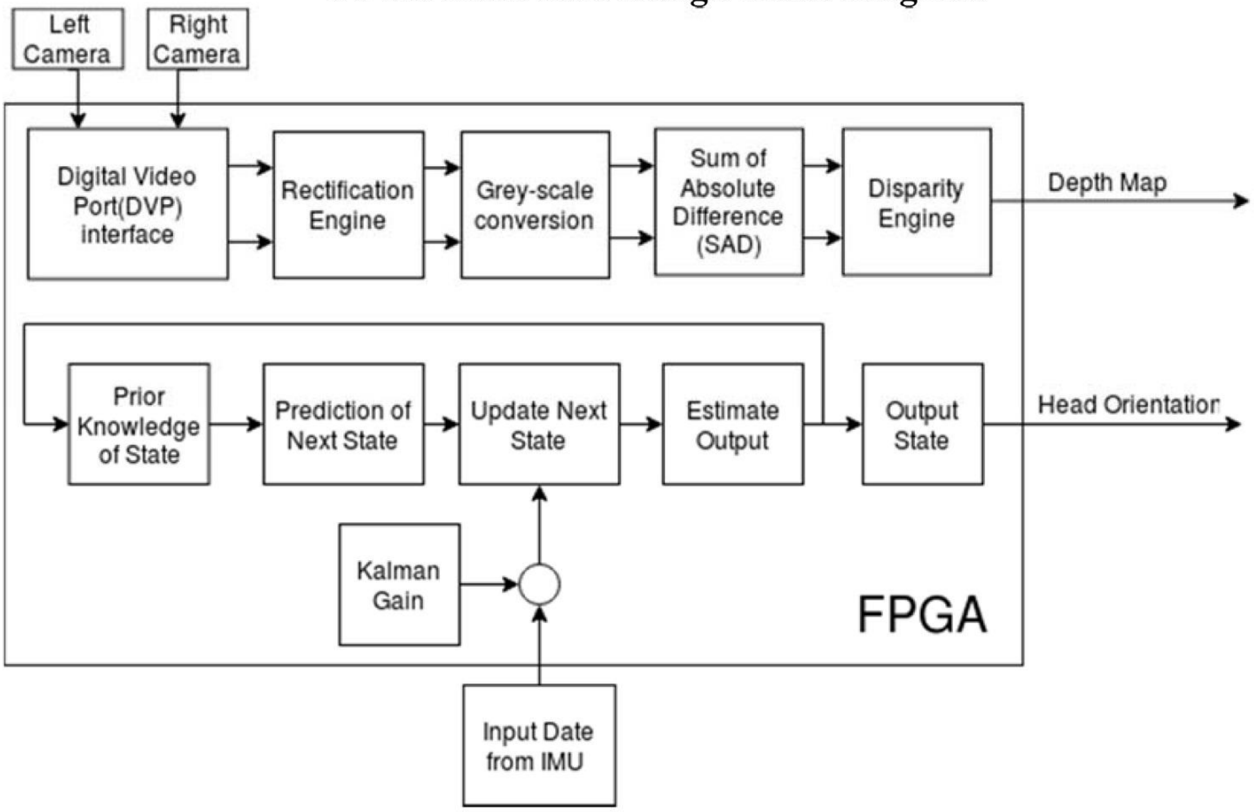
V. Performance metrics / goals

The project uses 3 module to compute the desired result concurrently. The computation time for each module block is measured by using software implementation on HPS alone as a baseline and then compared with expected FPGA implementation time:

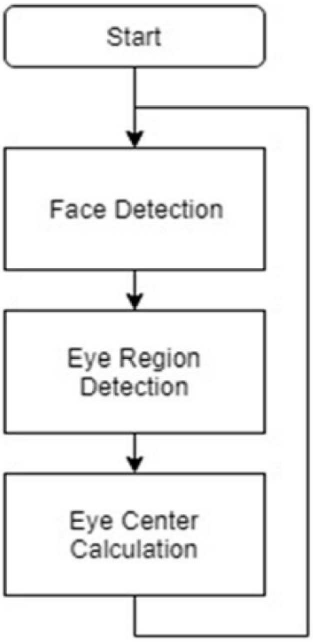
Process	Software Implementation	FPGA Implementation
Eye Tracking	1.6 s	~ 33.3 ms (30 fps)
Stereo Vision	2 s	~ 65 ms (15 fps)
Kalman Filter Head Tracking	-	1 us

VI. Design Method

FPGA Hardware Design Block Diagram



Eyes Detection Algorithm Flow



VII. Conclusion

In conclusion, a sensor hub which collect the raw sensor reading and perform some pre-processing on the sensor data is proposed in order to give a more user friendly data representation to any host device. The data provided including head posture data, gaze direction and depth map of surrounding in front of the headset which is the crucial data for the application of VR/AR. The host device can then just call the APIs to obtain the required sensors data from our proposed device. The collected data will be send out to another host computing device. This can lower the computing power required and provide a hardware abstraction layer to the AR/VR developers. The pre-processing of data can also eliminate the need of a host computing device.

VIII. References

Altera.com. (2014). *Extended Kalman Filter*. [online] Available at:

https://www.altera.com/content/dam/alterawww/

global/en_US/pdfs/literature/ds/extended_kalman_filter.pdf.

Altera.com. (2014). *Stereo Vision IP Suite*. [online] Available at:

https://www.altera.com/solutions/partners/partner-profile/fujisoftincorporated/

ip/stereo-vision-ip-suite.html.

Bhatia, Z. (2012). *On Kalman Filter Implementation on FPGAs*. [online] University of

Victoria. Available at:

https://dspace.library.uvic.ca/bitstream/handle/1828/4365/Bhatia_Zorawar_MASc_2

012.pdf?sequence=3&isAllowed=y.

Hume, T. (2012). *Simple, accurate eye center tracking in OpenCV*. [online] Thume.ca.

Available at: http://thume.ca/projects/2012/11/04/simple-accurate-eye-centertracking-

in-opencv/.

Lee, C. and Salcic, Z. (1997). High-performance FPGA-based implementation of

Kalman filter. *Microprocessors and Microsystems*, 21(4), pp.257-265.

Mattoccia, S. (2013). Stereo Vision Algorithms for FPGAs. *2013 IEEE Conference on*

Computer Vision and Pattern Recognition Workshops.

Pizzini, E. and Thomas, D. (2012). *FPGA Based Kalman Filter*. [online]

WORCESTER POLYTECHNIC INSTITUTE. Available at:

https://web.wpi.edu/Pubs/E-project/Available/E-project-042512-

174736/unrestricted/MQP_FinalReport.pdf.

Pritsker, D. (2015). Hybrid implementation of Extended Kalman Filter on an FPGA.

2015 IEEE Radar Conference (RadarCon).

Strother, D. (2011). *Open-source FPGA Stereo Vision Core released*. [online]

danstrother.com. Available at: https://danstrother.com/2011/06/10/fpga-stereo-visioncore-

released/.

Xue, B., De-ming, W., Jun-jun, T. and Bu-hui, Z. (2011). The Hardware Design

and
Simulation of Kalman Filter Based on IP Core and Time-sharing Multiplex. *2011 Fourth International Conference on Intelligent Computation Technology and Automation*.

Zicari, P., Perri, S., Corsonello, P. and Cocorullo, G. (2012). Low-cost FPGA stereo
vision system for real time disparity maps calculation. *Microprocessors and Microsystems*, 36(4), pp.281-288.