Literature Review

Submitted in partial fulfilment
of the requirements of the degree of

Bachelor of Science (Honours)

of Rhodes University

S. A. Currie

Grahamstown, South Africa
May 2012

0.1 Introduction

The aim of this research project is to create a system that simulates an autopilot for the remote controlled Syma S107 mini helicopter. This helicopter must be tracked using a Kinect camera and sent messages in real time, using an Arduino board, in order to make it autonomous. It will therefore be necessary to find a suitable method of object tracking. The main constraint of the chosen object tracking algorithms is that they have to be very fast, in order to track a helicopter accurately. They also need to accurately determine the position and orientation of the helicopter at any given time. Other challenges, such as noise and occlusion, should not be a problem as the scene will be static and indoors with very few illumination changes and little noise. There will also be only one moving object in the scene. Since this project is a first step into the world of autonomy, it is necessary to give a brief background of the different classes of autonomous vehicles and examples that already exist in these classes. Image processing and object tracking form the main focus of this project; therefore it is necessary to discuss the relevant literature and algorithms already existing in this area.

0.2 Autonomous Vehicles

Helicopters are fairly difficult to control due to their complexity [7]. They are, however, very useful vehicles and have a lot of real-world applications. Autonomous helicopters have many advantages and uses. They could be used for security, such as helping pilots in situations where they lose control of the vehicle, or provide surveillance of aerial space. They can also be used for monitoring things like volcanoes, air pollution and other environmental factors. Another very pertinent use is to help in dangerous situations, such as removing mines and going into radioactive atmospheres [7, pp. 1–2]. Autonomous vehicles generally fall into 3 categories, namely, on-board vision, external vision and both on-board and external vision.

0.2.1 On-board vision

On-board vision is when all the tracking and navigation functionality is on-board the vehicle. The Stanford entry in the DARPA Grand Challenge is an example of on-board vision. The vehicle was named Stanley and it won the challenge. The rules of the challenge

included that no manual intervention was allowed and the vehicles had to drive themselves [21]. According to [21], the car had a GPS system, camera, many antennae and sensors attached to its roof rack. All the navigation was done on-board. Similarly, in a project done by [3], an autonomous model helicopter was created with on-board processors, video cameras, gyroscopes and accelerometers. All image processing and navigation is done on-board the helicopter itself.

0.2.2 Remote or external vision

External vision is when the vehicle is tracked and sent navigation data from an external source. According to [19], the vehicle used may have weight limits or size constraints, making on-board sensors unsuitable. In this case, external devices need to be used. In the study carried out in [19], an optical 3D capturing system, consisting of many cameras, was used to keep track of a mini remote-controlled helicopter. Therefore all the measurements and navigation decisions were done on an external computer, and information was sent to the helicopter from the computer via a serial connection.

0.2.3 External and on-board vision

External and on-board vision is a mixture of the above categories and has some on-board and some external functionality. The study carried out in [2] falls into this category, as two cameras are used to estimate the position of a remote controlled quad rotor helicopter; one of the cameras is located on the ground and the other is located on-board the helicopter. Each of these cameras are connected to different computers which are connected via a network. The on-board camera sends information to the external computer through the network and the external camera sends information to the helicopter using a remote control device with a parallel port.

0.3 Image Processing

Image processing is the method of taking in an image as input and doing a number of operations on the image in order to enhance it, or extract necessary information out of it. [1] Digital image processing is when these actions are performed by a computer, and

[1] http://www.engineersgarage.com/articles/image-processing-tutorial-applications

not a human. The digital image which is used by the computer is made up of many elements called pixels. Analog image processing is when these actions are performed by humans, but this is limited because human vision only includes the visual band of the electromagnetic spectrum. Computers however, can process images which range over almost the entire electromagnetic spectrum (such as ultrasound, radio waves and electron microscopy) and therefore is a lot less limited [12]. According to [15, p. 5], most of the low-level methods used for image processing were developed in the 1970s or earlier.

0.3.1 Image preprocessing

Preprocessing is performed on images, to try and correct or enhance certain features in the image. According to [15, pp. 108–111], there are four main types of preprocessing:

- Brightness transformations - adjust the brightness of the image
- Geometric transformations - remove distortions in the image
- Local neighbourhood preprocessing - consists of smoothing to reduce unwanted noise and edge detection to find region boundaries
- Image restoration - reduce image degradation. In the study by [25], an image restoration method was used to reduce the effects of fog, smoke or haze which cause an image to have low contrast and faded colours.

0.4 Object Tracking

Object tracking is defined by [28], as the estimation of a moving object's trajectory in an image plane. According to [28], object tracking methods can be divided into four main areas. Each of these areas can be implemented in a number of different ways and the methods chosen for each will differ according to the environment and context in which the tracking is performed.

0.4.1 Object Shape Representation

There are many different ways of representing an object in an image. One representation is to use points, which are suitable for smaller objects. In the study by [11], objects

are represented by interest points based on colour. Another representation makes use of primitive geometric shapes which are suitable for rigid objects. In the study by [8], the object is represented by a circular region. A rectangle is used to represent the object in [6], which also falls under primitive shapes. An object can also be represented by its silhouette and contour. The object in [17] is represented by a contour which is made up of the edges of the object. This is suitable for non-rigid objects. Skeletal models can also be used to represent and object. These are suitable for articulated and rigid objects. An object can also just be represented by a blob or many blobs with similar colour or flow [16].

0.4.2 Feature Selection

Certain features need to be extracted from an object in order to distinguish it from the rest of the scene. These features should be chosen according to the type and appearance of the object. Different aspects of the scene will also affect which features should be chosen as large changes in illumination and noise can negatively affect features like colour.

Colour

Colour is the most popular visual feature that may be used. This would be most suitable for objects that are of a uniform, bright colour. In a study done by [10], the development of a tracking method is explained, which only uses colour information from an image. The colour tracking method works by identifying regions of similar average colour in each image frame. The object to be tracked is split into several regions, each of which is described by a colour vector. These regions can then be tracked by evaluating the goodness of fit between a measurement vector and its target. Colour is also used as a feature in [27]. Since RGB colour is very sensitive to illumination changes, [27] used a conversion method in OpenCV to convert RGB colour to YUV which is another colour space and is useful because the intensity component can be dealt with separately. The region of the object to be tracked is selected by the user and the colour cluster is determined for this region. This worked well on image sequences where the object's colour was quite distinct from the background. If the object being tracked is very similar in colour to other objects in the background, then it is not an optimal feature to use. Colours should also not be used in scenes with drastic illumination changes as the colour will vary too much. However, there are ways of overcoming this. The study in [18] proposes a method of object tracking

that combines colour distributions with particle filtering. Adaptive model updates are used which update the target object in slow changing images.

Edges

Edge information is another popular feature that can be used, which is less sensitive to illumination changes than colour. According to [22], edges are also very robust to noise. Edges are found where there are large intensity changes in the image. There are many edge detection algorithms and these are generally used with tracking algorithms that track the boundary of objects. According to [**?**], four examples of edge detection algorithms are the Canny edge detector, the Sobel edge detector, the Sarkar-Boyer edge detector and the Nalwa-Binford edge detector. The Canny edge detector is very well-known and according to [22, p. 271], is the best "all-round" method for detecting edges. It was created by John Canny and proposed in the paper "A Computational Approach To Edge Detection" in 1986. The main three criteria it satisfies are that it has a low error rate so as to not miss important edges, the edge points found should be localized meaning there should not be a large distance between them and the real edge and lastly, there should be a minimal response meaning an edge should only be detected once [22, pp. 271–272] [2]. In the study by [27], the OpenCV implementation of the Canny filter is used to detect the edges of an object.

Texture

Texture is another feature that may be used and it measures properties like smoothness and regularity of an object. This, like edges, is not very sensitive to illumination changes. If the texture of the object is very distinct from the background, then it is a good choice for feature extraction. According to [22, p. 246], operators that may be performed on an image to extract different textures are standard deviation or variance, entropy and local range. Some common uses of texture recognition include X-rays, cloud-type recognition and crop yield [15, p. 667].

0.4.3 Object Detection

According to [16], this can either be based on temporal data or spatial data. Temporal data usually relies on a static background and movement is therefore equal to the dif-

[2] http://www2.it.lut.fi/kurssit/07-08/CT20A6100/seminars/2009-2010/Canny.pdf

ference between images. Background subtraction is very popular in this case. Temporal data assumes that there is only one moving object in the scene. This means that any differences between images is probably due to the moving object. Spatial data usually uses thresholding or statistical approaches. Temporal data is generally easier to extract than spatial data [16].

Background Subtraction

Background subtraction is a way of detecting moving objects by finding the difference between the current image and a background image [3]. Background subtraction relies on a fixed camera [5].Background subtraction can be performed using only two images, but an improved method is to use three images to make it more robust [16]. Another method to make background subtraction more robust is to update the background model iteratively so that any changes in the scene are taken into account. In the study by [5], popular methods of background subtraction are evaluated. The simplest form of background subtraction is frame difference which literally subtracts an image frame from the previous image frame and if the difference for a certain pixel is higher than some threshold, that pixel is part of the foreground. This method will only work well if the object is moving constantly and if the object is quite distinct from the background [1, pp. 44–45]. Using frame differencing on an object with uniform intensity may result in holes in the foreground mask. These holes will have to be filled with other methods. According to [1, pp. 44-45], frame differencing is the fastest of all background subtraction algorithms. Frame difference is used by [6] on a low resolution image to detect moving objects. By using low resolution images, unwanted noise, like moving leaves or branches, can be eliminated. Morphological operators are used to fill holes in the object. The operators used by [6] are dilation and erosion. Another method is to use one Gaussian distribution for every pixel, but this is only sufficient for a scene with fairly static backgrounds [24] . For non-static backgrounds, it is better to use a Gaussian Mixture Model, where every pixel is modeled by a mixture of Gaussians. In [24], multiple Gaussians are used which are updated in order to keep track of changes in the scene such as lighting, or movement in the background.

Optical Flow

Flow can also be used for temporal data [16] and this uses points or features in images to detect motion. It analyses the image changes as a result of motion in image sequences

[3]http://wwwstaff.it.uts.edu.au/ massimo/BackgroundSubtractionReview-Piccardi.pdf

[15, pp. 685–686]. Optical flow is the velocity of the moving object. The Kalman filter is a popular way of measuring optical flow of a single object. It assumes noise to be white and Gaussian. Since this method also falls under object tracking, it is discussed in more detail later.

Segmentation

Segmentation is used to divide an image into similar parts or regions [28]. Segmentation uses spatial data and can be classified as a statistical approach. This approach is very robust compared to background subtraction and can handle more diverse scenes [16]. According to [15, pp. 123–210], segmentation can be divided up into three main categories. These are thresholding, edge-based segmentation and region-based segmentation. Thresholding is the fastest and most simple method. A threshold constant is chosen and if the object is brighter than the background, then pixels lower than this value are considered to be part of the background. Thresholding was used by [27] and [23]. In both cases, this method was very sensitive to noise and additional methods, such as low-pass filtering and noise filtering, had to be used to overcome this. Thresholding should only really be used if the object has a distinct intensity or colour compared to the background [16]. The next method is edge-based segmentation which uses edge information, obtained through methods mentioned earlier, to segment an image into objects. Like thresholding, edge-based segmentation is also sensitive to noise [15, p. 207]. This segmentation method is used by [13] and it was very successful and computationally efficient.The last category of segmentation is region-based segmentation. This works by grouping similar neighbor pixels, which results in the formation of regions. This segmentation method is not as sensitive to noise as the other methods. It is suggested by [15, p. 176] that results from edge and region based segmentation be combined. This is demonstrated in the papers by [20] and [14], where a hybrid of the two segmentation techniques were used for more accurate image segmentation.

0.4.4 Tracking

Tracking combines all these previously mentioned areas to find corresponding objects across image frames. Tracking is used to find the trajectory of an object and its current region at different times in an image sequence [28]. According to [28], there are three main tracking categories. Point tracking is the first category, where objects are represented by

points. The next category is kernel tracking which uses a kernel template representing the objects shape and appearance. The motion of this kernel can be computed to track the object. Examples of kernel tracking are mean-shift and block matching. Silhouette tracking is the third category and this estimates an object region containing information such as edge maps and density. Shape matching or contour evolution can then be used to track these regions or silhouettes. According to [18], tracking methods can be classified into a bottom-up approach, where an object is extracted from the image and then tracked, or a top-down approach, where an object hypothesis is made and then verified in the image.

Point Tracking

In point tracking, objects are represented as points and the motion and position of these points are tracked in consecutive image frames. According to [28], point tracking methods can either be deterministic or statistical. Deterministic methods define a cost function which is made up of constraints like maximum velocity, common motion and rigidity [28]. This cost function must then be minimized for tracking. A greedy algorithm can be used for this which iteratively optimizes point correspondences [26]. This algorithm is used by [26], which is based on the algorithm used in a paper by Sethi and Jain. The algorithm is modified in [26] to preserve a lot of motion information so that point measurements are not missed.

Statistical methods form the other category of point tracking and these model uncertainties to handle noise in an image. A well-known method for statistical point tracking is multiple hypothesis tracking. A set of hypotheses are defined for an object and predictions are made for each hypothesis for the object's position. The hypothesis with the highest prediction is the most likely and is chosen for tracking [28]. Multiple hypothesis tracking is used in [10], in order to overcome occlusion. A set of 10 hypotheses are defined which can be classed in a certain occluding state. Probabilities are then assigned to these hypotheses and the likelihoods are established. According to the results in [10], this method was very successful in tracking objects in different states of occlusion.

Other statistical methods that can be used to track single objects are the Kalman filter and Particle filters. The Kalman filter is limited to a linear system and uses prediction and correction to estimate an object's motion [28]. Since the Kalman filter relies on a

Gaussian distribution, a particle filter is necessary for state variables that do not follow this distribution. In the study by [18], particle filtering was used with colour distributions. According to [18], it models uncertainty which means that it works well in situations with a lot of clutter or occlusion. Initialization of the particle filter was done using an algorithm based on Support Vector Machines. The results from the study in [18], showed that this method of using colour distributions along with particle filtering is very effective in tracking fast-moving, non-rigid objects.

Kernel Tracking

Kernel tracking represents an object as a geometric shape, called a kernel, and estimates the motion of this kernel in consecutive frames. Template tracking is very commonly used to track a single object. It uses brute force to search an image for a region that matches the template in the previous image [28]. The brute force searching results in this method being computationally expensive, but this can be overcome by optimizations to the method, such as limiting the search to a certain region. Mean-shift is used for template matching, in [8], which eliminates the need for brute force. Mean shift was first introduced in 1975 by Fukunaga and Hostetler in the paper called "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". It is an iterative algorithm that shifts a point towards the average of other points in that area [4]. Kernel tracking methods can generally be used for real-time tracking as the rigidity constraint of the kernel allows for fast computation [28]. A limitation of kernel tracking is that parts of the background may appear inside the kernel, but this can be overcome by making the kernel inside the object, instead of around it [28].

Silhouette Tracking

Silhouette tracking is generally used for objects that cannot be represented by a simple shape and makes use of an objects appearance. The human body is an example of this as objects like hands or heads are complex shapes. According to [28], silhouette tracking can fall under two categories which are shape matching and contour tracking. Shape matching is quite similar to template matching, discussed in kernel tracking, in the way that it searches an image to find a region that matches the silhouette model from the previous image. Shape matching is used by [4], where the underlying shape of an object

[4]http://www.loria.fr/ berger/Enseignement/Master2/Exposes/meanShiftCluster.pdf

is described by its shape context and this is used to find corresponding, matching shape contexts in consecutive images. Contour tracking is the other form of silhouette tracking and this differs from shape matching in the way that it uses the contour of an object that evolves in consecutive images. In the study by [**?**], human motion is tracked by finding the contour of the body and using this to estimate the changing pose of the body in consecutive images.

0.4.5 Orientation or Pose Estimation

There are many instances when the orientation or pose of the object being tracked needs to be estimated. For example, to recognize different gestures or movements done by a human body, configuration of the body needs to be identified. According to [16], methods used to recognize these different poses or orientations can be classified into model-free or direct model use. Model free is when there is no a priori information used to estimate the pose. Instead, the pose or orientation can be represented by a set of points or shapes with meaning attached to them. Markers, attached to the object, can also be used in this case. For example for finding the orientation of an object, markers can be placed on it and a vector can be created that measures the distances between the markers [5]. Direct model use involves using a priori knowledge to match a certain pose or orientation to a predefined template [16]. This method requires a lot of searching and therefore very efficient search algorithms would need to be used to achieve fast results [6].

0.4.6 Performance Evaluation of Different Tracking Algorithms

The study done by [9], describes a set of metrics which can be used to evaluate the performance of an object tracking algorithm. These metrics are first described and then used to test an experimental industrial tracker from BARCO, which is a global technology company, and the blob tracker from OpenCV 1.0. Motion tracking is then defined by [9], as the process of estimating a non-background object's position for each image frame and producing a set of tracks for this object. These tracks are often represented by a bounding box.

[5] http://lasa.epfl.ch/publications/uploadedFiles/hersch08iterative.pdf
[6] http://www.robots.ox.ac.uk/ teo/thesis/Thesis/deCampos3DHandTracking2.pdf

The metrics proposed by [9] include high-level ones like True Positive, False Positive and False Negative tracks. True positive is when a track is correctly detected and false positive is when there is a false alarm and the track is considered to have been detected when it was not. False negative refers to a track detection failure where nothing has been detected. These metrics can then be used to produce other metrics like specificity and accuracy. Other metrics used are track fragmentation, which shows the lack of continuity of a track, and ID change. These two metrics help to evaluate tracks integrity. The remaining metrics used help to evaluate the accuracy of the motion tracking itself. Track matching error measures the error between the positions of a system track and the ground truth (which means the position of the actual object in real life). This error should ideally be very small. Similarly, closeness of track is a metric that measures the spatial overlap of the system track and the ground truth. Lastly, latency of the system track measures the time between the appearance of an object and when it starts to be tracked.

The two trackers in [9] were then evaluated on six different video sequences. Each video posed a challenge for the tracker such as illumination changes and fast moving objects. The results show that the BARCO tracker generally had a higher performance than the OpenCV tracker however, the OpenCV tracker was better at estimating an objects position as it had a lower track matching error. The authors argue that without their rich set of metrics, it would be difficult to find the causes of good or bad performance for a given tracker. The authors conclude that these metrics should be used to evaluate more trackers in the future.

0.4.7 Conclusion

The literature discussed shows that image processing and object tracking are both very broad subjects and are made up of a very large number of methods and algorithms. The literature discussed in this review provides a brief overview of the different implementations that can be used for object tracking and the environment for which they are most suitable. The different categories of autonomous vehicles were also discussed, giving examples from literature to indicate the differences, advantages and disadvantages between them.

Bibliography

[1] Aguilar-Ponce, R. M. *Automated object detection and tracking based on clustered sensor networks*. PhD thesis, University of Louisiana, 2007. AAI3294839.

[2] Altug, E., Ostrowski, J. P., and Taylor, C. J. Quadrotor control using dual camera cisual feedback. In *International Conference on Robotics & Automation* (2003), IEEE, pp. 4294–4299.

[3] Amidi, O., Mesaki, Y., and Kanade, T. Research on an autonomous vision-guided helicopter, 1993.

[4] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4 (Apr. 2002), 509–522.

[5] Benezeth, Y, J. P. E. B. L. H. R. C. Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition* (2008), pp. 1–4.

[6] Budi Sugandi, Hyoungseop Kim., J. K. T., and Ishikawa, S. *Object Tracking*. InTech, 2011, ch. 1, pp. 3–22.

[7] Castillo, P., Lozano, R., and Dzul, A. *Modelling and control of mini-flying machines*. Advances in industrial control. Springer, 2005.

[8] Dorin Comaniciu, V. R., and Meer, P. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25* (2003), 564–577.

[9] Fei Yin, Dimitrios Makris, S. V. Performance evaluation of object tracking algorithms, 2007.

[10] Fieguth, P., and Terzopoulos, D. Color-based tracking of heads and other mobile objects at video frame rates. In *in Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (1997), pp. 21–27.

[11] Gabriel, P., b. Hayet, J., Piater, J., and Verly, J. Object tracking using color interest points. In *in IEEE Proc. of Int. Conf. on Advanced Video and Signal based Surveillance, AVSS 05* (2005), pp. 159–164.

[12] Gonzalez, R., and Woods, R. *Digital Image Processing*. Pearson/Prentice Hall, 2002.

[13] Jiang, X., and Bunke, H. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding 73* (1999), 183–199.

[14] Jianping Fan, David Yau, A. E., and Aref, W. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transaction on Image Processing 10* (2001), 1454–1466.

[15] Milan Sonka, V. H., and Boyle, R. *Image Processing, Analysis and Machine Vision*, 2 ed. Brooks/Cole, 1999.

[16] Moeslund, T. B., and Granum, E. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst. 81*, 3 (Mar. 2001), 231–268.

[17] Myung-Cheol Roh, Tae-Yong Kim, J. P., and Lee, S.-W. Accurate object contour tracking based on boundary edge selection. *Pattern Recognition 40* (March 2007), 931–943.

[18] Nummiaro, K., Koller-meier, E., and Gool, L. V. Color features for tracking non-rigid objects. *Special Issue on Visual Surveillance, Chinese Journal of Automation, May 2003 29* (2003), 345–355.

[19] P. Sneep, J. R. Tracking system and communication interface for miniature rc helicopters. Hello, 2011.

[20] Pavlidis, T., and Liow, Y.-T. Integrating region growing and edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 3 (Mar. 1990), 225–233.

[21] Sebastian Thrun, Mike Montemerlo, H. D. D. S. A. A. J. D. P. F. J. G. M. H. G. H. K. L. C. O. M. P. V. P., and Stang, P. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics 23* (2006), 661–692.

[22] Solomon, C., and Breckon, T. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab.* Wiley-Blackwell, 2010. ISBN-13: 978-0470844731.

[23] Sommerfeld, J. Image processing and object tracking from single camera. Master's thesis, Royal Institute of Technology (KTH), 2006.

[24] Stauffer, C., and Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1999), vol. 2, IEEE Computer Society, pp. 246–252.

[25] Tarel, J.-P., and Hautiere, N. Fast visibility restoration from a single color or gray level image. In *Computer Vision, IEEE 12th International Conference* (2009), pp. 2201–2208.

[26] Veenman, C. J., Hendriks, E. A., Veenman, C. J., Hendriks, E. A., and Reinders, M. J. A fast and robust point tracking algorithm. In *In IEEE International Conference on Image Processing, volume III* (1998), pp. 653–657.

[27] Xu, R. Y. D., Allen, J. G., and Jin, J. S. Robust real-time tracking of non-rigid objects. In *Proceedings of the Pan-Sydney area workshop on Visual information processing* (Darlinghurst, Australia, Australia, 2004), VIP '05, Australian Computer Society, Inc., pp. 95–98.

[28] Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Computing Surveys (CSUR) 38* (2006), 45.