Extensions to Object Recognition in the Four-Legged League

Christopher J. Seysener, Craig L. Murch, and Richard H. Middleton

School of Electrical Engineering & Computer Science,
The University of Newcastle, Callaghan 2308, Australia
{cseysener, cmurch, rick}@eecs.newcastle.edu.au
http://robots.newcastle.edu.au

Abstract. Humans process images with apparent ease, quickly filtering out useless information and identifying objects based on their shape and colour. However, the undertaking of visual processing and the implementation of object recognition systems on a robot can be a challenging task. While many algorithms exist for machine vision, fewer have been developed with the efficiency required to allow real-time operation on a processor limited platform. This paper focuses on several efficient algorithms designed to identify field landmarks and objects found in the controlled environment of the RoboCup Four-Legged League.

1 Introduction

Robot vision systems are often required to identify landmarks relevant to the robot's operation. In autonomous robot soccer, situations arise where the robot operates for extended periods without viewing critical landmarks on the field such as goal posts and other objects. This lack of information leads to the robot becoming disorientated and uncertain about its position.

In the past, methods used to alleviate this problem have mostly involved so-called "active localisation". Active localisation requires the robot to periodically interrupt its normal operation to better determine its location. For example, a robot may interrupt its gaze to search for a landmark before executing a critical manoeuvre (such as kicking the ball). However, because of the time critical nature of robot soccer, the use of active localisation can allow opposition robots to pounce on the ball while the localising robot wastes time determining its position and orientation. Identification of additional landmarks is therefore beneficial as it reduces the need for active localisation.

Many vision algorithms developed in recent years are inappropriate for limited processor systems and real-time applications. In this paper, we present some efficient algorithms to solve complex problems such as edge detection and data association so commonly seen in both theoretical and practical problems. While the solutions presented are in the context of the RoboCup Four-Legged League [1], it is expected that some of the ideas and algorithms could be useful in areas such as automation so commonly seen in industry.

D. Nardi et al. (Eds.): RoboCup 2004, LNAI 3276, pp. 274–285, 2005.

The objective of the methods described in this paper is to increase the number of landmarks on the field that the vision system can recognise. By locating field and sidelines in the image, it is possible to locate points such as the corners of the penalty box, the centre circle and the intersection of the centre line with the sideline (the "centre corner").

The accuracy of measurements to objects detected by the vision system is also of great importance. In the context of a soccer match and in many other areas, obstructions can occur which impede both the identification of objects and measurements to them. The practice of fitting circles to the ball provides a robust mechanism through which accurate heading and distance information can be determined, even in the presence of obstructions and despite the limited processing power available.

2 Field and Sideline Detection

As discussed in the introduction the attainment of additional information from which to localise is of paramount importance. One idea to increase the quantity of objects identified by the vision system is to make use of the field and sideline information. The following sections document the process and algorithm used in identifying and determining the equation of the field and sidelines within the image.

2.1 Line Detection

While existing techniques for edge detection, such as those of the German team, have focused on examining the camera image (in YUV colour space) for a shift in the Y component [2], this report focuses on the identification of field and sidelines using a classified image. A classified image is one which has, through the use of a look up table, been mapped to a set of colours identifiable by the robot. Using the classified image, the key to determining the presence of field and sidelines lies in the systems ability to discern field-green/boundary-white and boundary-white/field-green transitions. These transitions identify the boundary between the field and both field and side lines. For example, for a field line to be present, a field-green/boundary-white transition and a corresponding boundary-white/field-green transition must be found. These two transitions mark the change from field to field-line and back to field again, ruling out the possibility that the point was actually part of the sideline.

The algorithm searches a sparse grid structure over the entire image to discover the field and sideline transitions. While a complete scan of all pixels would have provided more accurate data, line transitions are typically several pixels wide and so are still identifiable when checking only the pixels located on grid-lines. Under the current implementation, a grid with a spacing of 4 by 2 pixels is used in the identification of field and sideline points. While this grid is considered fine, testing has revealed that should additional processor time be required, the size of this grid may be increased without a serious negative effect on the identification of points in the image.

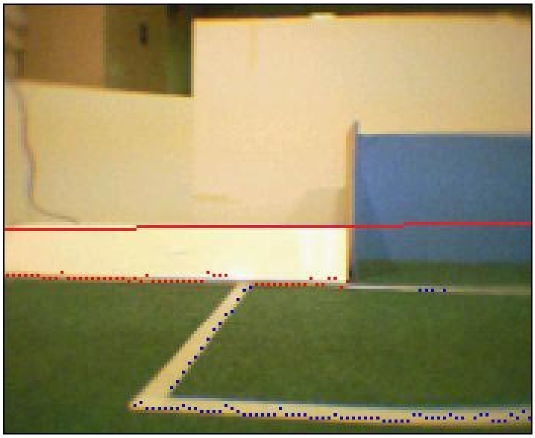

Fig. 1. Identification of field (blue) and side (red) line points

The points identified as field and sideline can be seen in Figure 1. The red line present in this image represents the horizon line, which is used to identify those parts of the image which should be scanned and those which can be ignored. This technique obviously reduces the area to be scanned and hence the processing load in many images. Further detail on the generation of the horizon line can be found in [2, 3].

2.2 Field and Sideline Identification

Having acquired the transitions in the image believed to be part of a field or sideline, the problem of associating points with individual lines offers a further challenge to be investigated. The sideline case is trivial, in that sidelines are always separated horizontally in the image. However, multiple field lines may appear stacked, a fact illustrated by Figure 1.

The data association problem for field lines was solved using a simple clustering algorithm such that if the distance of a point from any previously generated line is too great then a new line will be formed. Points are always processed from left to right. Once all points have been allocated to a line, the start and end points of each line are compared with other lines identified. This step determines whether the two lines actually represent a continuous field line broken in two due to a corner. Any pair of lines found to be continuous are merged. It is for this reason that the points are always processed from left to right: By placing an ordering on the initial line formation step, longer lines tend to generated and therefore less merging must be performed later.

For example, in Figure 1 the clustering algorithm results in three field lines. The first field line is obvious and represents the four points identified in the goalmouth, while the second and third lines are not so obvious and represent the two components of the penalty box. The presence of the corner forces the points to be split in two as in processing the points from left to right, the distance of the points from the end of the generated line eventually becomes too great forcing a new line to be formed. In this example, the two lines that form the corner due to the proximity of their ends will be determined to be part of the same field line and will be merged.

Upon successful discovery of a fixed number of points believed to be part of the same line, the principle of Least Squares Line Fitting can be used to determine the equation of the line in the image. The algorithm may also be used to filter outliers based on their distance from the generated line thus increasing the tolerance of the system to noisy input data.

3 Landmark Identification Using Field and Sidelines

This following sections document the way in which field and sideline data identified is used to locate landmarks on the field useful for robot localisation. The landmarks identified by the algorithms presented in the following sections are shown in Figure 2.

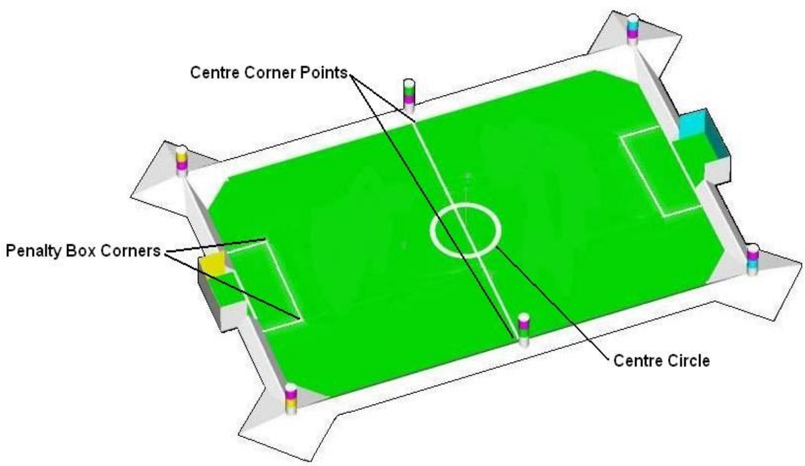

Fig. 2. Landmarks identified in the following sections

3.1 Penalty Box Corner Detection

While a number of approaches were trialled for the identification of penalty box corners, many of these proved too time consuming or error prone for use in a soccer game. The solution described below was successful in identifying corners both accurately and quickly, although it has the caveat of being unable to recognise the rare situation where multiple corners are present in a single line. This technique was inspired in part by the well known split and merge algorithm [4]. The first step is to identify the point furthest from the line fitted

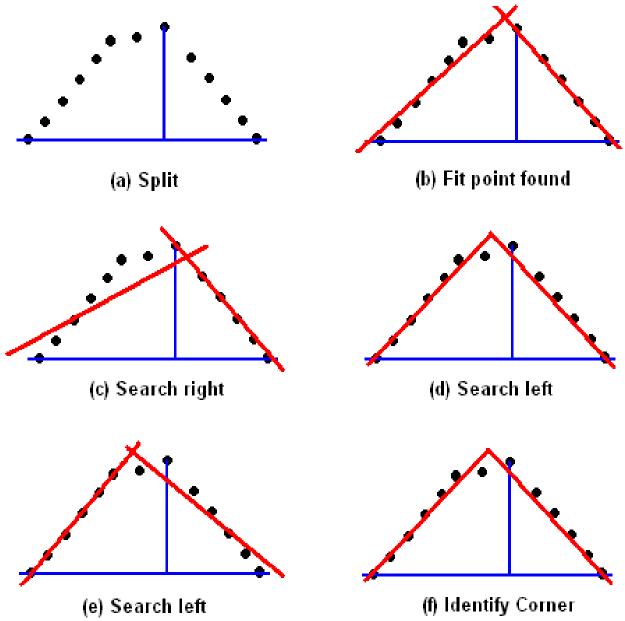

Fig. 3. The split and fit algorithm

through the start and end points of the data set. The Split and Merge algorithm then splits the data set into smaller and smaller line segments in an attempt to find an accurate equation for the line. In contrast, we use a simple Hill Climbing algorithm to search in the direction that minimises the residual of the fitting of two lines to the points either side of a candidate split point. The algorithm stops when the point identified is such that residual of the line fitting operations is minimised and no further improvements can be made. The corner point itself is taken as the intersection of the two lines fitted to the data set. This last step of taking the intersection point of the two lines minimises the risk of taking a point identified incorrectly in the original transition discovery phase due to misclassification or obstruction.

As can be seen in the example, the algorithm operates using a simple hill climbing process. The steps used can be described as follows:

a) Fit a line through the two extreme points of the dataset and find the point most distant to this line. This point defines the starting point of the search
b) Split the data set in two and calculate (using least squares) the residual of the two lines fitted either side of the centre point found
c) Take the first point to the right of the centre point and refit the two lines. Note that in this example the error of the system increases, indicating a move away from the corner

d) Take the first point to the left of the centre point and refit the two lines. Note that this decreases the error of the system and hence a move towards the corner
e) The algorithm continues to search in the direction that minimises the error of the system. In this case, the error has increased by searching left indicating that the corner point has already been passed.
f) Two lines are fitted through the point that minimises the error of the system and the intersection of these lines taken as the corner point. Despite the bad point in the example, the algorithm correctly identifies the true corner.

While the point identified in the first step is often the corner point, the additional steps have proved to give reliable and robust determination of the corner point even in situations where significant amounts of noise are present in the system. This algorithm has proved an extremely efficient solution to the problem of finding a corner in a system of ordered points, easily executed in real-time on the robot.

3.2 Centre Circle Detection

As the centre circle closely resembles a field line, it made sense that the detection of the centre circle should somehow be related to the detection of field lines within the image. This was achieved by observing that the centre circle provided a special case when present, in that from most angles three sets of field-green/border-white/field-green transitions occurred in the same column. The presence of the centre circle is therefore determined by the existence of three or more sets of field line transitions in the same column (referred to as a triple). An example of this situation can be seen in Figure 4.

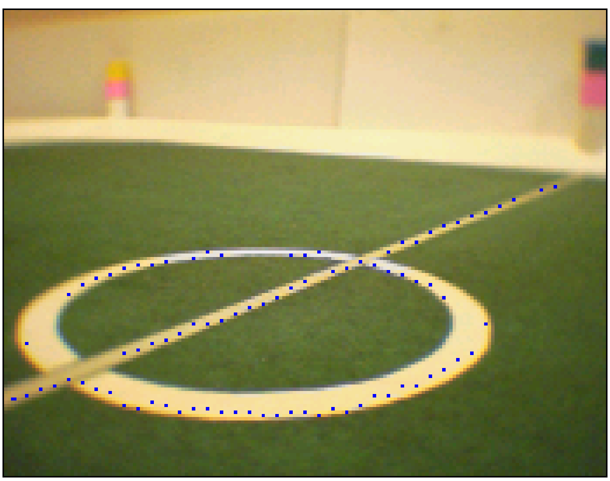

Fig. 4. Centre circle

The parameters of the centre circle are determined from the sets of triples identified in the image. Using the sets of triples, the centre of the circle was defined by the point that bisected the line fitted through the two triples most separated in the image.

Although the discovery of the centre circle is only possible from certain angles upon the field, it does provide a foundation from which further work can be performed. Future work may include research into the fitting of circles or ellipses to determine accurately the presence and parameters of the circle.

3.3 Centre Corner Point Detection

Another area in which field and side line information is used, is in the identification of the two points where the centre field line intersects with the sideline. An example of the point considered the centre corner point can be seen in Figure 5.

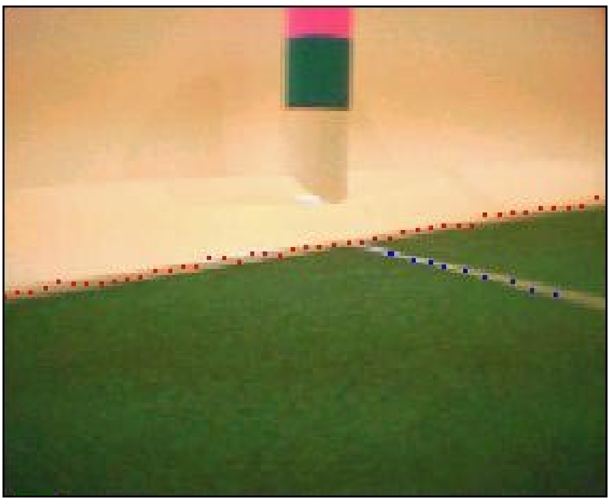

Fig. 5. Centre corner point

The detection of these landmarks requires the detection of a field line intersecting with a sideline. This condition is satisfied at the goal mouth as well as at the centre corner point, so an additional constraint to reduce the number of false positives is that the goal must not be present in the image. On checking that the end of a field line is sufficiently close to the sideline, the intersection of the field and sideline is taken as the centre corner point.

4 Circle Fitting to the Ball

A commonly used method for object recognition in the Four-Legged League is to find a rectangular blob around all colour objects within the image. The size of the blob determines the distance to the object. While this is convenient to code and can be extremely efficient, it also results in an overall reduction of information and introduces error that makes the perceived location of objects less precise. This problem is most evident in the recognition of the ball due to its circular shape (other notable objects on the soccer field are roughly rectangular).

In ball recognition, it was decided that points located on the perimeter of the ball should be identified and a circle fitted through these points. It was hoped that this method would make the system more tolerant of obstructions in front of the ball, such as another robot or the edge of the image. For example, consider the situation shown in Figure 6(a) where the lower part of the ball is obstructed by the robots leg.

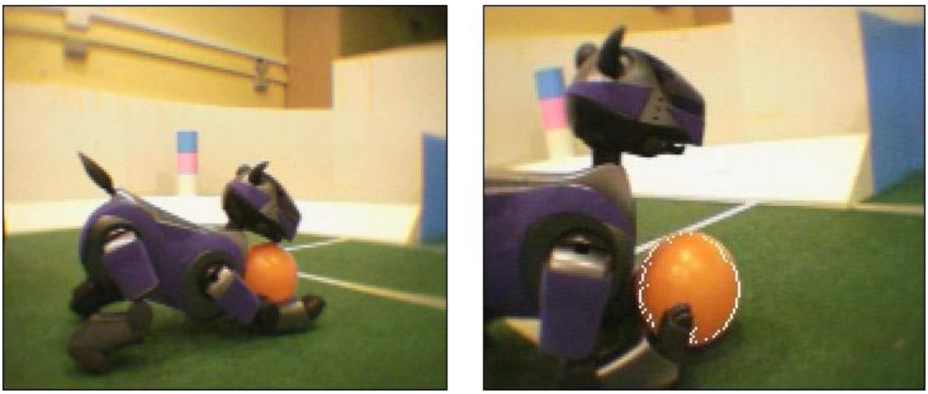

Fig. 6. (a) Obstruction caused by a robot. (b) Point determination after blob formation

The result is a blob smaller than the actual size of the ball and a resultant distance much greater than the actual distance to the ball. Fitting a circle using points on the perimeter helps overcome this problem, resulting in a better approximation for the radius of the ball.

4.1 Determining Fitted Points

In fitting a circle to the ball, it is necessary to know those points (coordinates in the image) which define the perimeter. The technique identified for locating perimeter points, involved the recovery of information lost during the identification of the ball. This algorithm performs a horizontal search from the edge of the circle blob towards the centre, recording the first pixel found to match the colour of the object (i.e. orange). In using this algorithm, the pixels searched are those shadowed by the blob identified as the ball, which are not orange. Specific knowledge of the point's location is used to exclude certain points from the set, such as points located on the edge of the image, which are not genuinely on the perimeter of the ball. Future work in this area may include the use of a Convex Hull to filter points incorrectly identified within the ball. Figure 6(b) shows the points determined by the algorithm presented above.

4.2 Least Squares Fitting to the Ball

Although techniques such as fitting a circle through three points were tested, the method of shape fitting using Least Squares Fitting (LSF) proved the most successful. Whilst this algorithm is slightly more computationally expensive, the algorithm generates the most accurate fit, especially in cases where obstruction of the ball occurs. It was also noted that as the algorithm uses a higher number of points the generated result is more tolerant of noisy data.

Whilst many different papers have been written about the Least Squares Fitting of circles through a set of points, the paper used as the primary reference for this research was that by Chernov and Lesort [6], which discusses the various strengths, weaknesses and efficiencies of several commonly used circle fitting algorithms. The concept of Least Squares Fitting of circles is based on minimising the mean square distances from the fitted circle to the data points used. Given n points $(x_i, y_i), 1 \leq i \leq n$, the objective function is defined by

$$F = \sum_{i=1}^{n} d_i^2 \tag{1}$$

where d_i is the Euclidian (geometric) distance from the circle to the data points. If the circle satisfies the equation

$$(x - a)^2 + (y - b)^2 = R^2 \tag{2}$$

where (a, b) is the centre and R is the radius, then

$$d_i = \sqrt{(x_i - a)^2 + (y_i - b)^2} - R \tag{3}$$

The minimisation of the objection function $min\{F\}$ for $a, b, R > 0)$ is a non-linear problem with no closed form solution. For this reason, there is no algorithm for computing the minimum of F with all known algorithms being either iterative and computationally expensive or approximate. It was therefore important that the correct algorithm be chosen for implementation on the robot to minimise the load placed on both processor and memory.

Having reviewed the results in [6] it was decided that a combination of both geometric and algebraic methods be used. The Levenberg-Marquardt [5, 7, 8] geometric fit method was selected above others, such as the Landau and Spath algorithms, due to its efficiency. This algorithm is in essence a classical form of the Gauss-Newton method but with a modification known as the Levenberg-Marquardt correction. This algorithm has been shown to be stable, reliable and under certain conditions rapidly convergent. The fitting of circles with the Levenberg-Marquardt method is described in many papers.

It is commonly recognised that iterative algorithms for minimising non-linear functions, such as the geometric circle-fitting algorithm above, are sensitive to the choice of the initial guess. As a rule, an initial guess must be provided which is close enough to the minimum of F to ensure rapid convergence. It has been shown that if the set of data points is close to the fitting contour, then the convergence achieved is nearly quadratic. It was therefore required that in order to minimise the CPU utilisation on the robot, an initial guess should be provided by a fast and non-iterative procedure. After further research, it was decided that the fast and non-iterative approximation to the LSF, provided by algebraic fitting algorithms, seemed ideal for solving this problem.

Algebraic fitting algorithms are different from their geometric equivalents in that they attempt to minimise the sum of the squares of the algebraic errors rather than minimising the distances of the sum of the squares of the geometric distances. If the above equations are considered.

$$F = \sum_{i=1}^{n} [(x_i - a)^2 + (y_i - b)^2 - R^2]^2 = \sum_{i=1}^{n} (x_i^2 + y_i^2 + Bx_i + Cy_i + D)^2 \quad (4)$$

where $B = -2a$, $C = -2b$ and $D = a^2 + b^2 - R^2$. Differentiating the equation with respect to B, C and D yields the following system of linear equations

$$M_{xx}B + M_{xy}C + M_xD = -M_{xz} \quad (5)$$

$$M_{xy}B + M_{yy}C + M_yD = -M_{yz} \quad (6)$$

$$M_xB + M_yC + nD = -M_z \quad (7)$$

where M_{xx}, M_{xy} etc. define moments, for example $M_{xx} = \sum_{i=1}^{n} x_i^2$ and $M_{xy} = \sum_{i=1}^{n} x_iy_i$. The system can be solved using Cholesky decomposition (or any other method for solving a linear system of equations) to give B, C and D, which in turn gives parameters a, b and R that define the circle. The algorithm used on the robot varies slightly from this definition, in that it uses a gradient

Fig. 7. Circle fitting to the ball

weighted [9, 10] form of the algorithm. This algorithm is commonly considered a standard for the computer vision industry [11, 12, 13] due to its statistical optimality. An example of a circle fitted on the robot can be seen in Figure 7. The circle fitted in the image is resilient to noise caused by the obstruction (the edge of the image), which would have previously affected both the distance and elevation values of the resultant ball.

4.3 Results

A comparison of the distances returned by examining the height of the object, the new circle fitting method and the actual distances are shown below in Figure 8. The ball is approximately 75% obstructed by the robot in each measurement.

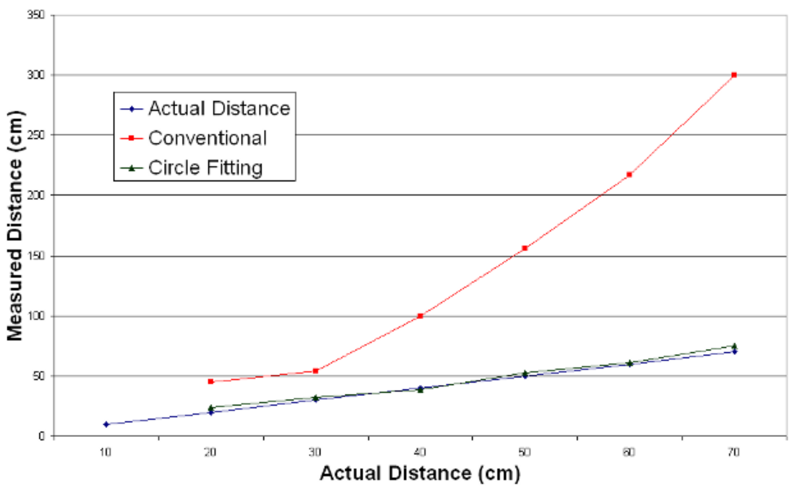

Fig. 8. Comparison of ball distances with 75% obstruction

5 Summary

The implementation of real-time object recognition systems for use on a robot is a complex task. It was observed throughout this project that while complex algorithms exist for image processing, many of these are not appropriate for use on a robot.

This study focussed on the development of edge detection and data association techniques to identify field landmarks in a controlled environment. These techniques were designed with efficiency as their key requirement so as to function effectively despite processor limitations. It has also been shown that a common method - circle fitting - can be scaled for use on a robot with a slow processor. Additionally, the circle fitting algorithm is able to operate in conjunction with existing Four-Legged League vision systems

Hence while many conventional algorithms are unsuitable for use on the robot, appropriate algorithms have been developed or adapted for real-time visual processing, greatly assisting in the identification of key objects and in turn robot localisation.

Acknowledgements. We would like to thank all past and present members and contributors to the University of Newcastle Legged League RoboCup team.

References

1. RoboCup Four-Legged League web site. `http://www.openr.org/robocup/index.html`
2. Burkhard, H.-D., Düffert, U., Hoffmann, J., Jüngel, M., Lötzsch, M., Brunn, R., Kallnik, M., Kuntze, N., Kunz, M., Petters, S., Risler, M., v. Stryk, O., Koschmieder, N., Laue, T., Röfer, T., Spiess, Cesarz, A., Dahm, I., Hebbel, M., Nowak, W., Ziegler, J.: GermanTeam 2002, 2002. Only available online: `http://www.tzi.de/kogrob/papers/GermanTeam2002.pdf`
3. Bunting, J., Chalup, S., Freeston, M., McMahan, W., Middleton, R.H., Murch, C., Quinlan, M., Seysener, C., Shanks, G. Return of the NUbots! The 2003 NUbots Team Report, `http://robots.newcastle.edu.au/publications/NUbotFinalReport2003.pdf`
4. Tardós, J.: Data Association in SLAM, `http://www.cas.kth.se/SLAM/Presentations/mingo-slides.pdf`
5. Levenberg, K.: A Method for the Solution of Certain Non-linear Problems in Least Squares, Quart. Appl. Math. Vol. 2, 1944, 164-168.
6. Chernov, N., and Lesort, C.: Least Squares Fitting of Circles and Lines, `http://arxiv.org/PS cache/cs/pdf/0301/0301001.pdf`
7. Marquardt, D.: An Algorithm for Least Squares Estimation of Non-linear Parameters, SIAM J. Appl. Math., vol. 11, 1963, 431-441.
8. Shakarki, C.: Least-squares Fitting algorithms of the NIST algorithm testing system, J. Res. Nat. Inst. Stand. Techn., vol. 103, 1998, 633-641.
9. Turner, K.: Computer perception of curved objects using a television camera, Ph.D. Thesis, Dept. of Machine Intelligence, University of Edinburgh, 1974.
10. Sampson, P.D.: Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm, Comp. Graphics Image Proc., vol. 18, 1982, 97-108.
11. Taubin, G.: Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined By Implicit Equations, With Applications To Edge And Range Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. 13, 1991, 1115-1138.
12. Leedan, Y. and Meer, P.: Heteroscedastic regression in computer vision: Problems with bilinear constraint, Intern. J. Comp. Vision, vol. 37, 2000, 127-150.
13. Chojnacki, W., Brooks, M.J., and van den Hengel, A.: Rationalising the renormalization method of Kanatani, J. Math. Imaging & Vision., vol. 14, 2001, 21-38.