Webcam-based Mobile Robot Path Planning using Voronoi Diagrams and Image Processing

Shahed Shojaeipour[1,1], Sallehuddin Mohamed Haris[1], Elham Gholami[2] and Ali Shojaeipour[2]
Dept. of Mechanical & Material Engineering[1], Dept. of Computer Software Engineering[2]
Universiti Kebangsaan Malaysia[1], Islamic Azad University of Shirvan[2]
Bangi, Selangor, 43600, Malaysia[1], Shirvan University Ave, Iran[2]
Malaysia[1], Iran[2]
shojaei@vlsi.eng.ukm.my, salleh@eng.ukm.my, gholami_e@yahoo.com, ali.shojaeipour@yahoo.com

Abstract: - In this paper, we present a method to navigate a mobile robot using a webcam. This method determines the shortest path for the robot to transverse to its target location, while avoiding obstacles along the way. The environment is first captured as an image using a webcam. Image processing methods are then performed to identify the existence of obstacles within the environment. Using the Voronoi Diagrams VD(s) method, locations with obstacles are identified and the corresponding Voronoi cells are eliminated. From the remaining Voronoi cells, the shortest path to the goal is identified. The program is written in MATLAB with the Image Processing toolbox. The proposed method does not make use of any other type of sensor other than the webcam.

Key-Words: - mobile robot, Path planning, Voronoi Diagrams, Image processing, Visual servo.

1 Introduction

Image processing is a form of signal processing where the input signals are images such as photographs or video frames. The output could be a transformed version of the input image or a set of characteristics or parameters related to the image. The computer revolution that has taken place over the last 20 years has led to great advancements in the field of digital image processing. This has in turn, opened up a multitude of applications in various fields, in which the technology could be utilised.

The aim of this paper is to present a method for visual servo control using only visual images from a webcam. Visual servo is the use of image data in closed loop control of a robot. Without doubt, today, the use of vision in robotic applications is rapidly increasing. This is due to the fact that vision based sensors such as webcams are falling in price more rapidly than any other sensor. It is also a richer sensor than traditional ranging devices, particularly since a camera captures much more data simultaneously [1].

Images can be captured by camera, and subsequently, processed using some particular software. Among them, MATLAB, with its Image Processing toolbox, is well suited to perform such tasks. Information obtained from the image processing exercise can then be used to generate motion commands to be sent to the mobile robot. This sequence is depicted in Fig. 1. Consequently, the robot imitates human vision in stages as follows:

- Image acquisition
- Image processing
- Image analysis and assimilation
- Image intelligence
- Control signal reception
- Motion control of parts of the robot

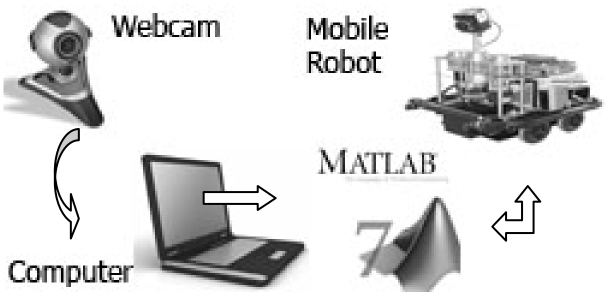
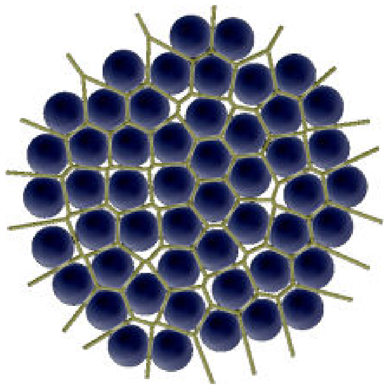


Fig. 1 Experimental Setup.

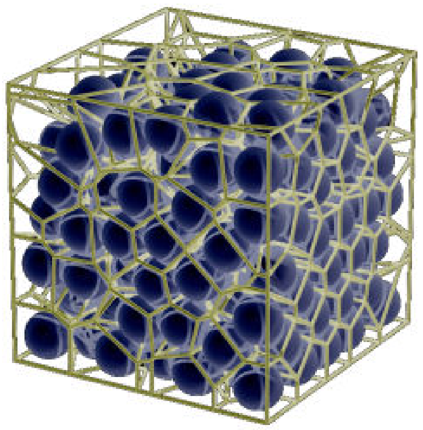
The Voronoi tessellation, originally proposed by Georgy Voronoi in 1907[2], is a special decomposition of a metric space based on the distances of points to a specified discrete set of sites within the space. For a set S containing i number of sites in Euclidean space, the tessellation is defined by associating a cell to each site. Cell i contains all points that are closest to si and the cell boundaries are hyperplanes made up of points that are equidistant to two or more sites in S (Fig. 2). The Voronoi diagram (VD(S)) perfectly partitions the space, and it has found use in the sciences for solving problems that involve the assignment of space between groups of objects. In general, the interested reader should refer to the book written in 0.

Many methods have been developed and used by different researchers to compute Voronoi tessellations.

For example, in 0, the Voronoi C tree data structure was introduced to generate generalized 3D Voronoi diagrams. In 0, morphological operations were used to transform image data obtained from sensors into its corresponding VD(S) where the skeletal lines represent obstruction free paths. There also exist a number of software packages readily available for computing the Voronoi tessellation. These include, for example, the software package 0 which can compute VD(S) in arbitrary dimensions and the Voronoi function in MATLAB. The program in 0 which is well-known for mesh generation via Delaunay Triangulation also computes Voronoi tessellations.



A



B

Fig. 2 (A) the 2D Voronoi tessellation (B) the 3D Voronoi tessellation.

Mesh computation codes can compute VD(S) of single objects which, given a set of points, will return the complete mesh dividing the space containing the set of sites into a mesh of cells as seen in figure 3. However, in practical applications, it is more natural to associate a single VD(S) with each particle, and compute them separately. This makes it easier to compute just a subset of Voronoi meshes, or to tailor the computations to handle special cases and complex boundary conditions. This also makes it straightforward to compute mesh-based statistics, such as mesh volumes, or the number of faces per mesh 0, 0.

2 Visual Obstacles

The path to be transverse by the robot must be ensured to be free of obstacles. For this, their existence must be identified and their positions located. This section describes how this could be done.

The image is recorded by a webcam which is installed above the robot. The image is then sent via a USB cable to a PC, to be processed by MATLAB. The experiments were carried out using a computer with 3.60GB free space hard disk and 1GB RAM memory. The algorithm was developed using MATLAB (version 7.6 R2008a).

The image is divided into segments, which become the export databases; usually they are the raw pixels data abstracted from the captured image [10], [11] and [12]. The picture could be in JPG or BMP format, in which case, every pixel point uses three numerical values, representing intensity levels of the primary colours: red, green and blue (RGB) to depict its characteristics. Therefore, in such format, computing workload to perform image processing would be very high. Hence, it would be desirable to convert the coloured picture into a grayscale image [13], [14].

Image processing methods are firstly used to identify the existence of obstacles within the image frame. This is implemented in an eight step MATLAB (with the Image Processing Toolbox) program. The following describes the steps:

Step 1. Generate input video objects.

This can be implemented using the command

Obj = videoinput('adaptorname', device name, 'format');

where the adaptor name can be determined using the 'imaqhwinfo' command, device name is the name given to the device and format refers to the required image format.

Step 2. Preview the webcam video image.

Preview('object name');

where object name is Obj in the last command

Step 3. Set brightness level of image.

set(obj,' property name' property value);

Step 4. Capture still image from webcam video.

getsnapshot(object name);

The captured image is stored as an array whose elements represent the light level.

Step 5. Remove the input device from memory.

delete(object name);

Step 6. Convert from RGB to grayscale mode.

I=rgb2ind(I,colorcube(150));

Step 7. Find edges of objects in the image.

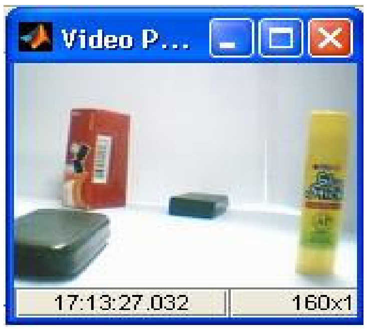
I=edge(I,'sobel',(graythresh(I)*.1));

Step 8. Remove noise.

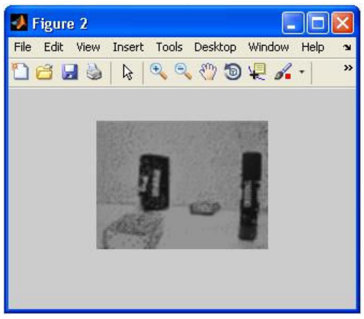
Se90=strel('line',3,90); Se0=strel ('line',3,0); I=imdilate(I,[se90 se0]);

I=imfill(I,'holes');

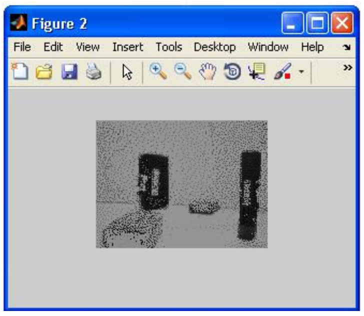
The results of running this program can be seen in Fig. 3, where (A) is the original image, (B)-(F) are the intermediate stages of the image and (G) is the final image.



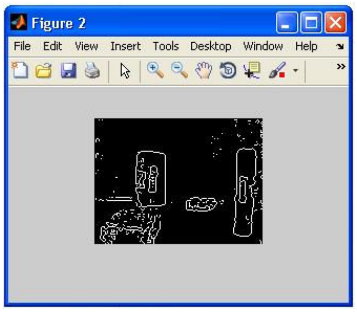
(A)



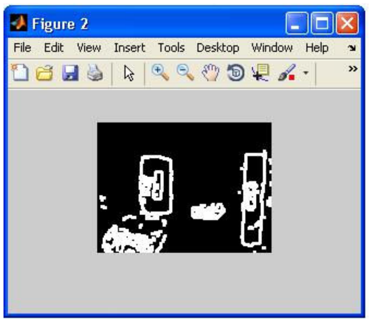
(B)



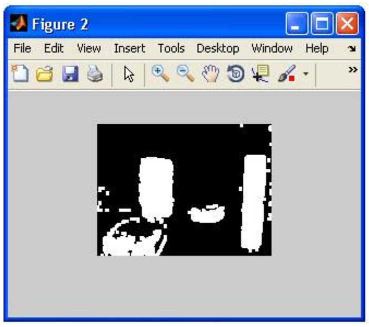
(C)



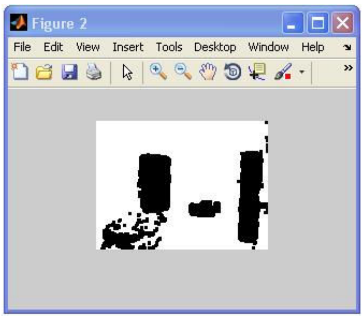
(D)



(E)



(F)



(G)

Fig. 3 (A) Original Image, from B to F- intermediate stages of filtering and (G) Final image.

3 Path Planning via Voronoi Diagrams

In order to select the shortest path for the robot to transverse, three actions need to be considered: the first is image capture using webcam, the second is to convert the image into a 3D scene, using the Spectral Fractal Dimension (SFD) technique [15] and [16]. Previous works have used just two images of the scene for this purpose [17], [18] and [19]. The third action is to use cell decomposition to identify and eliminate paths that are obstructed. The robot would then be able to chose the shortest of the remaining paths. Fig. 4 illustrates the path finding problem, where three paths are being considered. Then,

- Path(1) is not possible (encounter obstacle)
- Path(2) is the shortest distance and possible
- Path(3) is possible but it isn't the shortest path

Obviously, the number of objects and the number of paths would vary, depending on situation. We will focus on using a general program to find the shortest path between the robot and the target [20].

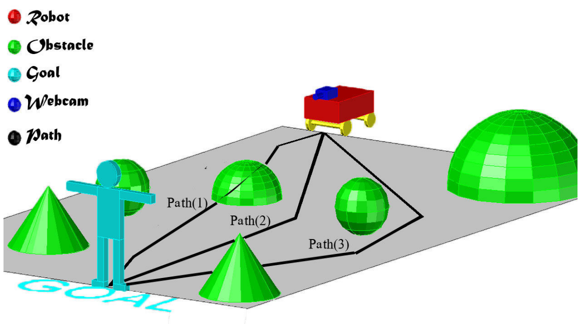


Fig. 4 Analyzed Paths.

In order to send the correct control commands to the robot such that the chosen path is followed, the precise position of objects must be measured. Feedback is the comparison of the target and actual positions, and is a natural step in implementing a motion control system. This comparison generates an error signal that may be used to correct the system, thus yielding repeatable and accurate results. [21]
For data transfer to and from the robot, the MATLAB Instrumentation Control toolbox is used. Data transfer may be performed via GPIB, VISA, Serial, TCP/IP or UDP interface.

3.1 Generation of Voronoi Diagrams:

VD(S) are constructed by first performing the Delaunay Tessellation which is regarded as the dual to Voronoi Tessellations. Firstly, any two sites p, q for which there exists a circle C that passes through p and q and doesn't contain any other site of S in its interior or boundary, are connected by a line segment. The set of such line segments form the edges of the Delaunay Tessellation DT(s), called Delaunay edges 000. Now, bisection of the Delaunay edges by another set of line segments results in the formation of Voronoi cells where each cell contains one site enclosed by the line segments forming Voronoi edges.
An example is depicted in figure 3, where VD(S) is depicted by solid lines and DT(S) by dashed lines. Note that a Voronoi vertex need not be contained in its associated face of DT(S). The sites p, q, r, s are co-circular, giving rise to a Voronoi vertex v. Consequently, its corresponding Delaunay face is boarded by four edges 0.

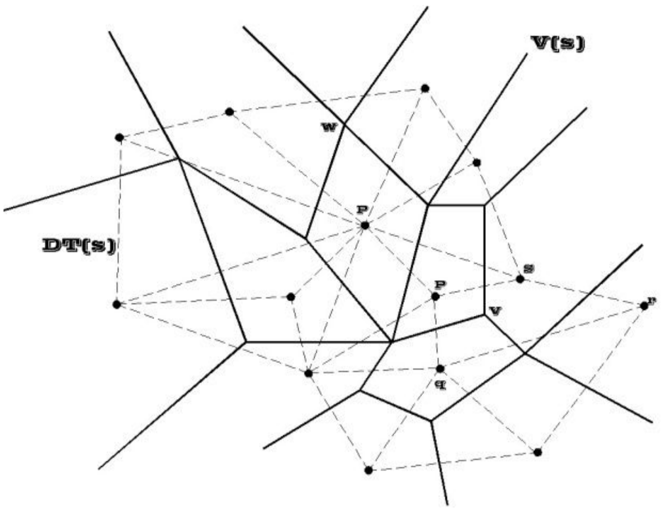


Fig. 5 Voronoi Diagram and Delaunay Tesselation.

3.2 Motion Planning using VD(S):

Application of Voronoi diagrams to mobile robot motion planning has been discussed in 0. The concept is reproduced in the following excerpt.
Suppose that for a disc-shaped robot centered at some start point, *s*, a motion to some target point, t, must be planned in the presence of n line segments as obstacles.
We assume that the line segments are pair wise disjoint, and that there are four line segments enclosing the scene, as shown in figure 6. While the robot is navigating through a gap between two line segments, l_1 and l_2, at each position *x* its "clearance ", i.e. its distance

$$d(x, l_i) = \min\{d(x, y) : y \in l_i\} \qquad (1)$$

to the obstacles, should be a maximum. This goal is achieved if the robot maintains the same distance to either segment. In other words, the robot should follow the bisector $B(l_1, l_2)$ of the line segments l_1 and l_2 until its distance to another obstacle gets smaller than $d(x,l_i)$.
Roughly, this observation implies that the robot should walk along the edges of the Voronoi diagram VD(S) of the line segments in $S = \{ l_1, \ldots\ldots, l_n\}$. This diagram is connected, due to the four surrounding line segments.
If start and target points are both lying on VD(S), the motion planning task immediately reduces to a discrete graph problem: After labeling each edge of VD(S) with its minimum distance to its two sites, and adding *s* and *t* as new vertices to VD(S), a breadth first search from *s* will find, within O(n) time, a path to t in VD(S) whose minimum label is a maximum. If this value exceeds the robot's radius, a collision-free motion has been found.
If the target point, *t*, does not lie on VD(S), we first determine the line segment *l(t)* whose Voronoi region contains *t*. Next, we find the point *z(t)* on *l(t)* that is closest to *t*; see figure 6. If its distance to *t* is less than the robot's radius then the robot cannot be placed at t and no motion from *s* to *t* exists. Otherwise, we consider the ray from *z (t)* through *t*. It hits a point t' on VD(S) which serves as an intermediate target point.

Similarly a point *s'* can be defined if the original start point, *s*, does not lie on VD(S) 0.

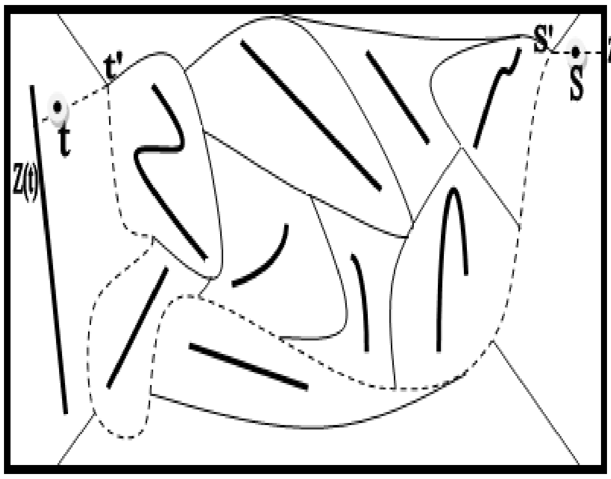


Fig. 6 The Motion Planning in Voronoi Diagrams Algorithm.

4 Conclusion

This paper provides a framework for mobile robot navigation using a robot mounted webcam. Using images captured by the webcam, the location of obstacles are identified. Then, using the Voronoi Diagrams VD(s) technique, the shortest path to the target destination is determined. Future work would be to translate the generated optimal path into input commands to the actual robot. The system would also be further developed for situations with moving obstacles and moving targets.

References:

[1] J. Campbell, R. Sukthankar, Nourbakhsh, I. Sukthankar and A. Pahwa, A robust visual odometry and precipice detection system using consumer-grade monocular vision. Proc. ICRA2005, Barcelona, Spain. 2005.

[2] G. Voronoi. Nouvelles applications des param`etres continus `a la theorie des forms quadratiques, Journal f¨ur die Reine und Angewandte Mathematik 133, pp. 97–178, 1907.

[3] A. Okabe, Barry Boots, K. Sugihara, and S. N. Chiu," Spatial tessellations: concepts and applications of Voronoi diagrams," John Wiley & Sons, Inc., New York, NY, 2000.

[4] I. Boada, N. Coll, N. Madern and J.A. Sellares," Approximations of 3D Generalized Voronoi Diagrams," EWCG 2005, Eindhoven, March 9-11, 2005.

[5] S.Garrido, L.Moreno, M. Abderrahim and F. Martin," Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," Int. Conference on Intelligent Robots and Systems Oct, 2006.

[6] Qhull code for convex hull, Delaunay triangulation, Voronoi diagram, and halfspace intersection about a point, http://www.qhull.org/.

[7] Triangle: A two-dimensional quality mesh generator and Delaunay triangulator, http://www.cs.cmu.edu/_quake/triangle.html

[8] E. Masehian and G. Habibi," Robot Path Planning in 3D Space using Binary Ineger Programming," International Journal of Mechanical System Science and Engineering ISSN 1307-7473, 2008.

[9] G. Habibi, E. Masehian, M.T.H. Beheshti," Binary Integer Programming Model of Point Robot Path Planning," The 33rd Annual Conference of the IEEE Industrial Electronics Socitey IECON ,Nov 5-8 , 2007.

[10] R. Gonzalez, R. wood," Digital Image Processing" 2nd edition. Prentice-Hall Inc, 2002.

[11] A. Jain," Fundamentals of Digital Image Processing". Prentice-Hall Inc, 1989.

[12] R. Duda, E. Peter Hart and D. Stork." Pattern Classification "2nd edition. John Wiley & Sons, Inc, 2000.

[13] G. Blanchet, M. Charbit," Digital Signal Image Processing Using MATLAB" ISTE Ltd, 2006.

[14] L. Xingqiao, G. Jiao, J. Feng, Z . Dean," Using MATLAB Image Processing to Monitor the ealth of Fish in Aquiculture". Proceeding of the 27th Chinese Control Conference July 16-18, Kunming, Yunnan China , 2008.

[15] H. Akbar, A.S. Prabuwono," Webcam Based System for Press Part Industrial Inspection" .IJCSNS International Journal of Computer Science and Network Security, VOL.8 NO.10, October, 2008.

[16] G. Modesto, M. Medina, D. Baez-Lopez," Focusing and Defocusing vision system (SIVEDI)". International Conference on Electronics, Communications and Computers (CONIECOMP 2005) IEEE, 2005.

[17] K.T. Jenn,"analysis and application of Auto focusing and Three-Dimensional Shape Recovery Techniques based on Image Focus and Defocus". PhD Thesis SUNY in Stony Brook, 1997.

[18] S.K. Nayar, M. Watanabe, M. Noguch," Real-time focus range sensor". Intl. Conference on Computer Vision, PP.995-100, June, 1995.

[19] M. Subbarao," Spatial-Domain Convolution/ Disconsolation Transform". Technique Report No.91.07.03,Computer Vision Levorotatory, State University of New York , Stony Brook, NY 11794-2350

[20] S.Shojaeipour, S.M.Haris and M.I.Khairir, " Vision-based Mobile Robot Navigation using Image Processing."international visual Informatic

conference (IVIC'09) , Springer lectuer Note in computer science LNCS-5857, Nov, 2009.

[21]Ch. Aung , H. Lwin , Y. K.T, M. Myint ," Modeling Motion Control System For Motorized Robot Arm using MATLAB". PWASET VOLUME ISSN 2070-3740 32 August, 2008.

[22]E. Masehian and G. Habibi," Robot Path Planning in 3D Space using Binary Ineger Programming," International Journal of Mechanical System Science and Engineering ISSN 1307-7473, 2008.

[23]S.W. Bae and K.Y. Chwa," Shortest Paths and Voronoi Diagrams with Transportation Networks Under General Distances," Springer – Verlag Brilin Heideberg, 2005.

[24]F. Aurenhammer, R. klein," Voronoi Diagrams," Resarch Concerning Voronoi diargrams, www.pi6.fernuni-hagen.de/publ/tr198.pdf