Journal of Engineering Sciences, Assiut University, Vol. 35, No. 4, pp. 975-994 , July 2007

3D INFORMATION EXTRACTION USING REGION-BASED DEFORMABLE NET FOR MONOCULAR ROBOT NAVIGATION

Khaled M. Shaaban and Nagwa M. Omar
Electrical Engineering Department, Assiut University, Assiut, Egypt

(Received June 18, 2007 Accepted July 15, 2007)

This paper proposes a new method to extract the objects' 3D information for monocular robot navigation. The proposed method is based upon the Region-Based Deformable Net (RbDN) technique that we developed in [1]. This technique is modified to segment any real time video sequence captured from a single moving camera. Instead of deforming a single contour, typically used with other deformable contour methods, RbDN technique deforms a planner net. The net consists of elastic polygons that represent the segmented regions' boundaries. The deformation process tracks the location change of the polygons and their vertices across the frames. The 3D information of each object's corner is extracted based on the location change of the corresponding vertex. Furthermore, the change in the area of each region across the frames is used to accurately extract the average depth of the surface corresponding to that region. The algorithm is completely autonomous and does not require user interference, training or pre-knowledge. The experimental results demonstrate the capability of the algorithm to extract the objects' 3D information with high accuracy within a reasonable time.

KEYWORDS: *Machine Vision, Robot Navigation, Landmarks, Objects 3D Information Extraction, Monocular Vision, Stereo Vision, Correspondence Problem, Deformable Contours.*

1. INTRODUCTION

Machine Vision as a technique for providing navigation information has been receiving attention since the early 80^{th} [2-5]. This attention could be explained by the observation that most animals depend upon their vision system for navigation. This observation is true for animals ranging from insects like bees up to almost intelligent animals like monkeys. Studies have suggested that these animals use visual landmarks as navigation aides [2, 3].

Navigation based upon self-measurements like odometer for moved distance and compass for angles leads to accumulative error in the final position. This error grows with time until the robot completely loses orientation. Observing landmarks then estimating the position relative to them does not suffer from this error accumulation. As a confirmation for this fact, consider a man walking in the desert with no landmarks, it is impossible for him to maintain a straight heading. Furthermore, unexpected obstacles may appear in the target path, which may require dynamic

navigation around them. From these observations it seems natural to seek navigation using Machine Vision.

Calculating the 3D information of scene objects relative to the position of the camera is essential for navigation. Two basic vision techniques for extracting this information are available. One technique is Monocular Vision [5-9], in which the 3D information is extracted from a sequence of images acquired under a relative motion of the camera. The other is Stereo Vision [10-12], in which the 3D information is obtained from two separate views of the same scene. Stereo Vision accuracy decreases rapidly with the increase of the distance of the object compared to the baseline distance separating the two views. For example, during car driving the length of the baseline separating the two eyes of the driver is negligible when compared to the distance of the faraway cars. Therefore there is no difference between the two images acquired by the two eyes and consequently no stereo vision. The estimation of the distance in this case must depend upon a monocular vision strategy. As another support to the suggestion that monocular vision is enough for navigation, a person with one eye can still walk around without bumping into things.

Monocular Vision navigation requires tracking of different regions as they change position across the frames in the sequence. This paper proposes a Deformable Contour Method (DCM) for accomplishing this tracking. DCMs are energy minimizing techniques that deform a single contour under the influence of internal and external forces [13-19]. The internal forces impose the contour smoothness and the external forces attract the contour to the object boundary. DCMs try to minimize the integration of these forces around the contour. Although DCMs are usually used for tracking a single region, the Region-Based Deformable Net (RbDN) that we developed in [1] automatically segments all the regions in the image. Furthermore the deformation process tracks the changes in shape and location of these segmented regions across the frames. These changes are used to estimate the distances of the objects corresponding to these regions. Due to the small time separating successive frames, tracking the change in the image is relatively easy when compared with the classical feature matching usually necessary in stereo vision systems. This ease allows for the real time performance necessary for robotic application.

The rest of this paper is organized as follows: Section 2 provides a review for the RbDN technique. Section 3 describes the use of the RbDN technique to segment a video sequence. Section 4 explains using the RbDN technique to extract the objects 3D information. Section 5 shows some of the experimental results. Section 6 concludes this work.

2. RBDN TECHNIQUE

As mentioned earlier, the heart of the proposed method is using a deformation technique for continuous tracking of the various regions in the image. The RbDN technique that we developed in [1] is modified to be used for this purpose. Unlike other deformable contour techniques, RbDN deforms a planner net that covers the entire image. This net consists of a group of vertices that symbolize the regions corners. The vertices are connected by edges without crossing each others forming elastic polygons

(contours) that represent the segmented regions' boundaries. The following sections will give more details about this method.

2.1 Net Structure

In order to fully understand the RbDN technique, a mathematical formalism is needed. The net is simply a plane graph, $Net = (V, E)$, that consists of a group of vertices, V, connected by edges, E. Each vertex, $v_i \in V(Net)$, is represented by a point in the Euclidian plane, $v_i(x, y)$, where x and y are Euclidian distances from an origin at the center of the Net. Each edge, $e_i \in E(Net)$, is represented by a line segment that connects two vertices, $e(v_i, v_j)$, i.e. $E \subseteq [V]^2$. For the rest of this work the term edge will be used to represent this defined mathematical meaning and will not be used to indicate a point with high value of gradient in the image. Nontrivial network covers a limited area of the Euclidian plane that is referred to as Q.

The plane graph has a unique characteristic: it can be sketched on a piece of paper in such a way that no edges meet in a point other than the common ends (the vertices). The following few restrictions are added to the general definition of the planer graph to form the definition of the *Net*:

- The Net has vertices at the corners of Q, to identify the Net extent. These vertices are connected with edges to surround Q. These edges form the outer boundary of the *Net*.
- The set of edges, $E(Net)$, could be partitioned into subsets, such that each subset, p_k, represents a ***polygon*** within Q. The edges within each polygon are ordered such that the interior of the polygon is always on the right hand side of the edges. Note that, each edge contributes in exactly two polygons except the edges at the outer boundary of the Net. The sequence of edges, $\{e_1, e_2, \cdots e_f \mid e_i \in p_K\}$, could be represented by an ordered set of vertices. Therefore, we can rewrite the polygon as $p_k = \{v_1, v_2, \cdots, v_f\}$ which signify that, each pair (v_i, v_{i+1}) is an edge in, p_k.The pair (v_f, v_1) represents the last edge in the polygon, p_k. Each polygon covers an area of Q that we call, $\mathrm{A}(p_k) \subseteq Q$. These areas are not mutually exclusive, that as $\mathrm{A}(p_i) \cap \mathrm{A}(p_j)$ does not necessary represent a zero area. A polygon can contain another polygon within its area.
- Except for very special networks, there is a large number of ways in which a network can be partitioned into polygons. A unique partitioning is to use polygons with the smallest possible area. That is to minimize the overlapping of polygons.

Therefore, the Net represents a way to partition the space, Q, into set of polygons, $P(Net)$. In other words the polygons resample the pieces of a puzzle that when fitted together form the full area, Q. At this point we need to refine the notation of the net to be, $Net = (V, E, P)$.

Given a real life image, I, and a $Net = (V, E, P)$ with extent, Q, that has the exact same dimension of the image, we can overlay the Net over the image. Each

Polygon of the Net, $p_k \in P(Net)$, or the difference of two or more polygons represents a segment of the image. Therefore, we can consider the Net as a formal mathematical notation to represent a segmentation of an image. This mathematical representation is necessary to introduce the concept of deformation to the process of image segmentation. One can easily imagine the process of deformation as the process of adjusting the location of the vertices (the corners of the polygons) to coincide the segments in the image. The mathematical description of the segments as a Net, provides the language to describe the different deformation operations like, inserting a new vertex into a polygon or merging two polygons to form a single larger one.

The general structure of the proposed net is illustrated through simple example shown in Figure (1). As shown in this figure the image under segmentation has three regions R_1, R_2 and R_3. The first region, R_1, is represented by one polygon, $p_1 = \{v_1, v_5, v_6, v_7\}$, while R_2 is represented by two polygons $p_2 = \{v_2, v_3, v_4, v_7, v_6, v_5\}$ and $p_3 = \{v_8, v_9, \ldots\ldots, v_{23}\}$, the area of $R_2 = \mathrm{A}(p_2 - p_3)$, the third region, R_3, is represented by p_3.

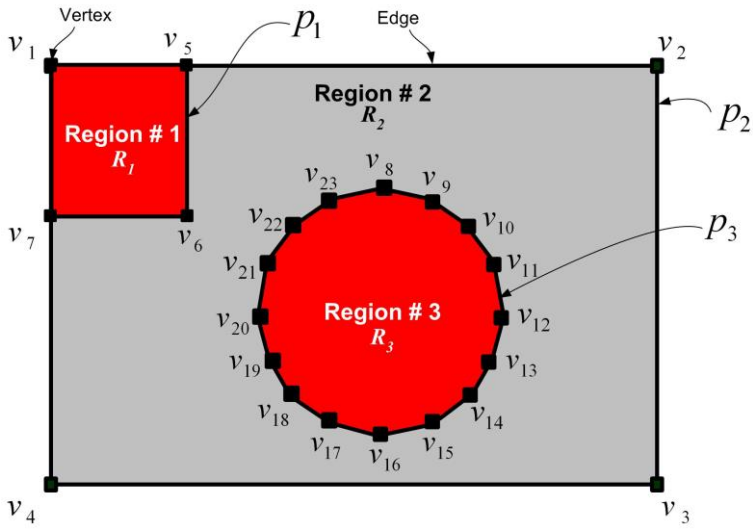


Figure 1: Segmentation example clarifies the net structure.

2.2 Net Deformation

The proposed net is automatically initialized to fully cover the real life image, I. That is the corner vertices that define Q should coincide the image corners. The proposed net can have arbitrary initial structure but we choose the simple one illustrated in Figure (2). As shown in this figure the net extent, Q, is partitioned into equal sized squares.

The net deforms under the effect of forces generated around the common edge between every adjacent polygon pair. The average color of each polygon in the pair

and the color of the pixels around the common edge, generate these deformation forces. Each polygon searches a thin area outside its boundary for pixels with color that are close to its average color. If considerable number of such pixels is found, the polygon attempts to inflate itself to include these pixels. We call these thin areas the sensitivity regions. Naturally the forces of the neighboring polygon oppose this inflation and the system settles at the equilibrium of all these forces.

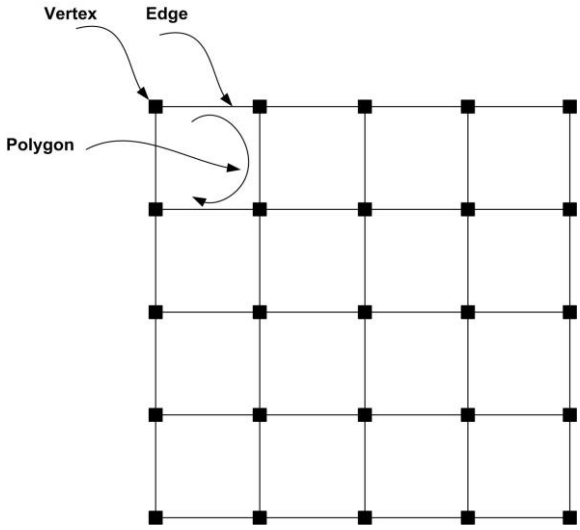


Figure 2: The initial shape of the proposed net, equaled size squares.

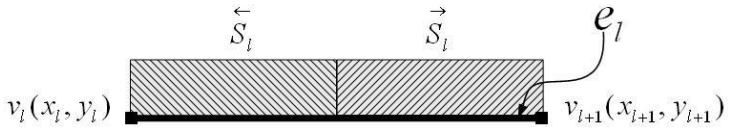


Figure 3: Edge, e_l, surrounded by two sensitivity regions. Left and right sensitivity regions are represented by $\overleftarrow{S_l}$ and $\overrightarrow{S_l}$ respectively.

The left hand side (outside) of every edge in each polygon contains two non overlapped sensitivity regions as shown in Figure (3). For the edge, e_l, these sensitivity regions are denoted $\overleftarrow{S_l}$ and $\overrightarrow{S_l}$. Each sensitivity region is a rectangular area having a height of w and width equals to half of edge length. To understand how the forces are generated consider the arrangement shown in Figure (4).

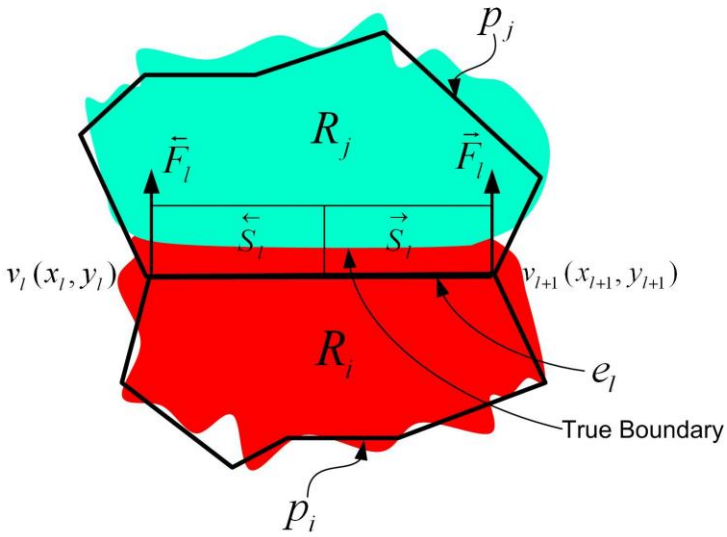


Figure 4: A part of the proposed net shows forces affect edge e_l from the point of view of p_i.

In the figure, there are two adjacent regions having different colors, R_i and R_j, and two polygons, p_i and p_j, that are not aligned over the regions. The two polygons cover image areas, $\mathrm{A}(p_i)$ and $\mathrm{A}(p_j)$ and their respective colors averages are represented by $C(p_i)$ and $C(p_j)$. The edge separating the two polygons does not coincide with the true boundary separating the two regions forming alignment disparity. From the point of view of p_i, this disparity is measured by the number of pixels within each of its sensitivity regions $\overleftarrow{S}_l$ and $\overrightarrow{S}_l$ that satisfy the following conditions:

1. The pixel ρ is located within the area of the neighboring polygon, $\rho \in \mathrm{A}(p_j)$.
2. The color distance between the pixel color and its current polygon color is large, $ColorDist(\ C(\rho), C(p_j)\ ) > \eta$. That is, the pixel should not belong to this region based on the color distance.
3. The distance between the pixel color and the neighboring polygon color is small, $ColorDist(\ C(\rho), C(p_i)\ ) \le \eta$.

Where,

$C(\rho)$: The color vector of the pixel ρ.

$ColorDist(C_1, C_2)$: A measurement of color dissimilarity between two color vectors, C_1 and C_2.

η: The color distance threshold.

We denote such alignment disparity measure $H(\overleftarrow{S_l})$ and $H(\overrightarrow{S_l})$ respectively. A small value of $H(\overleftarrow{S_l})$ and $H(\overrightarrow{S_l})$ represents a good fit of the edge e_l. The deviation from this state leads to the deformation forces:

$$\overleftarrow{F_l} = \frac{H(\overleftarrow{S_l})}{2\lambda} \tag{1}$$

$$\overrightarrow{F_l} = \frac{H(\overrightarrow{S_l})}{2\lambda} \tag{2}$$

Where, λ : The length of the edge.

From the point of view of p_j (not shown in the figure), there is no color mismatch under its sensitivity regions and thus no opposing forces.

In general any vertex v_k is a member in a set of polygons ζ_k. In each polygon, this vertex connects exactly two edges each generates forces that affect its position. Thus, the number of forces that affect the vertex v_k is $\psi_k = 2\zeta_k$, see Figure (5).

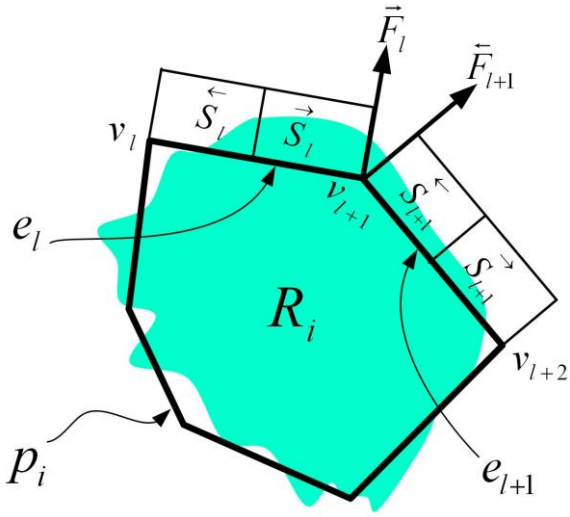


Figure 5: A part of the proposed net shows forces affect vertex v_{l+1} due to its existence in p_i.

These forces are arbitrary oriented and are treated as real forces. They are added as vectors to generate the total force, F_k^T, that affecting the vertex v_k,

$$F_k^T = \sum_{i \in \psi_k} F_i \tag{3}$$

F_k^T could be decomposed into two components one in the x direction that we denote F_k^{Tx} and the other in the y direction that we denote F_k^{Ty}. These components are the

best estimation of the position change needed to enhance the fit of the polygon edge over the region boundary, that is:

$\Delta x_k = F_k^{Tx}$: The total deviation of the vertex v_k in the x direction. (4)

$\Delta y_k = F_k^{Ty}$: The total deviation of the vertex v_k in the y direction. (5)

Therefore the position update rule could be written as:

$$L(v_k) + (\Delta x_k, \Delta y_k) \rightarrow L(v_k) \quad (6)$$

Where, $L(v_k)$: The Euclidian location of the vertex, v_k .

A complete round of vertices adjustment forms a single deformation cycle. Usually more than one cycle is needed to get good results.

2.3 Net Maintenance

During the deformation process situation that requires special treatment may arise. The system periodically checks and handles these situations to keep the net simple. The most import situations and the way to handle them are as follows:

Polygon merge: If during the deformation, two neighboring polygons with almost the same average region colors emerge, they should be merged in order to reduce the overall number of the polygons. Assume that these two polygons colors averages are represented by $C(p_i)$ and $C(p_j)$ and if $ColorDist(C(p_i), C(p_j)) <= \eta$ then p_i and p_j should be merged.

There is another type of polygon merging that depends on the polygon size. Polygons with very small area (smaller than 200 pixels) are merged to one of its neighbors. The neighbor to be merged with is the one with minimum color distance (to the polygon to be deleted) regardless of the magnitude of this distance.

Vertex deletion: There are three states that require deleting a vertex in order to minimize the overall number of vertices. These states are:

1. Two edges that almost lie on the same line.
2. Small length edges that have a negligible effect on the net shape.
3. Spike (thorn) edges, the edges which enclose small angle.

Vertex Insertion: Since there is no prior knowledge about the regions' shapes, the optimum number of vertices for each specific polygon is not known. Therefore, and during the deformation process a polygon with less than adequate number of vertices may arise. The solution for such case is the vertex insertion operation. Figure (6) shows an edge, e, that needs vertex insertion to enhance its fit. As shown in the Figure, the two alignment disparity measures of this edge from the point of view of the polygon p_i are $H(\overleftarrow{S_l})$ and $H(\overrightarrow{S_l})$ and from point of view of p_j are $H(\overleftarrow{S_t})$ and $H(\overrightarrow{S_t})$.

In this arrangement the force due to $H(\overleftarrow{S_l})$ is balanced with the force due to $H(\overrightarrow{S_t})$ and the force due to $H(\overrightarrow{S_l})$ is balanced with the force due to $H(\overleftarrow{S_t})$. That is, the overall forces affecting e are small but the quality of the fit is not good. This special balance state could be easily detected by observing that the overall small forces are not accompanied with small value of its alignment disparity measures. If any of the measures is above a specific limit, δ, then there is a need for a new vertex. The

insertion operation is performed by breaking the edge e into two edges then re-indexing the vertices in the polygon.

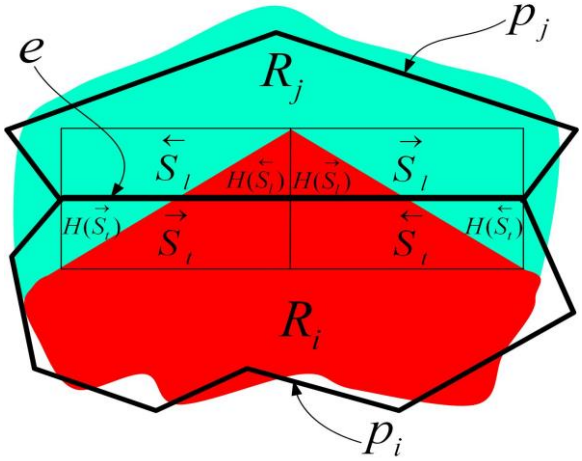


Figure 6: Condition at which a vertex should be inserted.

The net deformation and the maintenance cycles are repeated periodically until a good fit is reached. Stopping the iteration process depends upon the maximum displacement over of all the vertices in the net. If this displacement is under a specific preset value the algorithm stops.

RbDN technique automatically segments the entire image into a small number of regions in a compact mathematical form represented by the net. This net is rich with topological and other information about the regions and their shapes that are useful for other Vision algorithms especially image sequence analysis.

3. SEGMENTING VIDEO SEQUENCES

The RbDN technique as described in Section 2 is intended for still image analysis. It needs two modifications to be useful for analyzing image sequences. The first modification seeks increasing the analyses speed by using the result of each frame as a starting point for the next one. The idea is that, the minimum changes between the successive frames require smaller number of deformation cycles for convergence. This modification considerably shortens the processing time leading to the real-time performance necessary for monocular vision navigation.

The second modification adds to the algorithm the capability to handle any extreme scene changes. After convergence and as the robot movies new objects may enter the filed of view generating new regions in the image. Accordingly, the algorithm should be able to inject new polygons into the net. The need for new polygons is detected by observing the filling factors of the regions. The new object appearance increases the off-pixels and consequently decreases the filling factor. In this case the region with a small filling factor is fragmented into smaller regions. The deformation process then regroups these smaller regions constructing considerable size regions

ready for deformation. The fragmentation process could be considered as a local reinitialization for the region with lower filling factor.

4. 3D Information Extraction

Extracting the objects' 3D information requires the solution of two problems. The first is the correspondence problem, in which the corresponding features are to be matched between the image pair [10]. The second problem is utilizing the locations of the corresponding features to get the required 3D information using triangulation [5]. The accuracy of the extracted 3D information highly depends upon the baseline distance between the points of view of the two images. Using longer baseline distance increases the extracted 3D information accuracy. Unfortunately it also increases the search space leading to a more complex matching process. Therefore, the 3D information accuracy and the complexity of solving the correspondence problem are conflicting factors.

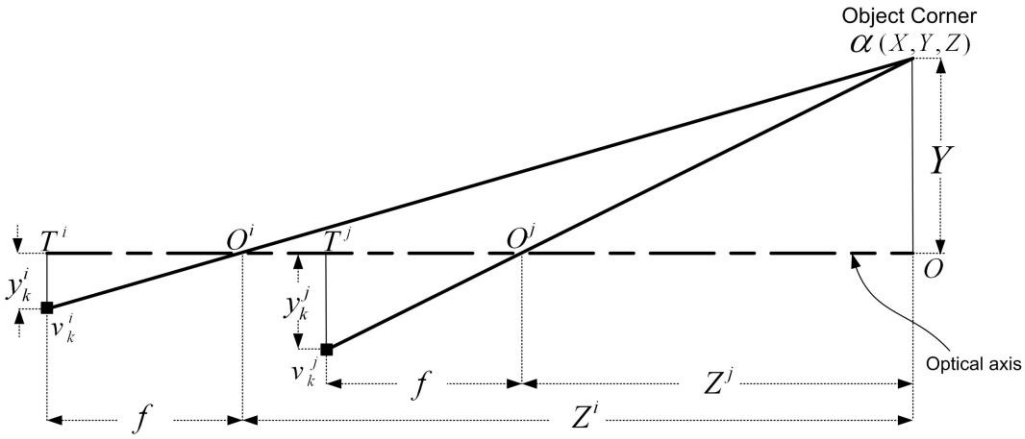
In monocular vision, these conflicting factors can be treated easily using an image sequence [9]. To get a good accuracy two frames separated by a significant ground distance are used. These frames are not consecutive frames but separated by a sequence of intermediate ones. Matching feature between the first and the final frames would be a complex process because of the extended search space. Instead, tracking the changes of the objects' features through the intermediate image sequence, as described in Section (3), provide a simpler alternative. The location of vertices and the regions are continuously adjusted for each new intermediate frame using the deformation process. Therefore the correspondence of the vertices between the first and the final frames is readily available after deformation. That is, tracking the vertices through the intermediate frames is used instead of the complex feature matching to solve the correspondence problem.

The second step is utilizing the corresponding feature locations and the baseline distance to get the 3D information. As will be illustrated in the next sections two techniques are suggested to perform this operation: the Vertex-Based Extraction method and the Area-Based Extraction method.

4.1 Vertex-based Extraction Method

Vertex-Based extraction method aims to obtain the 3D information of the objects corners using the locations of the corresponding vertices. This work uses the standard triangulation technique [5] described by the geometric model shown in Figure (7).

As the camera moves from position O^i to O^j two frames are taken which are denoted r^i and r^j. A specific corner α of a certain object is represented by the vertex $v_k^i(x_k^i, y_k^i)$ in the net of frame r^i, the vertex location changes during the net deformation to be $v_k^j(x_k^j, y_k^j)$ in frame r^j. This change of the vertex location is the bases used to get the depth information. Note that since the robot moves on a horizontal plane the distance, Y, between the object point, α, and the optical axis is constant. Under this assumption the triangulation operation could be simplified as follows:



O^i and O^j : The position of the camera at frames r^i and r^j respectively.
Z^i and Z^j : the depth of the object's corner, α, at frames r^i and r^j respectively.
f : The focal length of the camera lens.

Figure 7: The geometric model of extracting 3D information from single moving camera.

Comparing the similar triangles $O^i\alpha O$ and $O^i v_k^i T^i$, we get

$$\frac{Y}{Z^i} = \frac{y_k^i}{f} \tag{7}$$

Similarly, from the similar triangles $O^j\alpha O$ and $O^j v_k^j T^j$, we get

$$\frac{Y}{Z^j} = \frac{y_k^j}{f} \tag{8}$$

But

$$Z^j = Z^i - \Delta Z \tag{9}$$

Where, ΔZ : The moving distance in the Z direction between the two captured frames (baseline distance).
Solving these three equations we get,

$$Z^i = \Delta Z\left(\frac{y_k^j}{y_k^j - y_k^i}\right) \tag{10}$$

Substituting Y and y_k^i by X^i and x_k^i respectively in Equation (7), also, substituting Y and y_k^j by X^j and x_k^j respectively in Equation (8), the X coordinates of the point α can be found as follows:

$$X^i = \frac{x_k^i Z^i}{f} \tag{11}$$

$$X^j = \frac{x_k^j Z^j}{f} \tag{12}$$

The Y coordinates of the point α could be found from Equations (7) or (8).

By applying Equations (7-12), the 3D information, (X, Y, Z), of the objects' corners could be determined from the corresponding vertices location.

Further analysis can be applied on Equation (10) to calculate the sensitivity of the vertex-based extraction method. From the equation we get,

$$\tau_V = \frac{\Delta y}{\Delta Z} = \frac{y_k^j}{Z^i} \tag{13}$$

Where, Δy : The change in the y value of the vertex.

τ_V : The change in the y value of the vertex compared to the baseline distance (sensitivity).

For a faraway objects, Z is much larger than y, leading to a lower sensitivity value, τ_V. Also, from the equation the sensitivity is affected directly by the y value in the image plane. That is, the points near the optical axis, with a smaller y value, have a lower sensitivity leading to inaccurate depth estimation. From the symmetry, the same principle can be applied to the points with small x value. Therefore we could conclude that the sensitivity of the Vertex-Based method is small for the points that are close to the center of the filed of view if the displacement of the camera is parallel to the optical axis. This problem could be handled using the Area-Based Extraction method described in the next section.

4.2 Area-Based Extraction Method

The motion of the robot changes the camera point of view and consequently the projection area of the objects on the image plane. As the robot moves towards an object, its apparent area in the image increases. Knowing the moved distance of the robot (the baseline distance) a good estimate of the object distance from the camera could be obtained.

In the image plane a region and its area are denoted, R_k and $A(R_k)$ respectively. Due to the linear relationship between the object and the image plane dimensions, the area, $A(R_k)$, is proportional to the inverse of the distance square. That is:

$$A(R_k) \propto \frac{1}{Z^2} \tag{14}$$

Where, Z is the average depth of the region, R_k. For two captured frames r^i and r^j the following relationship could be derived:

$$\frac{A^i(R_k)}{A^j(R_k)} = \frac{(Z^j)^2}{(Z^i)^2} \tag{15}$$

Substituting Z^j by $Z^i - \Delta Z$ in Equation (15) we get,

$$Z^i = \frac{\Delta Z}{1 - \sqrt{\frac{A^i(R_k)}{A^j(R_k)}}} \tag{16}$$

Since the moved distance of the robot, ΔZ, and the area of the region in the two images $A^i(R_k)$ and $A^j(R_k)$ are available, the average depth of the object surface could be obtained using Equation (16),

To obtain the sensitivity of the Area-Based extraction method, substitute $A^i(R_k)$ by $A^j(R_k) - \Delta A(R_k)$ in Equation (16) then we get,

$$\Delta A(R_k) = 2\frac{\Delta Z}{Z^i} A^j(R_k) - \left(\frac{\Delta Z}{Z^i}\right)^2 A^j(R_k) \tag{17}$$

Where, $\Delta A(R_k)$: The change in the area value of region, R_k.

For the small value of $\Delta Z / Z^i$, the second term in the right hand side of Equation (17) could be neglected. Consequently we get,

$$\tau_A = \frac{\Delta A(R_k)}{\Delta Z} \approx 2\left(\frac{A^j(R_k)}{Z^i}\right) \tag{18}$$

Where, τ_A: The change in the area value compared to the baseline distance (sensitivity).

Comparing the sensitivities of the Area-Based and the Vertex-Based extraction methods, as given in Equations (18) and (13) respectively, one can notice that: the sensitivity in Equation (18) is proportional to the area but in Equation (13) the sensitivity is proportional to the y value only. Thus, for surfaces with reasonable areas the sensitivity using the Area Based Method is higher than that of the Vertex Based method which results in a more accurate depth estimation. This sensitivity enhancement is more vivid for objects near the optical axis of the camera. Unlike, Vertex-Based, the Area-Based extraction method is used mainly to calculate the average depth of the object surface not for extracting the 3D information of the object corners. This average depth is important for robot navigation especially for objects still at long distance from the current robot position.

In monocular vision navigation, the camera is usually pointing forward to collect information regarding the robot path. In such case objects near the center of the filed of view are more important than other objects. Using the Vertex-Based extraction method to obtain the depth information in this case leads to poor results. The Area-Based extraction method is a more practical alternative. The depth measurement enhancement for such monocular configuration is the main contribution of this work.

5. Experimental Results

To test the algorithm a simple mobile Robot was designed and constructed as shown in Figure (8). The robot carries a PC that is dedicated to the navigation purposes with the following specification: 3GHz, 512 MB of Ram running MS Windows XP. A stander webcam is connected to the PC using USB 2 connection. The camera is mounted at the front of the robot such that robot motion is parallel to the optical axis of the camera. The captured bitmap images are with size 320x240 pixels only, to keep the execution time reasonable. The robot locomotion is controlled by a microcontroller. The wheel is equipped with encoders to measure the traveled distance within 0.5 cm accuracy. This platform is used to capture the image sequences for test purposes. The extracted 3D

information from the proposed system is used for navigation. The details of the navigation process are beyond this work.

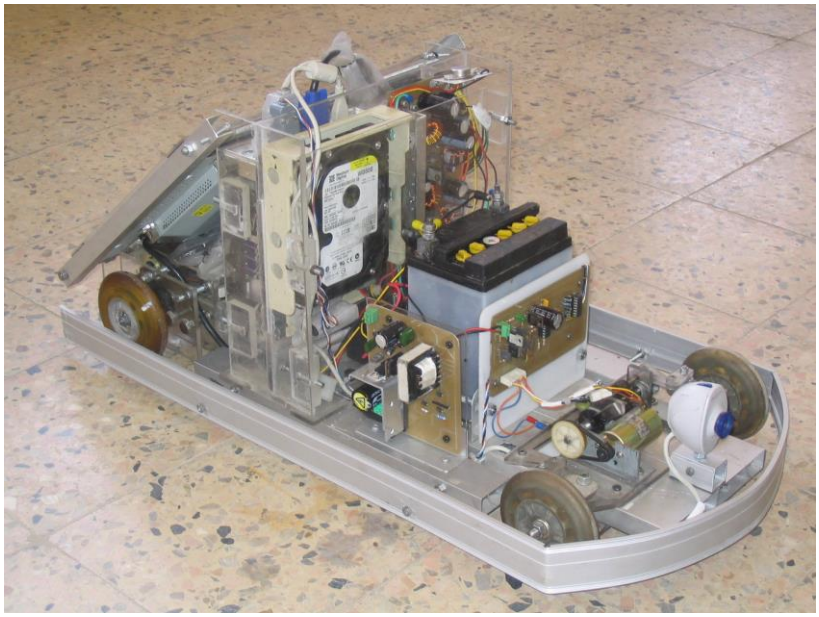


Figure 8: A simple mobile Robot designed and constructed to carry out the experiments

The first experiment is performed to compare Vertex-Based and Area-Based extraction methods. In this experiment the first and the final frame are taken from two points of view separated by 10 cm as shown in Figure (9).

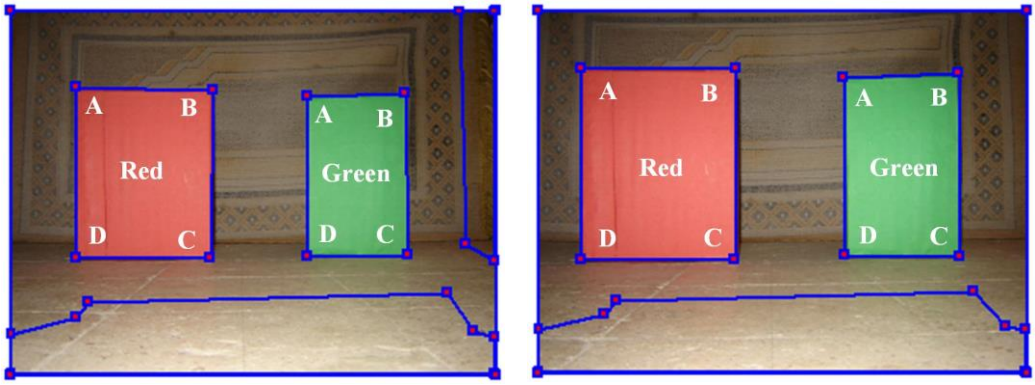


Figure 9: Two frames, baseline distance 10 cm.

In these images, there are two boxes with different sizes and colors. The apparent area of the red box is 530 cm^2 and for the green one it is 360 cm^2. RbDN technique segments the first frame as a still image with a good fitting in 0.18 second. The deformation process tracks the changes in the locations of the vertices and the regions from the first frame to the final frame in 0.13 second. The 3D information of the objects is measured using the Vertex-Based and the Area-Based extraction methods. The 3D information extraction time for both methods is negligible in comparison with the deformation time.

As given in Table (1), the errors in the estimated depths using the Vertex-Based extraction method are much higher than those using the Area-Based extraction method. The Vertex-Based extraction method gives poor results especially for vertices near the optical axis (error up to 306.1%). This could be explained if the sensitivity Equations (13, 18) are considered. The changes in the y values between the two frames are small when compared to the changes in the area values. Also from the Table, one can conclude that, for the Area-Based method the accuracy of the depth information increases with the increase of the objects' areas.

The second experiment is performed to test the ability of the Area-Based extraction method to determine the average depth of real objects having various sizes, shapes and depths. Figure (10), shows the starting and the ending frames used in the analysis. These frames are taken from two points of view separated by 5 cm. The RbDN technique segments the first frame in 0.2 second. Tracking the changes in the location of the vertices and the regions from the first frame to the final frame is achieved in 0.08 second. Due to the smaller baseline distance, the tracking time is small. The estimated average depths of the objects surfaces are reported in Table (2). As shown from the table, the errors are within 2% for all objects.

As mentioned before the standard stereo vision technique gives poor results with faraway objects. Monocular systems utilize the apparent larger baseline distance to provide better results for such objects. This experiment tests the ability of the proposed technique to extract depth information for objects at longer distances (7 meters). To test the effect of the baseline on the quality of the results, two values of the baseline length are used. That is, the depth information is extracted using images separated by 100 cm and 200 cm for comparison. As shown in Figure (11), the images contain two objects, pot and tree at distances of 712.5 cm and 725.0 cm respectively (relative to location # 1). The first frame is segmented using the RbDN technique in 0.17 second. The tracking process from the first frame to the second one and from the second frame to the third each took 0.15 second. The resulted average depths are illustrated in Table (3). The window average depth could not be calculated at location # 3 because a significant part of the window disappeared from the filed of view. As shown from the table the accuracy increases using larger baseline distance. Using baseline distance 200 cm decreases error to less than 0.2 %.

Table (1): Comparison between Vertex-Based and Area-Based Extraction methods to obtain the 3D information of objects illustrated in Figure (9).

Object	Points	Real Values (cm)			Vertex-Based Extraction Method							Area-Based Extraction Method						
					Estimated Values (cm)			Absolute Error %			$\Delta y/\Delta Z$ (Pixels) at $\Delta Z =$ 10 cm	Estimated Values (cm)			Absolute Error %			$\Delta A/\Delta Z$ (Pixels) at $\Delta Z =$ 10 cm
		X	***Y***	***Z***	***X***	***Y***	***Z***	***X***	***Y***	***Z***		***X***	***Y***	***Z***	***X***	***Y***	***Z***	
Red	**A**	-26	16.25	85	-19.7	12.1	65.72	24.23	25.53	22.6	13.05	-26.46	16.28	84.86	1.769	0.184	0.164	2779
	B	-5.5	16.25	85	-4.26	11.23	61.66	22.5	30.8	27.4	12.24	-5.73	16.13	84.86	4.181	0.738	0.164	2779
	C	-5.5	-8.7	85	-27.07	-40.6	345.2	392.1	366.6	306.1	1.3	-5.7	-8.98	84.86	3.636	3.218	0.164	2779
	D	-26	-8.7	85	-75.99	-25.8	224.8	192.2	196.5	164.4	1.97	-26.5	-9.1	84.86	1.923	4.597	0.164	2779
Green	**A**	8.5	15.2	85	5.67	10.58	62.5	33.29	30.39	26.47	11.94	8.02	15.029	84.25	5.647	1.125	0.882	1937
	B	23.5	15.2	85	16.86	11.43	64.63	28.25	24.8	23.96	11.99	22.887	15.429	84.25	2.608	1.506	0.882	1937
	C	23.5	-8.7	85	54.9	-20.28	185.2	133.6	133.1	117.8	2.31	23.214	-8.2	84.25	1.217	5.747	0.882	1937
	D	8.5	-8.7	85	24.25	-24.95	228	185.2	186.7	168.2	1.85	8.09	-8.43	84.25	4.823	3.103	0.882	1937

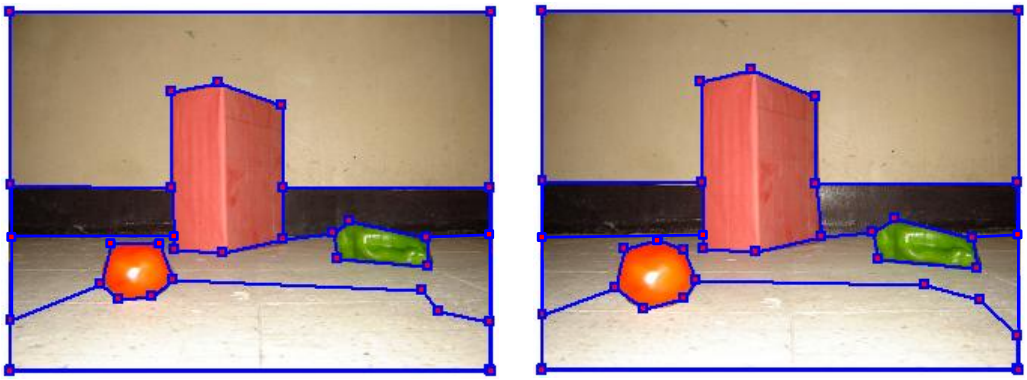


Figure 10: Two frames, baseline distance 5 cm.

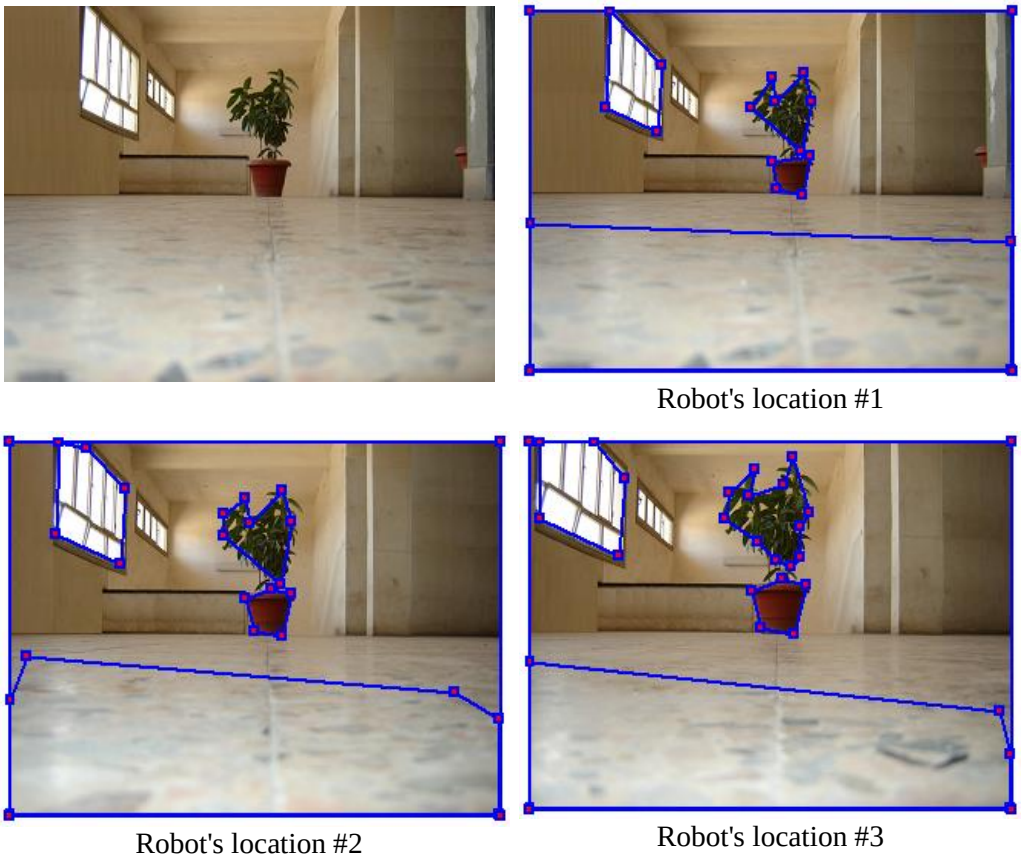


Figure 11: Three frames, baseline distance 100 cm.

Table (2): Average depths obtained for objects illustrated in Figure (10) using Area-Based extraction method.

Object	Real Average Depth (cm)	Estimated Average Depth (cm)	Error %
Box	92.33	94.4	2.2
Tomato	47	46.93	0.14
Pepper	68.5	67	2.1

Table (3): Average depths obtained for objects illustrated in Figure (11) using Area-Based extraction method.

Robot Location	Object	Real Average Depth (cm)	Estimated Average Depth (cm)	Absolute Error %
Location #2 Baseline 100 cm	**Pot**	**612.5**	**630**	**2.85**
	Tree	**625**	**640**	**2.4**
Location #3 Baseline 200 cm	**Pot**	**512.5**	**513.5**	**0.195**
	Tree	**525**	**526**	**0.19**

6. Conclusion

This work, proposes using the Region-Based Deformable Net (RbDN) technique for image sequence segmentation. It further proposes using the sequence segmentation results to obtain 3D information for the objects in the scene. This process is intended to be used for monocular vision navigation of mobile robots. RbDN technique is particularly suitable for this task. It deforms an elastic net that represents the contours of the different areas in the images as they change locations and/or shapes across frames. The correspondence of the areas and their vertices are automatically tracked which eliminates the need for solving the correspondence problem. From the corresponding position of the vertices, the objects' 3D information could be obtained using triangulation. As shown in the paper the estimation sensitivity for the points near the optical axis is small which leads to poor 3D results. To overcome this problem another method is proposed to get the average distance of the different surfaces of the objects. This method depends upon the changes in the areas of the regions as the camera moves to estimates the objects' distances. This Area-Based method is mathematically proven more accurate and experimentally provided better results.

7. References

1 K. Shaaban, and N. Omar, "Automatic Color Image Segmentation Using Deformable Net", Journal of Engineering Science, Assiut University, Egypt, vol. 35, no. 2, pp. 457-476, 2007.

2 M. Srinivasan, S. Zhang, M. Lehrer, and T. Collett, "Honeybee Navigation en Route to The Goal: Visual Flight Control and Odometry", Journal of Experimental Biology, vol. 199, pp. 237-244, 1996.

3 M. Srinivasan, M. Lehrer, W. Kirchner, and S. Zhang, "Range Perception Through Apparent Image Speed in Freely-Flying Honeybees", Visual Neuroscience, vol. 6, pp. 519-535, 1991.

4 G. DeSouza, and A. Kak, "Vision for Mobile Robot Navigation: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp.237-267, February 2002.

5 Y. Murphey, J. Chen, J. Crossman, J. Zhang, P. Richardson, and L. Sieh, "DepthFinder: A Real-time Depth Detection System for Aided Driving", Proceedings of IEEE Intelligent Vehicle Symposium, pp. 122-127, 2000.

6 V. Lepetit, and P. Fua, "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey", Foundations and Trends in Computer Graphics and Vision, vol. 1, no 1, pp. 1-89, 2005.

7 T. Repo, "Modeling of Structured 3-D Environments From Monocular Image Sequences", PhD, Department of Electrical and Information Engineering and InfoTech Oulu, University of Oulu, 2002.

8 D. Koller, K. Danilidis, and H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes", International Journal of Computer Vision, vol. 10, no. 3, pp. 257-281, 1993.

9 C. Tomasi, and T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method", Int'l J. Computer Vision, vol. 9, no. 2, pp. 137-154, 1992.

10 K. Yoon, and I. Kweon, "Adaptive Support-Weight Approach for Correspondence Search", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 650-656, April 2006.

11 M. Brown, D. Burschka, and G. Hager, "Advances in Computational Stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 993-1008, August 2003

12 P. Ho, and R. Chung, "Stereo-Motion with Stereo and Motion in Complement", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 2, pp.215-220, February 2000.

13 M. Niethammer, A. Tannenbaum, and S. Angenent, "Dynamic Active Contours for Visual Tracking", IEEE Transactions on Automatic Control, vol. 51, no. 4, pp. 562-579, April 2006.

14 M. Sakalli, K. Lam, and Y. Hong, "A Faster Converging Snake Algorithm to Locate Object Boundaries", IEEE Transactions on Image Processing, vol. 15, no. 5, pp. 1182-1191, 2006.

15 Ch. Chang, "Deformable Shape Finding with Models Based on Kernel Methods", IEEE Transactions on Image Processing, vol. 15, no. 9, pp. 2743-2754, 2006.

16 G. Foresti, and F. Pellegrino, "Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation", IEEE Transactions on Systems, Man, and Cybernetics, vol. 34, no. 3, pp. 325-333, 2004.

17 K. Shaaban, "Model Deformation Using Hit or Miss Operation", Journal of Engineering Science, Assiut University, Egypt, vol. 32, no. 1, pp. 471-484, 2004.

18 S. Sun, D. Haynor, and Y. Kim, "Semiautomatic Video Object Segmentation Using Vsnakes", IEEE Transactions on Circuits and Systems for Video

Technology, vol. 13, no. 1, pp. 75-82, Jan. 2003.

19 Y. Zhong, A. Jain, and M. Dubuisson-Jolly, "Object Tracking Using Deformable Templates", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 5, pp. 544-549, 2000.

استخلاص المعلومات الثلاثية الأبعاد باستخدام شبكة مُتَشَكِّلة تعتمد علي صفات المناطق لاستخدامها في ملاحة روبوت أحادى الرؤية

يقدم هذا البحث تقنية جديدة مبنية على التشكل للمنحنيات لإيجاد المسافات الحقيقية بين روبوت أحادى الرؤية والأشياء المحيطة به لاستخدامها في ملاحته ويتم استخلاص هذه المسافات من مجموعة صور متتالية يتم التقاطها على مسافات بينية بكاميرا وحيدة محمولة على الروبوت أثناء حركته للأمام والنظام المصمم بهذه التقنية يعمل أوتوماتيكيا ولا يحتاج إلى تدريب أو معرفة مسبقة بمكونات الصورة أو تدخل من المستخدم.

وتعتمد هذه التقنية على شبكة متشكلة لتجزئة الصور للحصول على الحدود لكل مناطقها والشبكة المستخدمة مكونة من مجموعة من رؤوس المضلعات موصلة معا بخطوط غير متقاطعة ومتقابلة فقط عند هذه الرؤوس و تشغل الشبكة مساحة من المستوى الإقليدى محدودة بخطوط خارجية موصِّلةً للرؤوس الممثِّلة لأركان هذه المساحة وكل مضلع من هذه الشبكة يمَثِّل رياضيا بمجموعة من الرؤوس المرتبة بحيث تكون المضلعات دائما على يمين الخطوط الموصلة لهذه الرؤوس وتشغل هذه المضلعات مساحات متباينة من المستوى الإقليدى واتحاد هذه المضلعات يكَوِّن المساحة الكلية للشبكة وهي مساوية لمساحة الصورة المراد تجزيئها وعند بسط الشبكة على الصورة تمَثِّل كل منطقة من مناطقها بمضلع واحد أو بالفرق بين عدد من المضلعات و يستخدم الخوارزم المقترح قوى يتم توليدها حول الخطوط المشتركة بين المضلعات بناء على تجانس توزيع اللون في مناطق الصورة لتشكيل الشبكة وتعمل هذه القوى على تحسين انطباق المضلعات على الحدود الحقيقية لأجزاء الصورة

ولاستخدام هذه الشبكة في ملاحة الروبوت أحادي الرؤية تقوم عملية التشكل بتتبع التغير في مساحات المضلعات ومواقع رؤوسها خلال الصور المتتابعة ويقدم هذا البحث طريقتين لاستخدام هذا التغير فى الحصول على المسافات الحقيقية بين الروبوت والكائنات المُمَثَّلة بهذه المضلعات وتستخدم الطريقة الأولى مقدار التغير في مواقع رؤوس المضلعات للحصول على الأبعاد الثلاثية لأركان الكائنات وتعانى هذه الطريقة من ارتفاع نسبة الخطأ فى إيجاد المسافات خاصة بالنسبة للنقاط الواقعة بالقرب من المحور البصري عندما تكون حركة الروبوت موازية لهذا المحور لذلك تم استنباط طريقة أخرى تستخدم التغير فى مساحات المناطق الممثِّلة لأسطح الكائنات خلال الصور المتتابعة للحصول على المسافات بين هذه الأسطح و الروبوت بدقة عالية وتعتبر هذه الطريقة أكثر ملائمة لملاحة روبوت باستخدام كاميرا موجهه للأمام حيث أن مواقع الكائنات القريبة من المحور البصري للكاميرا ذات أهمية خاصة لتجنب العوائق في عملية الملاحة وقد أثبتت التجارب مدى كفاءة الخوارزم المقترح في استخلاص المسافات بين الروبوت والكائنات المحيطة به خلال الملاحة.