another landmark and locate it as well, in order to transform the robot's local position into coordinates suitable for the global coordinate frame as defined by the experimenter. However, in case a landmark has not been found in the camera's optical range, the robot will revolve about itself with relative increments of 90° (knowing that the robot's optical range is about 100°). This rotation is terminated in either of the two situations:

- The robot subsequently detects, recognizes the landmark, and estimates the latter's local pose
- A complete turn has been performed, so the robot will proceed with the same rotational activity but at a radius of 1 meter from the previous position. This radius is incremented every time a complete turn is finished without finding a recognizable landmark.

This scenario realizes all the possible combinations that may ensue, when the robot is placed in a previously unknown environment. In this case, no action is conducted without a reasonable motive, thus emulating human capacity in wandering through a previously unseen setting.

In this overall layout, the robot will be required to perform landmark recognition/detection and self-localization. These processes will be described in the following sections.

B. Landmark Recognition and Detection

In the context of this real-time *"kidnapped robot problem"*, the robot is challenged by various environmental and technical issues that have to be resolved before a suitable localization method is applied.

The robot is equipped with a Canon VC4 coloured camera with a natural resolution of 768x480 pixels and an optical range of approximately 100°. On the basis of the available camera features, the CV application must be able to distinguish a suitable landmark, if available, and locate it within the processed image in a real-time situation. The main limitation is caused by variations in illumination, which require the robot to be insensitive to image fluctuations induced by temporal and environmental settings.

At start up, the robot is designed to use its frame grabber to periodically capture a medium resolution image of its surroundings to be stored as "raw" data. This is based on the assumption that the robot has prior knowledge of the specific colour and shape of the available landmarks, which in the initial stage are limited to three cardboard cylinders red, green and blue in colour. This choice of landmark type comes as a compromise between complex detection and localization. In the case of cylindrical landmarks, the robot can simply check for a rectangular shape of a certain colour to detect the presence of the landmark; however, the issue of localization becomes more elaborate due to the inability of the robot to distinguish its bearing towards the landmark from a single frame image. The need for accurate localization in the presence of a single camera obliges the application to require a pair of images of the same landmark at two distinct positions, which are somewhat far apart. The alternative approach to landmark selection is to facilitate landmark localization, while complicating landmark detection. In this method, a distinctive shape or feature can be drawn on these landmarks to be distinguished by the application. Due to the variation in shape, size, and perspective view, landmark detection becomes more involved; however, localization is possible with only one collected image.

The CV application commences operation by executing a stage of illumination normalization, which maintains a mean illumination value of 100 for the "raw" image. This ensures that the natural lighting affects neither landmark detection nor localization [4]. After illumination augmentation, this image is further processed for the specific colours of the given landmarks. This demands selective background elimination on the basis of chromatic properties (i.e. all blobs incomparable in colour to the landmark are whitened). This process can be done in numerous ways, which include the two developed so far. The first methodology requires that *each* pixel is compared to a "generic" range for landmark colour, which is distinctive of the landmarks themselves. The other approach makes use of the correlation between landmark pixels and their neighbours via a normalization correlation matrix, which is excerpted from a typical subsection of the landmark image. Although simpler in form and implementation, the first method requires pixel-to-pixel processing with no relation to what the surrounding pixels may yield. On the other hand, the second method is burdened with a relatively longer processing time than the first, yet it renders more details about the landmark, while eliminating the contrast factor from the images.

After chromatic background rejection, the "crude" image is analysed by a dilating morphological filter, which tends to remove noisy factors evident in the residual image. Also, it renders a slightly "smoother" contour of the landmark, if present, since it deals with expanding the majority colour within a sampling matrix traversing the whole image. In this regards, sharp and/or jagged edges are removed, thus, facilitating the corner/line detectors that are applied later.

Moreover, a simple median-smoothing filter is applied to the pre-processed image to remove any other insignificant noise elements due to random fluctuations. As a consequence, the resulting image after preliminary landmark recognition must be void of any colour range other than the one to be tested for (red, green, or blue). This colour analysis can be performed in a variety of methods depending on the preferred colour space to be used (i.e. Red Green Blue space [RGB], Hue Saturation Illumination space [HIS], or Chrominance Luminance space [ChLu]) [5]. The optimal colour space to be implemented should be able to perform image processing independently of the changing environmental factors. Despite the fact that most images are represented as RGB sequences, this colour space remains sensitive to lighting that may disturb the exact detection of a certain colour. For example, a coloured picture of a red landmark under bright light will represent a totally different RGB range than that of the same landmark but in normal lighting. As a better approach, either HSV or ChLu spaces can be used directly to eliminate lighting sensitivity. The resultant analysed image can be compared to the raw one taken in Figure 1 below.

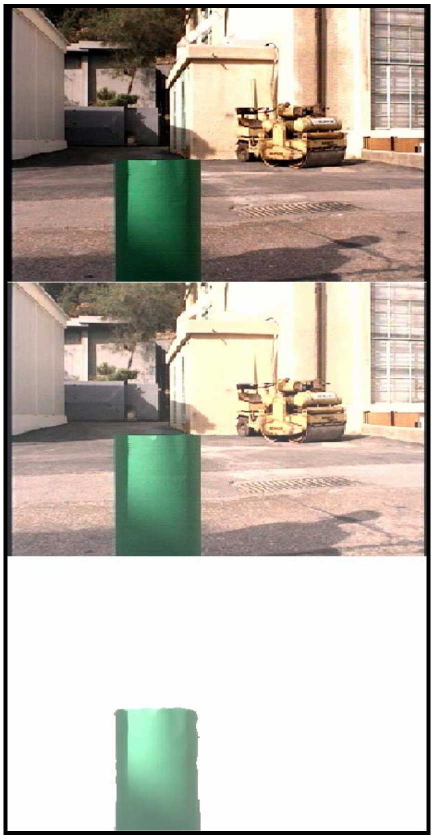

Fig. 1 Background elimination for a green colour landmark

The remaining problem to be resolved is also the most crucial for correct landmark detection. It has been shown that much of the essential information of an image is stored in the edge map of the image, and that edge structures have an apparent relevance in biological vision systems [9]. In addition, the edge information in an image tends to be robust under changes in illumination or related camera parameters. For these reasons, edge structure has been used extensively in computational vision. A variety of edge detectors are currently available including the edge detector proposed by Canny as published in the OpenCV library [6].

This algorithm is utilized as the basis for line detection used in landmark detection. In fact, the edge points located by the Canny algorithm are fitted to suitable lines according to certain user-defined threshold parameters and printed onto the processed image for visual verification using Hough line detection [9]. However, not all the found lines are appropriate for the purpose of landmark detection, since vertical lines (i.e. with a directional angle along the length of the image ranging from 90° to 45°) are the only selected candidates. This restriction is valid because it reduces the number of lines to be processed for landmark suitability later on, thus, making the process more compatible to real time situations. These induced lines (as seen in Figure 2) are analysed according to their respective directional vectors in the image plane to decide whether they are "parallel" up to a certain extent defined by another user-defined threshold. This takes into account the inaccuracies that may result from both the Canny and Hough methods and the relative slanting of the landmark itself with respect to the robot's camera.

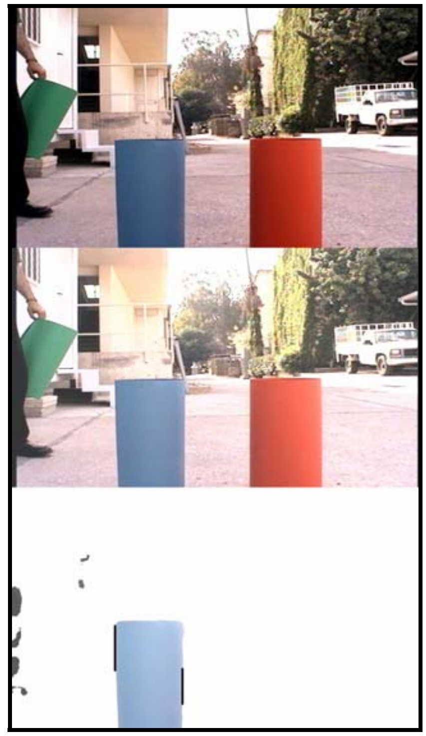

Fig. 2 Hough borders of the detected landmark

The average number of pixels that separate the two "parallel" lines farthest from each other is an essential input to the Chromatic Distance Estimation process, which is a heuristic approach that tends to approximate the distance between the robot and the landmark been visualized.

III. Self-Localization: Targeting and Triangulation

The localization process is subdivided into two sequential tasks, beginning with the determination of the first two landmarks with respect to the robot's local frame followed by world frame conversion. Those two scenarios are explained in details in the following section.

A. Targeting

In this procedure, two camera pictures are used, while assuming that the respective positions of the robot are obtained from odometry sensors. The robot angle is the angle by which the robot has tilted from the direction it was at the moment the application was executed.

Using Euclidian geometry based on the camera angles, the position of the landmark (the red circle in Figure 3) can be determined. In order to prevent errors due to approximating the robot as a point object, the separation between the center of mass and the camera was approximated to be 20 cm. This factor was crucial for calibrating the readings provided by the camera and those by the robot encoders.

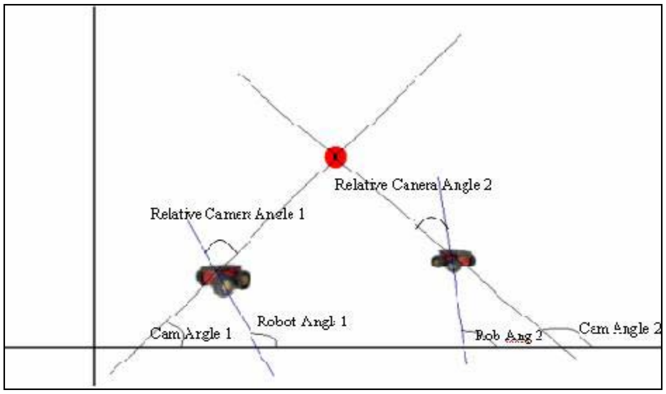

Fig. 3 Targeting the landmark from two distinct positions

B. *Triangulation Method for Localization*

After recording the local positions of the two landmarks via targeting, enough information suffices to estimate the global position of the robot. This task can be accomplished in three equivalent methods that calculate the robot's position with respect to each landmark in the global reference frame. The robot's position can be better estimated as the weighted average of the following methods that uniquely determine a triangle (see Figure 4):

1. Using two sides and the angle between them
2. Using three sides and no angles
3. Using one side and the two adjacent angles

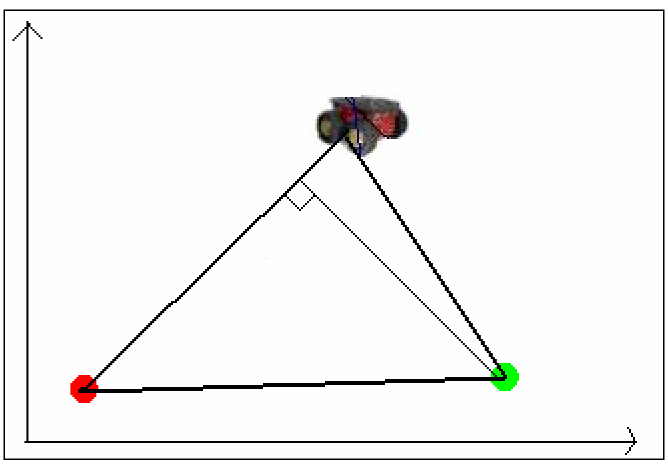

Fig. 4 Triangulation using two landmark positions

In fact, this triangulation method is the core functional component for our localization procedure, since it attempts to determine an efficient estimate of the robot's global position from the knowledge of the absolute locations of the landmarks in the world coordinate frame. However, it assumes that the robot's camera detects the same landmark in two different scenes or frames. In case this last condition is not fulfilled, the heuristic Chromatic Distance Estimation method is utilized as a means of providing an estimate of the distance between the robot and the landmark. This estimation makes use of prior experiments that were done offline by examining the number of pixels distinguishing the landmark (via colour referencing) at various distances from the camera.

After conducting multiple runs of tests at each distance, a discrete graphical representation of the experimental data can be extracted, which is used during the online localization process to determine the approximate distance to the landmark via linear interpolation. This method assumes proportionality in the correlation between the number of pixels and the distance separating the camera from the landmark. Even though this assumption is an approximation, yet it renders a simple and rapid method for distance estimation using computer vision. The following graph (Figure 5) shows the inversely proportional relationship between the number of pixels representing the landmark width and the robot-to-landmark distance.

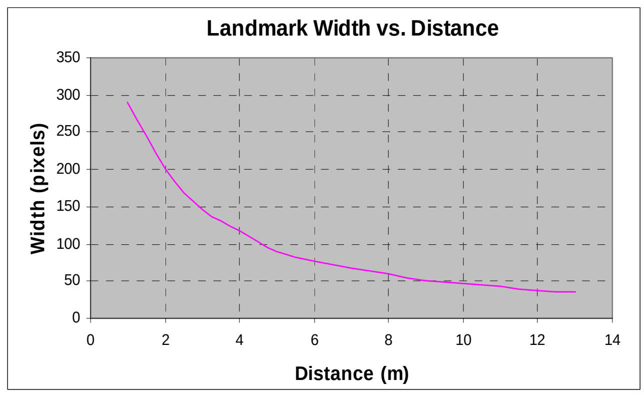

Fig. 5 Graph showing the inverse relationship between landmark width and distance from camera

As a consequence of this localization process, the robot must be able to transform any position in its initial local frame to the desired global frame, in which the landmarks are defined. For this purpose, an ideal transformation operator must be determined and updated to allow for this conversion to occur. This can be summarized by the following transformation matrix:

$$ {}^{W}_{L}T = \begin{bmatrix} {}^{W}_{L}R & & p_x \\ & & p_y \\ 0 & 0 & 1 \end{bmatrix} $$

In this expression, ${}^{W}_{L}R$ denotes the rotational matrix between both coordinate frames, while *(p_x, p_y)* represent the relative coordinates of the center of the world local frame with respect to the local one. This matrix is used to determine the following relation of position: ${}^{W}P = {}^{W}_{L}T\,{}^{L}P$, which transforms the position of the robot in the local frame (^{L}P) to that in the world frame (^{W}P). This transformation operator can be calculated using the given world coordinates of the two landmarks (i.e. W_1 and W_2) and their respective local coordinates (i.e. L_1 and L_2) that were previously estimated.

$$
{}^{W}_{L}T = \begin{bmatrix} \cos\theta & -\sin\theta & p_x \\ \sin\theta & \cos\theta & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} A & -B & p_x \\ B & A & p_y \\ 0 & 0 & 1 \end{bmatrix} \text{.........} \quad (1)
$$

$$
\textit{where} \quad \underline{X} = \begin{bmatrix} A \\ B \\ p_x \\ p_y \end{bmatrix} \textit{ is the solution matrix of the equation}
$$

$$
\underline{K}\,\underline{X} = \underline{b} \quad \textit{where}
$$

$$
\underline{K} = \begin{bmatrix} L_{x1} & -L_{y1} & 1 & 0 \\ L_{y1} & L_{x1} & 0 & 1 \\ L_{x2} & -L_{y2} & 1 & 0 \\ L_{y2} & L_{x2} & 0 & 1 \end{bmatrix} \textit{and } \underline{b} = \begin{bmatrix} W_{x1} \\ W_{y1} \\ W_{x2} \\ W_{y2} \end{bmatrix} \text{.........} \; (2)
$$

IV. Client-Server Application

The mobile servers constituting the underlying client server control architecture for the mobile platform, embodied in the Pioneer 3-AT Operating System software and found embedded on the robot's microcontroller, manage the low-level tasks of robot control and operation, including motion as well as acquiring sensor information and driving accessory components like the PTZ camera.

It is the job of an intelligent client running on a connected PC to perform the full gamut of robotics control strategies and tasks, such as obstacle detection and avoidance, localization, features recognition, mapping, PTZ camera control. Nearest the robot, ARIA's ArDeviceConnection class, at the command of your application code, establishes and maintains a communication channel with the robot server, packaging commands to ArRobotPacketSender and decoding responses ArRobotPacketReceiver from the robot in safe and reliable packet formats ArRobotPacket prescribed by the client-server protocols.

On the other hand, ACTS provides a server port for retrieving JPEG-compressed images. This is useful for remote viewing of the video data, but without needing the overhead of a large bandwidth. The SAV server has an associated client, which is supplied as part of the ACTS package. The server can be started automatically by the Linux system, but then it must be shut down manually if we are going to do frame grabbing on the system for other purposes (vision processing). Hence, we can not make use of the SAV application supplied by ACTS.

Our main challenge was to create a visual platform in order to best illustrate the operation and analysis of the frames captured by our robot. For that purpose, we implemented a client-server application in which the server runs on a Linux platform while the client operates in a regular WIN32 environment. The client is a graphical user interface (see Figure 6) illustrating the various analytical milestones in the camera's image processing domain. In that respect, our server was configured to run on the local host of the robot, communicating with a connected client through a TCP socket over a wireless network (WLAN). TCP sockets provide reliable connection and transfer of information and mask data interference from other servers operating on the network. The server's main operation is to send the logged images captured by the robot during its run. The server fetches the IplImage structure information for each image along with its analysis data and sends them sequentially to a remotely connected client.

The ACTS client communicates with the server (IP address: 192.168.1.2) through the TCP socket, over the WLAN network. Hence, the client retrieves the IplImage structure and opens a window showing the retrieved frames sequentially in order to create a strip of frames. Our client-server application provides a means to connect interactively to the robot running simultaneously with this application. It lets you compare the analysed images to the initial ones, while displaying relevant coordinates and landmark properties in a real-time setting.

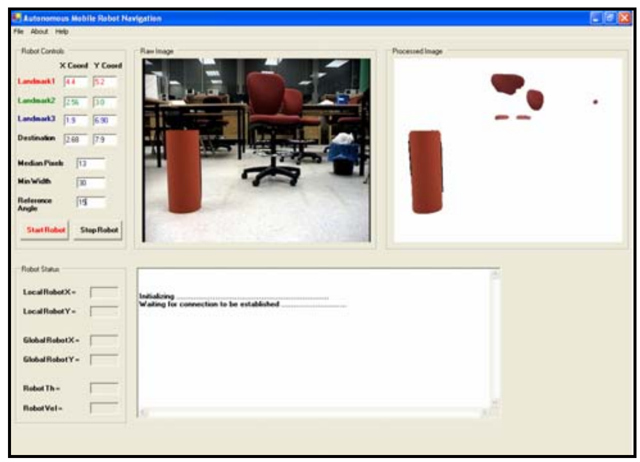

Fig. 6 Client-Server Interface

V. GIS Robot Tracking Application

A Geographic Information System (GIS) is a computer-based information system that enables the capture, modelling, manipulation, retrieval, analysis and presentation of geographically referenced data. The most important component in a GIS is the database. The value of the GIS database lies in the quality and usability of its data. As technology continues to advance, flexible options have become available that provide a wider range of applications and a host of opportunities to fit any particular functionality. In this perspective, GIS technology has found its way into many robotics applications.

A real-time GIS tracking package was implemented for the Pioneer 3-AT robot. This package is an autonomous real time GPS tracking system, which tracks the robot as it moves through along its trajectory. Using this package, we could track the robot (as indicated by triangular shapes and connected by blue lines in Figure 7) and be able to know its exact coordinates, velocity and the analysis that was generated during the specific run. In addition, a mapping option has been added to the overall functionality of the package whereby the obstacles sensed by the robot along its trajectory are plotted onto the map as small black squares to distinguish them from actual robot positions. A proposed extension to the performance of this package is to allow the user to specify the robot's destination, while the shortest path to the latter position is identified as computed using Dijkstra's algorithm.

Our package software provides different analysis and options that could be applied to help in tracking the robot at a certain time. Options available in this package include exporting the viewed map into jpeg and other compatible formats, printing a specific view of the map, loading shape files for previous runs, zooming, and panning facilities. Figure 7 illustrates a sample of the GIS software layout and interface.

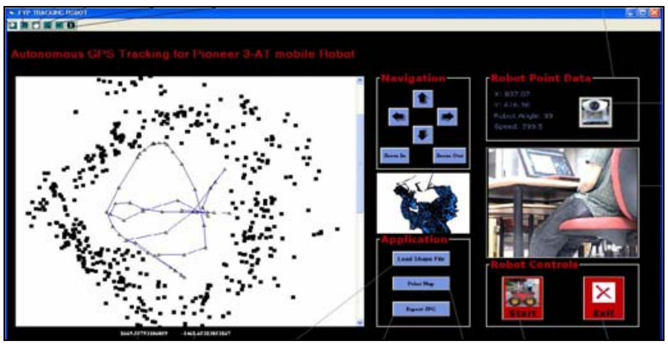

Fig. 7 Example of the robot trajectory traced by the described software

VI. Proposed Applications

The recent success of the Mars Pathfinder mission highlights the fact that autonomous robotic systems utilizing computer vision could be applied to demanding real-time applications. Clearly, the development of autonomous robots will be a significant factor in many domains of exploration of environments. Examples of such environments include outer space, the depths of the oceans, radioactive or contaminated sites or other extreme environments. Autonomous robots are already in use in automated delivery systems in some hospitals and warehouses. In order to render robots as an integral part of every household, solutions are required to be constructed, which are practical, efficient and cost-effective. We believe that the work presented here is a step in this direction.

VII. Future Work

During the research, a good deal of attention was paid to obtaining detected landmarks which were free of outliers. Future work would include proposing alternative methods for detecting the landmark. (This could include methods for selecting better or more appropriate thresholds that may be determined via online learning processes.) Additional extensions to the localization problem, in which neither robot nor landmark positions are known, form one of the other directions that could be followed by future research.

VIII. Conclusion

In this paper, we have presented a method for mobile robot localization using computer vision, whereby a robot placed in an unknown outdoor environment localizes itself and navigates successfully from source to destination while avoiding impeding obstacles. This method exhibits many advantages in terms of recent feature-based methods. This was achieved by exploiting and analyzing the strengths of several available alternatives. Experimental results indicate that the method is promising for practical, real-time and real-world implementation.

References

[1] Andrea Bonarnini, Paolo Aliverti and Michele Lucioni. *An Omnidirectional Vision Sensor for Fast Tracking for Mobile Robots*, IEEE Transactions on Instrumentation and Measurement, VOL. 49, NO.3, June 2000

[2] Andrew J. Davison, Yolanda Gonzalez Cid and Nobuyuki Kita. *Real-Time 3D SLAM with Wide-Angle Vision*, Proc. IFAC Symposium on Intelligent Autonomous Vehicles, 2004

[3] Andrew J. Davison. *Real Time Simultaneous Localisation and Mapping with a Single Camera*, ICCV 2003

[4] Dorian J. Spero and Ray A. Jarvis, *3D Vision for Large-Scale Outdoor Environments*, Proc. Australian Conference on Robotics and Automation, November 2002

[5] Guilherme N. DeSouza and Avinash C. Kak, *Vision for Mobile Robot Navigation: A Survey*, IEEE Transactions of Pattern Analysis and Machine Intelligence, VOL 24, NO. 2, February 2002

[6] Jacky Baltes, *Localization for Mobile Robots using Lines*, Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV'02), December 2002