

# Camera Positioning And Orienting For Hyperredundant Robots Visual Servoing Applications

Razvan Tanasie\*, Mircea Ivanescu\*\*, Dorian Cojocaru\*\*\*

\* *Software Engineering Department, University of Craiova, Romania (e-mail: razvan.tanasie@software.ucv.ro)*

\*\* *Mechatronics Department, University of Craiova, Romania (e-mail: ivanescu@robotics.ucv.ro)*

\*\*\* *Mechatronics Department, University of Craiova, Romania (e-mail: cojocaru@robotics.ucv.ro)*

---

**Abstract:** Hyperredundant robots produce changes of configuration using a continuous backbone made of sections which bend. The lack of no discrete joints is a serious and difficult issue in the determination of the robot's shape. A solution for this problem is the vision based control of the robot, kinematics and dynamics. The term "camera calibration", used in this paper with the meaning of positioning and orienting the cameras in order to support the accuracy of the image features extraction, is an essential procedure for all such applications. The paper presents methods to calibrate for a pan/tilt/zoom camera shape by means of an engineered environment and a graphic simulation module. The experimental details show the effectiveness of the technique used with the hyperredundant manipulator.

*Keywords:* visual servoing, camera calibration, graphic simulator, hyper-redundant robot.

---

## 1. INTRODUCTION

The snake-arm robots and elephant's trunk robots are also described as continuum robots, although these descriptions are restrictive in their definitions and cannot be applied to all snake-arm robots. A continuum robot is a continuously curving manipulator, much like the arm of an octopus. An elephant's trunk robot is a good descriptor of a continuum robot. The elephant's trunk robot has been generally associated with an arm manipulation – an entire arm used to grasp and manipulate objects, the same way that an elephant would pick up a ball. As the best term for this class of robots has not been agreed upon, this is still an emerging issue. Snake-arm robots are often used in association with another device meant to introduce the snake-arm into the confined space.

A linear movement of the introduction axis coordinates is in this case the shape of the arm, thus, enabling the arm to follow a path into confined spaces. Continuous diameter along the length of the arm self-supporting either tendon-driven or pneumatically controlled in most cases represent other features which are usually (but not always) associated with snake-arm robots. A snakebot mimics the biomorphic motion of a snake in order to slither along the ground and is not to be confused with a snake-arm robot.

Considering the structure, mechanics, and movement of octopus arms, a highly dexterous, high-DOF soft robotic manipulator, named "OCTARM", was designed and constructed (Walker et al., 2006). The arm was modeled as a segmental 2D array of point-masses and springs. OCTARM moves using McKibben air muscles as actuators. These air muscles have a large strength-to-weight ratio, allowing OCTARM to remain lightweight but powerful. The actuators are similar in many regards to the muscles found in muscular hydrostats such as octopus arms. Actuation force depends

both on the air pressure used for inflation as well as the cross-sectional area of the actuator. In total, OCTARM uses eighteen air muscles that extend when pressurized. The extension actuators all have large strains (> 40%). These large strains allow for great dexterity and arm manipulation range.

Behavior similar to biological trunks, tentacles, or snakes may be exhibited by continuum or hyper-redundant robot manipulators. Hence these manipulators are extremely dexterous, compliant, and are capable of dynamic adaptive manipulation in unstructured environments; continuum robot manipulators do not have rigid joints unlike traditional rigid-link robot manipulators.

However, the development of high-performance control algorithms for these manipulators is quite a challenge, due to their unique design and the high degree of uncertainty in their dynamic models. A controller for continuum robots, using a neural network feed-forward component to compensate for dynamic uncertainties was developed (Walker et al., 2005) and experimental results using the OCTARM, which is a soft extensible continuum manipulator, are provided to illustrate that the addition of the neural network feed-forward component to the controller provides improved performance (Hannan et al., 2005).

The movement of the continuum robot mechanisms is generated by bending continuously along their length to produce a sequence of smooth curves. This contrasts with discrete robot devices, which generate movement at independent joints separated by supporting links. The two types of mechanism have very different internal structure, and hence produce very different motions. In the AMADEUS project a dexterous underwater robot hand was developed for grasping and manipulation tasks using continuum fingers. In the FLAPS project, a propulsion system which replicate the

mechanics of fish swimming, as well as an inspection system for confined locations utilizing multiple actuator sections was developed (Davies *et al.*, 1998).

This paper will present techniques for orienting and positioning the camera using a graphic environment or an engineered workspace.

## 2. A TENTACLE MANIPULATOR: PROJECT DESCRIPTION

The research group from the Faculty of Automation, Computers and Electronics, University of Craiova, started working in research field of hyperredundant robots over 20 years ago. The experiments were performed on a family of TEROB robots which use cables and DC motors. The kinematics and dynamics models, as well as the different control methods developed by the research group were tested on these robots. An experiment of visualservoing for this class of robots was presented at RAAD'07.

### 2.1. General presentation of the project

Starting with 2008, the research group (with the contribution of Professor Nicolae Dumitru from the Faculty of Mechanics, University of Craiova) designed a new experimental platform for hyperredundant robots. This new robot is actuated by stepper motors. The rotation of these motors rotates the cables which by correlated screwing and unscrewing of their ends determine their shortening or prolonging, and by consequence, the tentacle curvature. In the present stage the manipulator is formed of three segments. All segments are cylindrical. The backbone of the tentacle is an elastic cable made out of steel, which sustains the entire structure and allows the bending. Depending on which cable shortens or prolongs, the tentacle bends in different planes, each one making different angles (rotations) respective to the initial coordinate frame attached to the manipulator segment – i.e. allowing the movement in 3D.

Due to the mechanical design it can be assumed that the individual cable torsion, respectively entire manipulator torsion can be neglected. Even if these phenomena would appear, the structure control is not based on the stepper motors angles, but on the information given by the robotic vision system which is able to offer the real spatial positions and orientations of the tentacle segments.

Due to the difficulties of using distributed sensors along the manipulator structures, it was decided to use a vision system. The vision system is composed by two pan / tilt / zoom cameras and a frame grabber. The term “camera calibration” in the context of this paper refers to positioning and orienting the two cameras at imposed values. This calibration is performed only at the beginning, after that the cameras remain still. First, a zoom that maximizes the image resolution of the working space used by the manipulator is performed. Second, positioning of the two cameras brings the manipulator in the middle of the two images. Third, a pan / tilt orientation is performed (as described later in the paper). At this step the manipulator is moved in a test position that allows free of (or minimum) errors calibration. The test images are compared to the images generated by the graphic

simulator (ideal images) which represent references for the calibration operation.

A second method is also presented. This method is based only on the use of the visual system and some especially designed elements for this application. One of these elements is a grey cylindrical standard object that has red vertical equidistant (90 degrees) axes. This standard object is positioned instead of the robot in the center of the workspace. The position of the vertical axes allows the algorithm to position and orient the vision system in the required positions for the control procedure.

The general control method is an image based visual servoing one instead of position based. By consequence, camera calibration based on intrinsic parameters (classic sense, not the one used in this paper) is not necessary.

### 2.2. Details of the implementation

A tentacle manipulator is a hyper-redundant or hyper-degree-of-freedom manipulator and there has been a rapidly expanding interest in their study and construction lately. The control of these systems is very complex and a great number of researchers have tried to offer solutions for this difficult problem (Ivanescu, 1984).

The inverse kinematics problem is reduced to determining the time varying backbone curve behavior. New methods for determining “optimal” hyper-redundant manipulator configurations based on a continuous formulation of kinematics are developed. The difficulty of the dynamic control is determined by integral-partial-differential models with high non-linearity that characterize the dynamic of these systems (Ivanescu, 2002).

In a paper presented at RAAD 2007, the problem of a class of tentacle arms of variable length was discussed (Ivanescu *et al.*, 2007). First, the dynamic model of the system was inferred. The method of artificial potential was developed for these infinite dimensional systems. In order to avoid the difficulties associated with the dynamic model, the control law was based only on the gravitational potential and a new artificial potential. Servoing was based on binocular vision, a continuous measure of the arm parameters derived from the real-time computation of the binocular optical flow over the two images, and is compared with the desired position of the arm. The control error function was built in 3D Cartesian space by the visual information obtained by two cameras in two image planes. The two 2D errors obtained in the two image planes were determined by the two differences between the actual and desired continuous angle values that define the projections of the arm shape. The plane errors can be considered as the errors of the arm shape. These errors were used to calculate the spatial error and a control law was synthesized.

A tentacle arm prototype was designed and the practical realization is now running. It is a cable based mechanism having three segments (see an “old” experimental segment in Fig. 1a, CAD images during the simulation in Fig. 1b and new model implementation in Fig. 1c) in the first implementation.



Fig. 1a. "Old" model

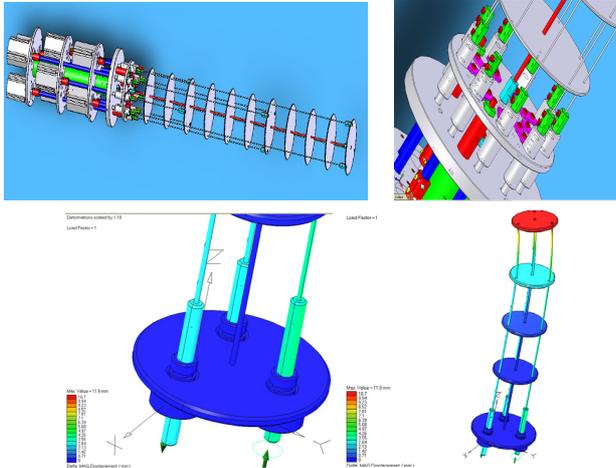


Fig. 1b. CAD images of the tentacle design



Fig. 1c. New model structure

The tentacle arm is designed to be actuated by 3-phase stepper motors. The interfaces are pulse direction based without rotation monitoring. Set-point position of the stepper motor is preset as a pulse signal by a controller via signal interface. A pulse corresponds to one step of the motor (0.5 pps-2.4M pps, Max. Acceleration Rate: 737M pps<sup>2</sup>, Speed resolution: 16-bit). An electronic relay contact reports operating readiness. The nominal torque MN is 2 Nm. Steps per revolution are selectable from 200 to 10000. The step angle  $\alpha$  is selectable from 1.8° to 0.036°. Three stepper motors are used for each segment of the tentacle.

4-Axis Stepper Motion Controller boards are used. It is a pulse train motion controller which provides T/S curve control, on-the-fly speed change, non-symmetric acceleration and deceleration profile control, and simultaneous start/stop functions. This controller also offers card index settings for multiple cards in one IPC system.



Fig. 2. Visual Control benchmark

The boards offer powerful speed change functions that can be executed while the axis is moving. After motion begins, the target speed can be changed as needed according to the application. By using either a software function or external input signal, the controller can perform simultaneously starts and stops on multiple axes in a one-card configuration, or multiple axes in a multiple-card application (our case). The simultaneously stop function can be selectively active when an axis (or axes) stops abnormally.

In order to implement the visual-servoing system, a benchmark was organized (see Fig. 2 – the old experimental model) based on two color camera with 0.05lux low light sensitivity and the DT3162 frame grabber from Data Translation.

The cameras have motorized Pant/Tilt/Zoom (10x optical zoom) and are mounted in perpendicular planes offering the input for the frame grabber. The Pant/Tilt/Zoom precision is sufficient for this step of the application development (Pan: range +135, 10 50/sec; Tilt: range +90 45, 7 25 /sec; Zoom: 1x~10x optical zoom). Two white screens are placed in front of the cameras in order to increase the image's contrast. The tentacle arm is placed between each camera and its screen.

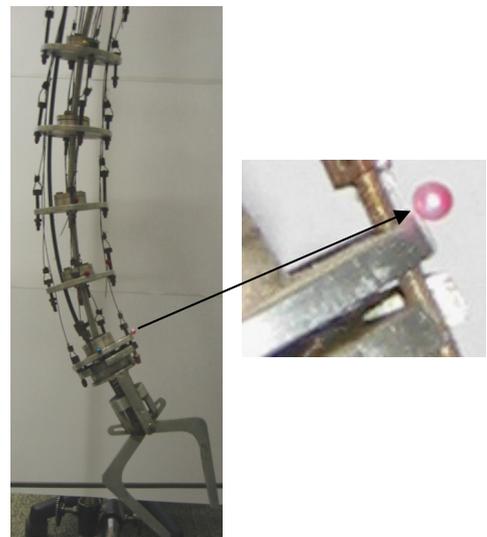


Fig. 3. Set of markers

The image processing tasks are performed using Global LAB Image2 from Data Translation. The robot control algorithms are implemented in a C++ program running on a Pentium IV PC. In order to facilitate the image feature extraction, a set of markers are placed on joints along the backbone structure (see Fig. 3).

### 3. CAMERAS CALIBRATION

Two video cameras provide two images of the whole robot workspace. The two images planes are parallel with XOY and ZOY planes from robot coordinate frame, respectively (see Fig. 4).

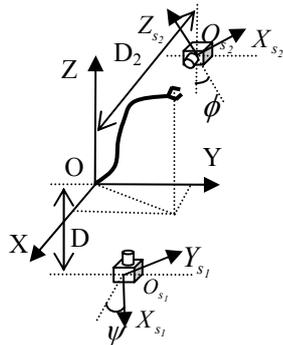


Fig. 4. Cameras frames

The cameras provide the images of the scene that stored in the frame grabber's video memory. Related to the image planes are defined two dimensional coordinate frames, called screen coordinate frames or image coordinate systems. Denote  $X_{s1}, Y_{s1}$  and  $Z_{s2}, Y_{s2}$  respectively, the axes of the two screen coordinate frames provided by the two cameras. The spatial centers for each camera are located at the distances  $D1$  and  $D2$ , with respect to the XOY and ZOY planes, respectively. The orientation of the cameras around the optical axes with respect to the robot coordinate frame, are noted by  $\psi$  and  $\phi$ , respectively (Grosso et al., 1996; Hutchinson et al., 1996; Kelly, 1996).

The control system is an image – based visual servo control (Pressigout et al., 2004) where the error control signal is defined directly in terms of image feature parameters. The desired position of the arm in the robot space is defined by the curve  $C_d$ , or, in the two image coordinate frames  $Z_{s1}O_{s1}Y_{s1}$  and  $Z_{s2}O_{s2}Y_{s2}$ , by the projection of the curve  $C$  (Fig. 5).

The control problem of this system is a direct visual servo-control, but the classical concept of the position control, in which the error between the robot end-effector and target is minimized, is not used. In this application the control of the shape of the curve in each point of the mechanical structure is used (Fig. 6).

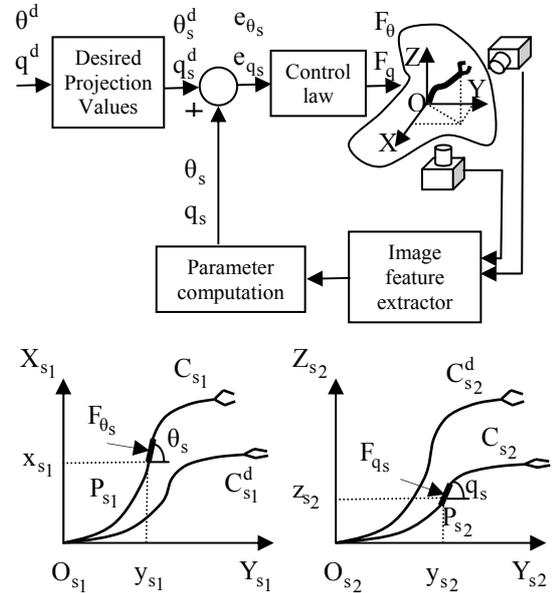


Fig. 5. The global control system

The method is based on the particular structure of the system defined as a “backbone with two continuous angles  $\theta(s)$  and  $q(s)$ ” (the tentacle bends in different planes, each one making different angles (rotations) respective to the initial coordinate frame attached to the manipulator segment,  $\theta(s)$  and  $q(s)$  representing the bending angle and the plane rotation angle). The control of the system is based on the control of the two angles  $\theta(s)$  and  $q(s)$ .

These angles are measured directly or indirectly. The angle  $\theta(s)$  is measured directly by the projection on the image plane  $Z_{s1}O_{s1}Y_{s1}$  and  $q(s)$  is computed from the projection on the image plane  $Z_{s2}O_{s2}Y_{s2}$ .

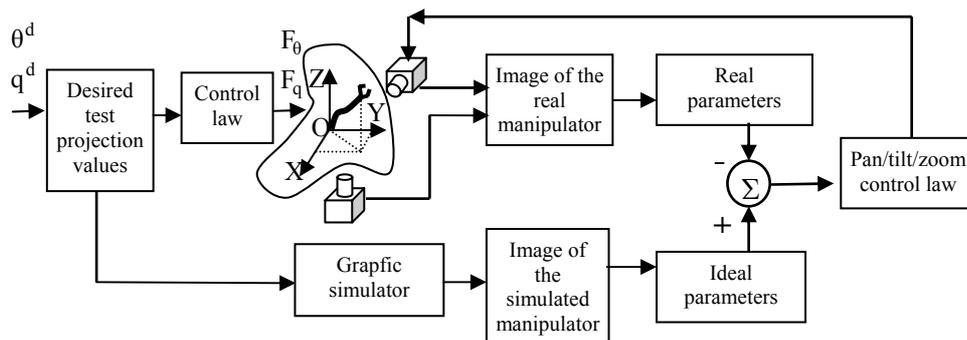


Fig. 6. Camera calibration system

A very important task in developing this application is to control the camera position and orientation. From this point of view, the calibration operation assures that the two cameras' axes are orthogonal.

If the selected algorithm is the one using the graphic environment, in the beginning, the tentacle manipulator receives the needed commands in order to stand in a test pose (imposed position and orientation). Then the same commands are sent to the Graphic Simulator. Two different sets of images are obtained: real images acquired by the real cameras and simulated images offered by the Graphic Simulator. From these two sets of images, two sets of parameters are computed: real parameters are computed from real images and, respectively, ideal parameters are computed from synthetic images. Comparing the two sets of parameters and knowing the image/parameters behavior for the camera orientation, the cameras are orientated (pan/tilt/zoom) in order to minimize the error.

If the predefined object method is applied for camera calibration, then the objects are placed in the desired position and orientation and only one set of images (obtained from the real visual system is used).

### 3.1. Camera Calibration Algorithms

Camera calibration is the essential procedure for all such applications: positioning and orienting the cameras in order to support the accuracy of the image features extraction. Calibration for a pan/tilt/zoom camera shape is achieved by means of an engineered environment and a graphic simulation module.

The term "camera calibration" in the context of this paper refers to positioning and orienting the two cameras at imposed values. This calibration is performed only at the beginning, after that the cameras remain still.

The general control method is an image based visual servoing one instead of position based. Camera calibration based on intrinsic parameters (classic sense, not the one used in this paper) is not necessary. Calibration operation assures that the two cameras' axes are orthogonal.

Taking into account the presented structure of the tentacle - vision system, in order to apply the tested visual servoing algorithm, the two cameras must be positioned and oriented as:

- both focus on the robot.
- their axes are orthogonal.
- both have the same zoom factor.

Two different algorithms were implemented:

- one uses a cylindrical standard object.
- other uses the graphical simulator.

For the first algorithm, special starting conditions were imposed in order to support the image processing tasks: white background, dark grey cylinder, red vertical equidistant (90

degrees) axes (Fig. 7.), friendly initial camera's positions and orientations, zoom x1.

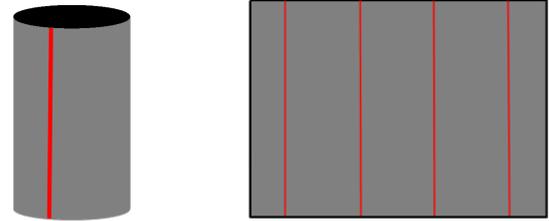


Fig. 7. The cylindrical model

Three successive and dependent calibrations are performed:

- Horizontal (pan): position and orientation are obtained in two successive, but dependent steps.
- Vertical (tilt): position and orientation are obtained in two successive, but dependent steps.
- Zoom: tuning the two cameras as both look to the cylinder from virtual equal distances.

Horizontal orienting:

1. Image acquisition & segmentation using histogram threshold,
2. Counting for each line in the image the number of black and respectively white pixels,
3. For the line with the greater number of black pixels, compute  $p_l^{horiz}$  and  $p_r^{horiz}$  that are the number of white pixels from the left and respectively from the right,
4. Compute the offset  $\Delta^{horiz} = \left| p_r^{horiz} - p_l^{horiz} \right|$ ,
5. IF  $\Delta^{horiz} \leq \varepsilon_{accept}^{horiz}$  THEN stop rotating the camera,
6. ELSE rotate the camera to the left or to the right using the horizontal offset,
7. Back to start, 1.

Horizontal positioning:

1. Image acquisition & segmentation using histogram threshold,
2. Scanning the image horizontally until the axes are detected; if two axes are detected then the camera is rotated so the most lateral axes goes outside the image,
3. Counting for each line in the image the number of black and respectively white pixels,
4. For each line, the number of white pixels on the left of the axes is  $p_l^{horiz}$  and the number of white pixels on the right of the axes is  $p_r^{horiz}$ ,
5. The horizontal offset is computed  $\Delta^{horiz-pos} = \left| p_r^{horiz-pos} - p_l^{horiz-pos} \right|$ ,
6. IF  $\Delta^{horiz-pos} \leq \varepsilon_{accept}^{horiz-pos}$  THEN stop translating the camera,

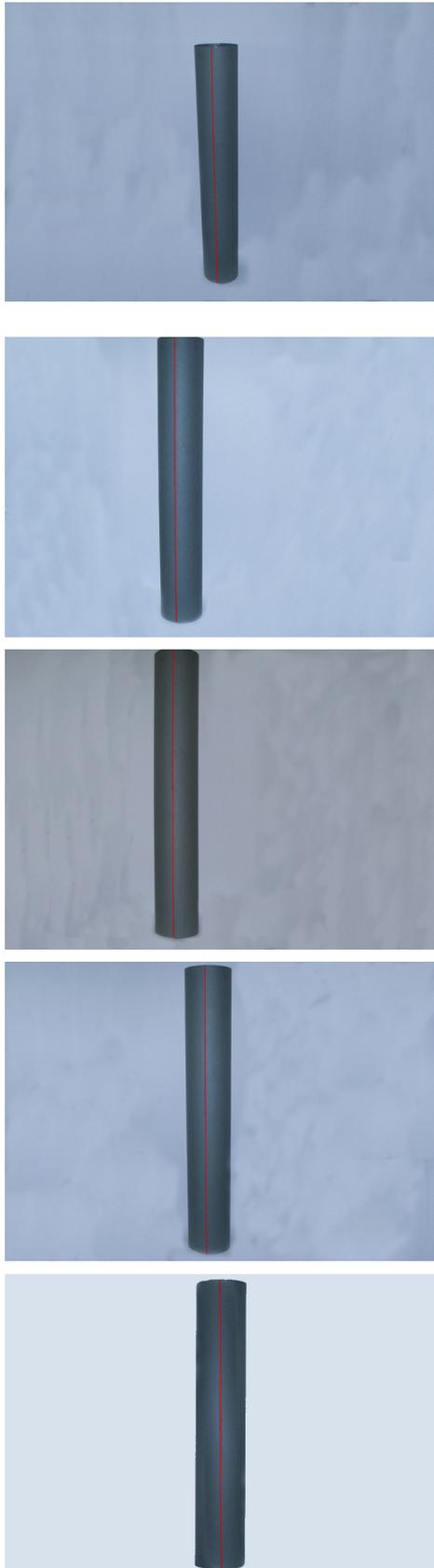


Fig. 8. Calibration results

7. ELSE translate the camera to the left or to the right using  $\text{sgn}(p_r^{\text{horiz-pos}} - p_l^{\text{horiz-pos}})$ ,
8. Back to start, 1.

The full horizontal calibration is performed as follows:

1. Horizontal orienting,
2. Horizontal positioning,
3. The offset for horizontal orienting is computed  $\Delta^{\text{horiz}} = |p_r^{\text{horiz}} - p_l^{\text{horiz}}|$ ,
4. IF  $\Delta^{\text{horiz}} \leq \epsilon_{\text{accept}}^{\text{horiz}}$  THEN the algorithm stops,
5. ELSE go to step 1.

Both offsets must be under the accepted thresholds. Else, the positioning destroyed the orientation and the procedure must be repeated.

A similar algorithm is developed for the vertical orientation and positioning. The results of this algorithm are presented for a particular example in Fig. 8.

The second algorithm works together with the graphic simulator. Consider the case when the robot tentacle is bended as a perfect circle arc. It was proven that the two camera axes are orthogonal if they „look” at the tentacle and they „see” circle arcs for positions of the robot that are obtained one from another by a 90 degree rotation around robot vertical axis. The previous condition is fulfilled if each camera looks at the center of the circle determining the arc and the view line is orthogonal on the plane's circle (Fig. 9).

Three calibration steps must be performed:

- Horizontal calibration - positioning and orienting the camera horizontally (pan),
- Vertical calibration - positioning and orienting the camera vertically (tilt),
- Zoom calibration - tuning the two cameras as both look at the robot from virtual equal distances.

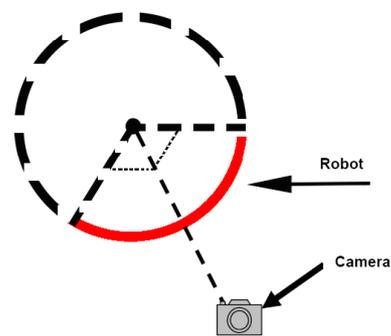


Fig. 9. Camera looks to the center of the circle

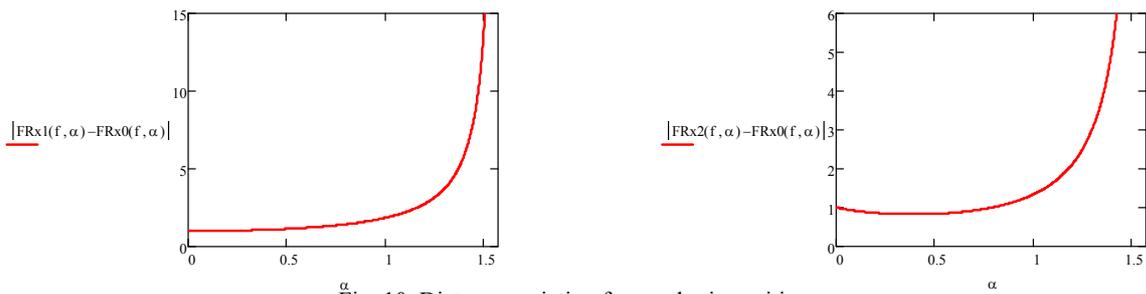


Fig. 10. Distance variation for quadratic positions

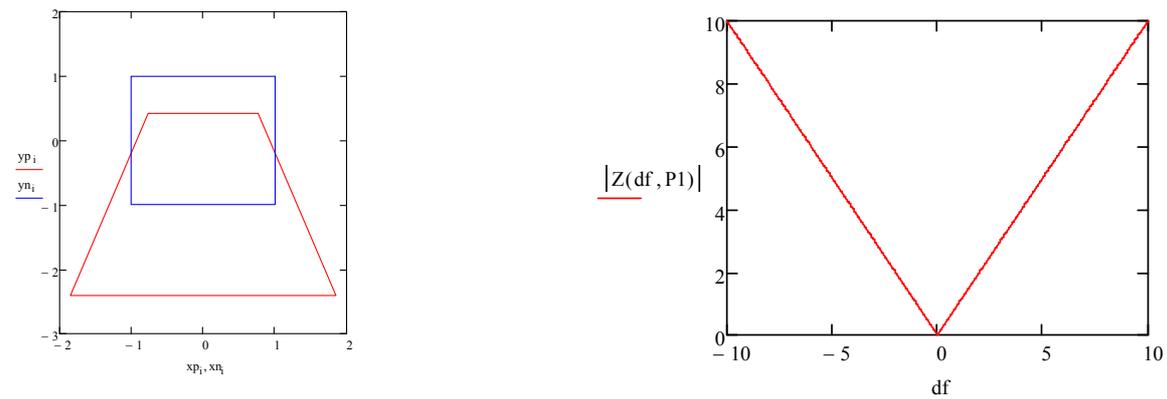


Fig. 11. Rectangle transformation and distance variation under zoom influence

The calibration algorithm has the following steps:

1. The graphic simulator computes the necessary bending and rotating angles in order to transform the tentacle into an arc in the XOY plane.
2. The robot's control commands are generated in order to achieve the imposed positions and orientations.
3. The first camera acquires an image.
4. The image is segmented.
5. If the detected curve (the median axis of the tentacle) verifies the circle's equation then the algorithm continues from step 7.
6. Else the first camera is moved and algorithm returns to the step 3.
7. The same operation is performed for the ZOY plane and the second camera.
8. The optical correlation for the equalization of "virtual" zoom factors is performed.

How to “move” the camera in step 6? The image behavior in accordance with camera’s movements was studied. The effect of pan and tilt rotations on two points placed in a quadratic position on a circle was geometrically described. Matrix for coordinate transformations corresponding to rotations with pan and tilt angles, respectively for perspective transformation were used.

The variation of the distance between the two points, placed in a quadratic position on the circle, and the centre of the circle, depending of the tilt angle X, are plotted in Fig. 10.

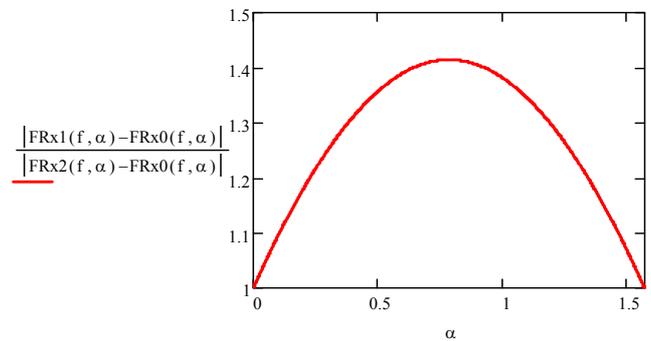


Fig. 12. Ratio distances variation

The following plot in Fig. 11. shows how a rectangle (inscribed in the circle and having the edges parallel with the axes OX and OY) is transformed when a tilt rotation is performed. Theoretically, by zooming, the distance between the two points varies in a linear way, as it is shown upper right. The variation of the ratio of the two distances is plotted in Fig. 12.

The image’s segmentation is basically a threshold procedure applied to the image’s histogram. All the procedures included in the calibration algorithms were mathematically proven. If the calibration algorithm was successfully applied then the system is ready to perform the visualservoing tasks.

#### 4. CONCLUSIONS

This paper deals with a project for building and controlling a hyperredundant robot which is currently developing. Different control methods and algorithms were proposed by the team members. A new tentacle manipulator using cables and stepper motors was designed and is under construction and a visual control using two video cameras is benchmarked. A graphic simulator was created in order to support the system implementation. In this paper, two camera calibration methods were presented. The term “camera calibration” in the context of this paper refers to positioning and orienting the two cameras at imposed values.

In specialized literature, the research of I.D. Walker from Clemson University in connection to the cable operated hyperredundant robots is widely acknowledged.

The computer vision system part of the robotics lab at Clemson University is based on a high speed camera, a frame grabber and a PC. The envisaged proposal in this paper is based on the idea of using a computer vision system based on two cameras orthogonally placed.

This method provides the advantage to determine the 3D form of the arm only by determining 2 angles strictly measured in the image without any global parameter (relative coordinates to a coordinates' system attached to the working space).

By its nature, this method involves a particular calibration, different from the regular one in a computer vision application. The content of this paper focuses exactly on this aspect of the calibration that ensures the particular orientation and position imposed by the theoretical foundation of this method.

As a conclusion of this comparative assessment, the envisaged method simplifies the determination of the 3D form using two angles for each articulation and introduces a supplementary calibration step.

#### REFERENCES

- Davies, J.B.C., Lane, D.M., a.o. (1998), Subsea applications of continuum robots, *Underwater Technology, Proceedings of the 1998 International Symp. on*, Issue 15-17 Apr 1998, Page(s):363 – 369.
- Grosso, E., Metta, G., a.o. (1996), Robust Visual Servoing in 3D Reaching Tasks, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 15, pp. 732-742.
- Hannan, M.W. and I. D. Walker. (2005), Real-time shape estimation for continuum robots using vision, *Robotica*, volume 23, pp. 645–651, Cambridge University Press.
- Hutchinson, S., Hager, G. D. and Corke, P. F. (1996), A Tutorial on Visual Servo Control, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 15, pp. 651-670.
- Ivanescu, M. (1984), Dynamic control for a tentacle manipulator, *Proceedings International Conference on Robotics and Factory of the Future*, pp. 315-327.
- Ivanescu, M. (2002), Position Dynamic Control for a Tentacle Manipulator, *Proceedings IEEE International Conference on Robotics and Automation*, 1531-1539.
- Ivanescu, M., Cojocaru, D., a.o. (2007), Visual Servoing By Artificial Potential Methods Of A Hyperredundant Robot, *Proceedings of 16th International Workshop on Robotics in Alpe-Adria-Danube Region - RAAD 2007*.
- Kelly, R. (1996), Robust Asymptotically State Visual Servoing, *Proceedings IEEE International Conference on Robotics and Automation*, vol. 22, no. 15, pp. 759-765.
- Pressigout, M., E. Marchand, (2004), Model-free Augmented Reality by Virtual Visual Servoing, *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 1051-4651/04.
- Walker, I.D., Darren M. Dawson (2005), *Continuum Robot Arms Inspired by Cephalopods*, DARPA Contr. N66001-C-8043, <http://www.ces.clemson.edu/~ianw/spie05.pdf>
- Walker, I.D. and Carlos Carreras. (2006), Extension versus Bending for Continuum Robots, *International Journal of Advanced Robotic Systems*, Vol. 3, No.2 (2006), ISSN 1729-8806, pp. 171-178.