

## On low cost model-based monitoring of industrial robotic arms using standard machine vision

Aris Karagiannidis<sup>a</sup> and George C. Vosniakos<sup>\*</sup>

*School of Mechanical Engineering, National Technical University of Athens,  
Heron Polytechniou 9, 157 80 Athens, Greece*

*(Received September 3, 2013, Revised October 29, 2013, Accepted November 9, 2013)*

**Abstract.** This paper contributes towards the development of a computer vision system for telemonitoring of industrial articulated robotic arms. The system aims to provide precision real time measurements of the joint angles by employing low cost cameras and visual markers on the body of the robot. To achieve this, a mathematical model that connects image features and joint angles was developed covering rotation of a single joint whose axis is parallel to the visual projection plane. The feature that is examined during image processing is the varying area of given circular target placed on the body of the robot, as registered by the camera during rotation of the arm. In order to distinguish between rotation directions four targets were used placed every 90° and observed by two cameras at suitable angular distances. The results were deemed acceptable considering camera cost and lighting conditions of the workspace. A computational error analysis explored how deviations from the ideal camera positions affect the measurements and led to appropriate correction. The method is deemed to be extensible to multiple joint motion of a known kinematic chain.

**Keywords:** industrial robot; pose monitoring; industrial computer vision; area features

---

### 1. Introduction

Modern robotic arms have long and invariably adopted encoders that are embedded in their joints in order to measure the respective joint coordinates which determine the pose of the respective kinematic chain and are used in forward and inverse kinematics algorithms in the robot controller. This information is not easily available outside the controller of the robot, or outside its programming environment. This paper contributes towards the creation of an optical monitoring system that measures the joint angles using computer vision techniques. Currently the developed system focuses on a single joint of the kinematic chain with vertical axis of rotation, namely the first joint but without loss of generality.

The original motivation for this work was the need to synchronize virtual models of industrial robots typically expressed in VRML with the real robots by using information that is acquired by low-cost cameras. This information should be processed locally to extract minimal data that is

---

<sup>\*</sup>Corresponding author, Associate Professor, E-mail: [vosniak@central.ntua.gr](mailto:vosniak@central.ntua.gr)

<sup>a</sup> Graduate student, E-mail: [arkaragian@gmail.com](mailto:arkaragian@gmail.com)

subsequently transferred over the internet to remote computers where the virtual model is actually run. Thus, a fast low-cost telemonitoring system is targeted mainly in the realm of surveillance of unmanned automated manufacturing systems.

The alternative to using image-based monitoring would be to obtain the same data from the encoders or the controllers of the machines involved. However, apart from circumventing technical difficulties in communicating this data to external computers, especially for older machines, an optical system acts as an independent sensor bypassing possible encoder malfunctions. In addition, a vision system can be thought of as a uniform monitoring sensor because it deals homogeneously with all needs involving motion mainly regarding machines, but also, and more intensively so in recent years, regarding humans, too, thereby addressing complex human-machine interaction systems. In the latter domain, task planning and training, especially within novel augmented or mixed reality environments, can also benefit from such an approach.

Technically the overall problem tackled is the detection and motion tracking of robotic rotary joints. Four immediate subproblems can be readily recognized, namely determining the number of cameras to be used, their positions, the features that are to be detected and the algorithms for feature detection and information extraction. The first two problems are tackled informally in this work by common sense solutions since the domain of application (one joint only) is trivial; otherwise they constitute a large research chapter as such (Chen *et al.* 2011). The technical issues entailed within the other two sub-problems can be briefly outlined as follows.

The features to be tracked should be chosen carefully so as to be unsusceptible to noise and computationally suitable for real time detection and tracking. This is desired, in this work at least, to be performed by standard machine vision algorithms, widely reported in literature (Golnabi and Asadpour 2007); therefore suitable properties of these features should be identified and the relevant algorithms should be selected and/or enhanced, extended or combined appropriately.

Common features that are used for target identification and tracking are color and shape. Shape detection often includes the sub-problems of edge and corner detection. The most common algorithm for shape detection is the Hough transform which can be extended for the detection of arbitrary shapes (Ballard 1981). Of course, more complex shapes impose higher demands in computational power. Additionally, color detection schemes combined with machine learning techniques have been developed with satisfactory results (Mirghasemi and Banihashem 2009). Other methods involve the use of both shape and color (Wu and Xiao 2010) for target recognition and have been proven to meet real time demands.

It is to be noted that the current work does not target visual servoing or generally any control applications at all. However, it is recognized that within visual servoing approaches explicit robot pose measurement or estimation is often addressed.

Wang *et al.* (2002) have created a 3D shape measurement system. The system utilizes targets that are placed on the measured object. Then the cloud of points from the partial views that is acquired through the use of computer vision techniques is transformed to the world coordinate system.

Camarillo *et al.* (2008) present a vision based system for quantifying the 3-D shape of a flexible manipulator in real-time. The sensor system is validated for accuracy with known point measurements and for precision by estimating a known 3-D shape. Two applications of the validated system relating to the open-loop control of a tendon driven continuum manipulator are presented. In the first application, 3-D performance is quantified, while in the second application, the system is used for model parameter estimation.

Li *et al.* (2010) present a method of rotation angle measurement in machine vision. This method is based on a calibration pattern with a spot array. As the pattern rotates, the coordinates of the spots on the calibration pattern are acquired with a camera. Using the initial position of the array as a calibration point and with the equations of coordinate rotation measurement, the rotation angle of the spot array is detected.

Di Leo and Paolillo (2011) investigate the propagation of uncertainty through calibration algorithms. Camera calibration is an important task for vision-based measurement systems in the general case. The uncertainty of these parameters is required in order to evaluate the induced uncertainty of vision-based final measurements. Numerical results are presented that highlight the usefulness of the method in the set-up of a vision-based measurement system.

Jungel *et al.* (2008) demonstrate how to use reference objects to improve vision-based distance measurements to objects of unknown size. Several methods for different kinds of reference objects are introduced. These are objects of known size, objects extending over the horizon, and objects with known shape on the ground. They also describe how to determine the rotation of the robot's camera relative to the ground and provide an error-estimation for all methods.

Heisele and Ritter (1995) presented ways to detect obstacles based on color blob flow. The authors were able to extract moving objects in traffic scenes. The methods developed, do not require previous knowledge about the shape or location of those moving objects.

Ming and Ma (2007) propose a novel and efficient framework for the extension of multi-scale blob detection for the color domain to prevent information loss due to gray scale transformation and to allow exploitation of photometric information. The framework also works well in gray scale space, and it is mainly composed of: a weighted multi-scale blob detector using a hybrid Laplacian and determinant of the Hessian operator, a blob filter, a color-based Forstner operator for roundness calculation and a hue-based color histogram.

Su *et al.* (1995) investigate introduction of neural networks for precise measurements in computer vision systems. They describe procedures for error correction from optical and other sources.

Tyrrell *et al.* (2004) present ways for efficient migration of complex off-line vision systems to real time implementations on generic computer hardware. This is achieved by using special modules in the kernel of the operating systems. The results were satisfactory when predictable vision algorithms were used.

Fung and Mann (2004) present methods for using graphics cards as generic hardware for computer vision. As a proof of concept they adjust and implement common vision algorithms that run on the multiprocessor platform. This is akin with the work of Fung *et al.* (2002) that differs in the application domain which is the creation of an augmented reality environment based on the images acquired from wearable cameras.

In this work, a simple system is sought in order to estimate angle of rotation of robotic joints visually. The system should be able to function in real time, to use low-cost equipment and to perform at an accuracy that was initially set to 3 degrees, but would be open to much improvement at a later stage to reach useful levels of a tenth of a degree.

The principle of the proposed system is the apparent deformation of targets of known shape that are marked on the body of the robotic arm as observed from a fixed camera viewpoint during the motion of the arm.

The pilot phase that is reported on in this paper consists of investigating rotation about just one joint. Given the kinematic model of the robot it is theoretically simple to extend the one-joint case

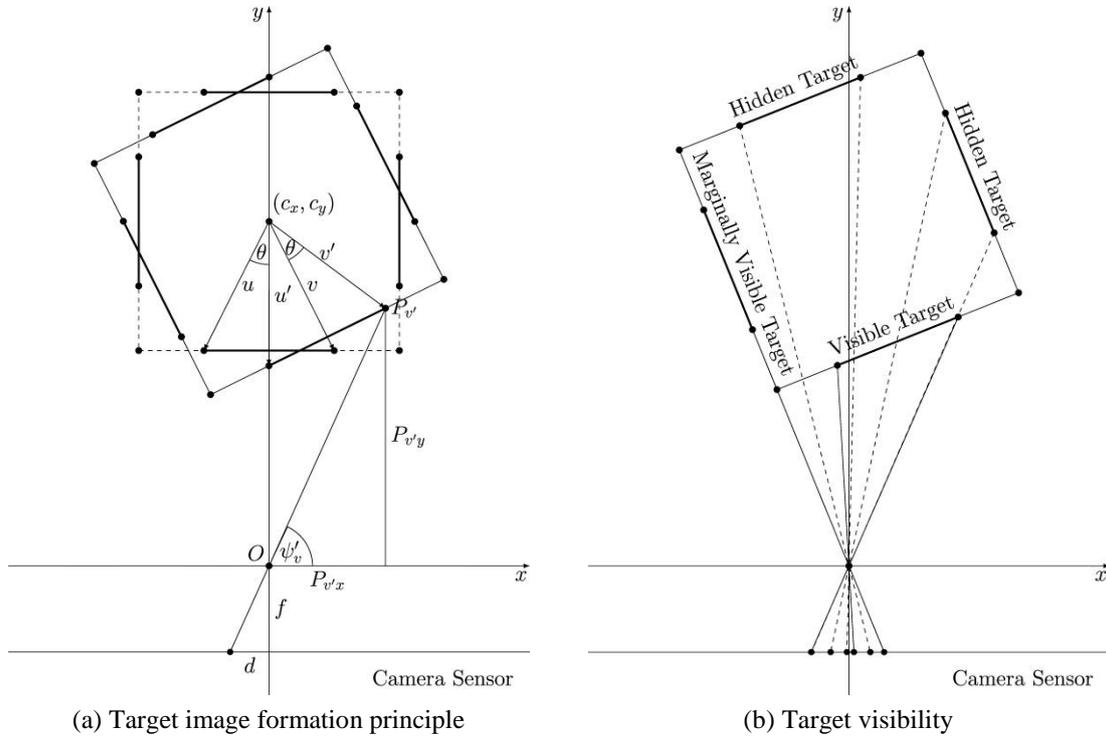


Fig. 1 Rotary joint cross-section normal to axis of rotation

to motion about multiple joints, by successive application of the relevant transformation matrices from the first to the last joint. The initial assumption is the absence of any coincident joints, but this may be relaxed later.

Note that target deformation is measured in terms of its area which is immediately translated to a number of pixels. A simple mathematical model is derived that yields the deformation of the target seen by the camera as a function of the angle of rotation of the robotic joint. A reverse solution of this model yields the angle of rotation.

The mathematical model and its reverse solution are presented in Section 2. Implementation issues of the system in terms of standard machine vision functions are reported on in Section 3. Typical results are reported on in Section 4. Section 5 deals with the error involved in the proposed approach estimating its components relating mainly to camera placement. Conclusions and planned extensions of this work are outlined in Section 6.

## 2. Model

The solution of the problem is based on the mapping of rotation of the joint to the visible area of the targets, as those are seen from the camera. The target shape that was selected was a circle due to its geometric simplicity. Rotation of a circle lying on a vertical plane about its vertical axis being parallel to the  $z$  Cartesian axis produces an ellipse. Hence, the problem is reduced to finding the area of the image of the projection of the target on the camera sensor.

The horizontal cross section ( $x$ - $y$  plane) of the joint rotating about the  $z$  axis is presented in Fig. 1(a). The targets are placed on the shell of the joint and represented as lines in the horizontal cross-section  $z=const$ .

Two vectors  $u$  and  $v$  are defined that start from the center of rotation of the joint and end on the circumference of the target circle on the plane  $z=const$ . During rotation of the joint those vectors can be expressed as

$$\begin{aligned} u' &= Rot_z(\theta) \times u \\ v' &= Rot_z(\theta) \times v \end{aligned} \quad (1)$$

where  $Rot_z(\theta)$  is the rotation matrix around the  $z$ -axis. Since the center of rotation is known, the endpoints of the vectors can be expressed as

$$\begin{aligned} P_{ux} &= x_u \cos(\theta) - y_u \sin(\theta) + c_x & P_{uy} &= x_u \sin(\theta) + y_u \cos(\theta) + c_y \\ P_{vx} &= x_v \cos(\theta) - y_v \sin(\theta) + c_x & P_{vy} &= x_v \sin(\theta) + y_v \cos(\theta) + c_y \end{aligned} \quad (2)$$

where  $c$  is the center of rotation and  $x_i, y_i$  the components of the vectors of Eq. (1). Considering the camera lens center  $O$  as the origin of the coordinate system it is possible to find the angles that are defined by the line segment  $PO$  and the  $x$ -axis, where  $P$  denotes the respective point  $P_{u'}$  and  $P_{v'}$  of the  $u'$  and  $v'$  vectors. Those are

$$\psi_i = \arctan\left(\frac{P_{iy}}{P_{ix}}\right) \quad (3)$$

With those angles known it is possible to find the distance of the “mirrored” point on the sensor from the  $y$ -axis as

$$d_i = \tan\left(\frac{\pi}{2} - \psi_i\right) f \quad (4)$$

and the horizontal radius of the formed ellipse is

$$r_h = \frac{\sqrt{(d_1 - d_2)^2}}{2} \quad (5)$$

The vertical radius of the ellipse can be directly calculated by using similar triangles as those are constructed in the pinhole camera model. Let  $S$  be the vertical distance of the radius  $R$  of the target from the camera lens. Then from similar triangles  $r_v$ , the projection of  $R$  on the camera sensor can be deduced as, see Fig. 2

$$r_v = f \frac{R}{S} \quad (6)$$

The distance  $S$  depends on the angle of rotation since, as the joint rotates, its center point goes further and further away from the camera. If  $L$  is the distance of the camera from the center of rotation, then  $S$  would be equal to, see Fig. 2(a)

$$S = L - r \cos(\theta) \quad (7)$$

with  $r$  being the radius of rotation of the target circle which is solely dependent on the geometry of

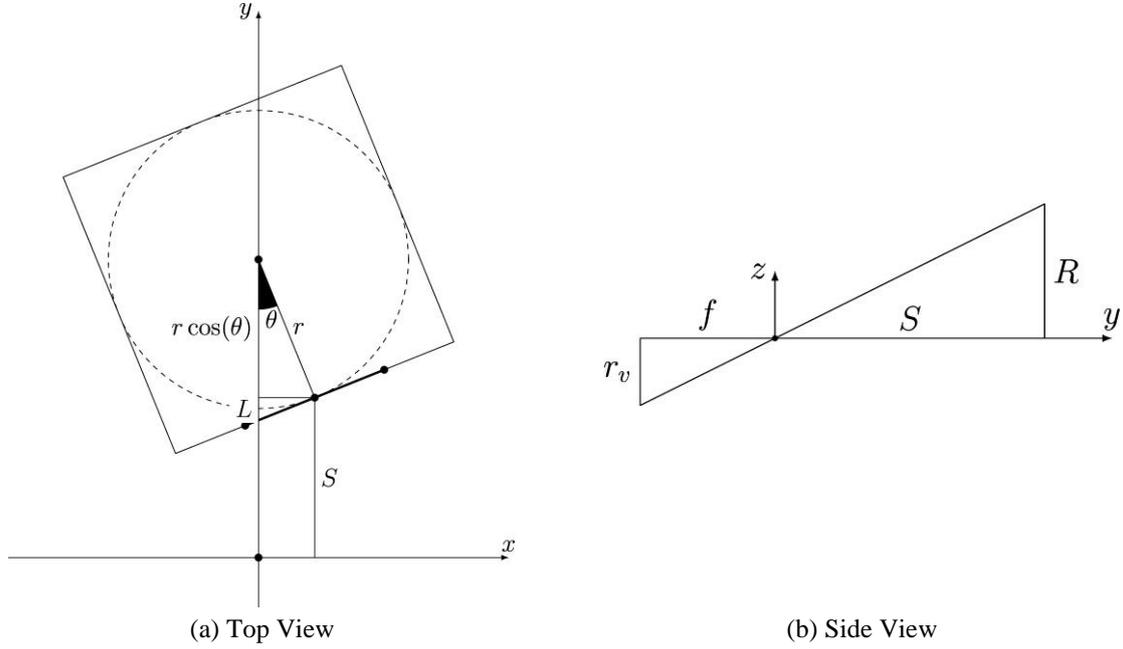


Fig. 2 Geometry of vertical radius solution

the shell of the joint and  $\theta$  the angle of rotation of the joint. Since the radii of the ellipse have been calculated, its area is equal to

$$A = \pi r_h r_v \quad (8)$$

### 2.1 Angles of no visibility

Referring to the cross section of Fig. 1, the monitoring camera is placed so that the camera vector passes through its center. By plotting all the straight lines from the circles to the camera lens it becomes obvious that not all circles can be visible at the same time Fig. 1(b). Since those straight lines pass through the origin of the coordinate system they can be expressed as

$$y = ax \quad (9)$$

A circle can be barely visible when its two lines coincide. This means slope equality, which leads to the following expression

$$\frac{x_u \sin(\theta) + y_u \cos(\theta) + c_y}{x_u \cos(\theta) - y_u \sin(\theta) + c_x} = \frac{x_v \sin(\theta) + y_v \cos(\theta) + c_y}{x_v \cos(\theta) - y_v \sin(\theta) + c_x} \quad (10)$$

Solving this equation reveals the angles for which the targets are barely visible to the camera. The values of those angles depend on the distance of the camera from the center of rotation.

Having solved the geometry of image representation on the sensor of the camera it is possible to plot the visible target area on the camera sensor as a function of the rotation angle of the joint.

As can be seen in Fig. 3, initially one target is visible and as the joint rotates the visible area of

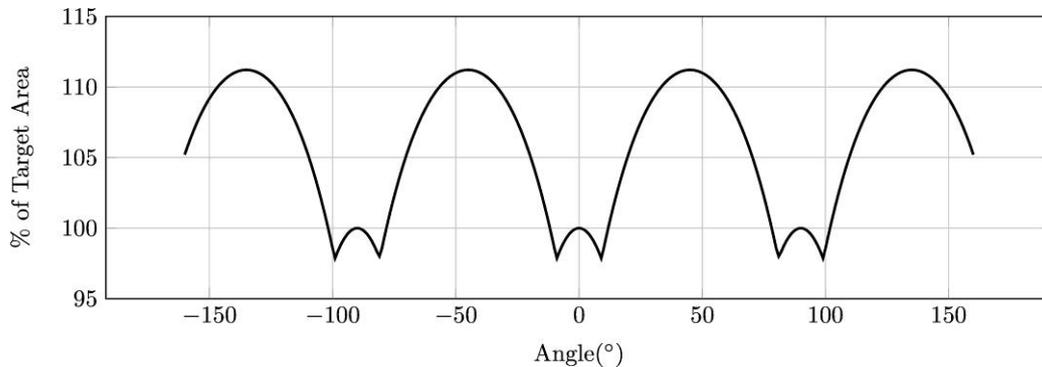


Fig. 3 Area mapping for a joint of square cross section

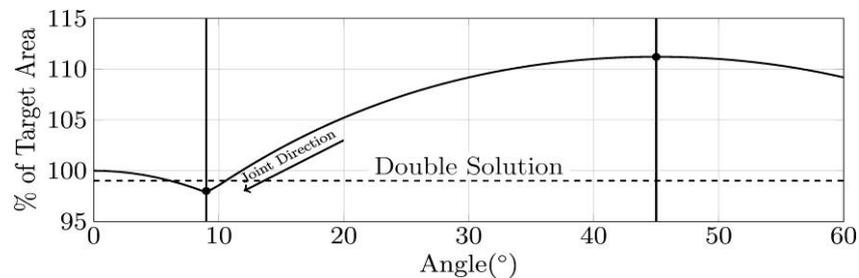


Fig. 4 Curve points with the constant rotation direction constraint

the target is reduced. Then the second target becomes visible and the total visible area begins to grow until it reaches its maximum value. After that point the situation is reversed until one target is again available and so on. The vertices observed in the lower part of the curve are the points where targets become hidden or visible. Those correspond to the solutions of Eq. (10).

Note that in some regions of the map the total visible area rises above 100%. This is due to the simultaneous visibility of two targets which can occur between two successive critical visibility points. In those regions the total visible target area amounts to a total greater than the area of an individual target. These regions alternate with regions where only one target is visible.

## 2.2 Curve Localization and color

As can be seen from Fig. 3, no unique solution exists for a given projected image area. Since the area is measured, it is imperative to define a solution range. The shape of the curve reveals the segments that have unique angle solutions, therefore the only remaining problem is the selection of the correct curve segment. Initially it was a prerequisite for the solution that the robotic arm starts from a known position with a known solution range. During the motion of the arm, as the joint reaches the limits of its current solution range, the solver would choose another curve segment and therefore a solution range based on the position of the joint in the current solution range. But the system has no way to detect a change in rotation direction, if it happens in the vicinity of local minima and maxima as can be seen in Fig. 4.

As a consequence, the joint should maintain constant rotational direction. This obviously renders such a solution impractical. To overcome the problem of solution localization in the curve, color targets were considered. Since the camera can see a maximum of two targets at any given

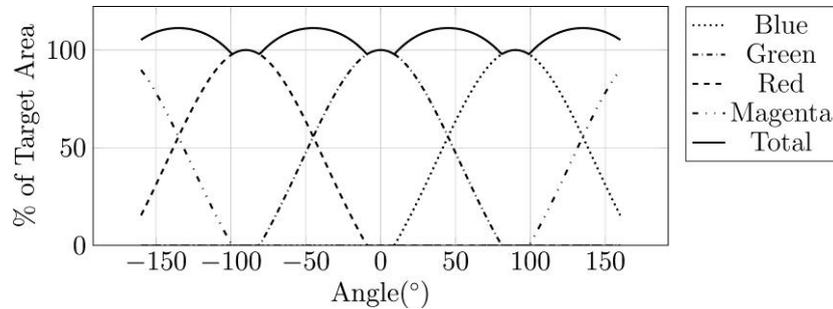


Fig. 5 Total area and color components

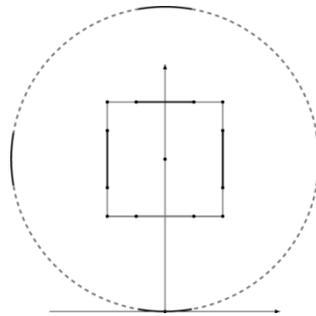


Fig. 6 Camera placement for two color visibility, dashed / full arcs defining the two location ranges

time the combination of colors of visible targets can be used to define the solution range. The visible area for each circle can be seen in Fig. 5.

However, tracking two targets at all times cannot be achieved with just one camera since for small rotations only one target is visible. Therefore a second camera is added. The two cameras lie on the circumference of an imaginary circle centered on the center of joint rotation. Their angular distance is such that one camera can always see two targets when the other camera sees one target. Thus, one camera should be inside the range defined by two consecutive critical angle values, while the other one should be inside the space between two consecutive blind spaces, depicted in Fig. 6 by dashed and full arcs respectively.

### 3. Implementation

The overall architecture of the system is shown in Fig. 7.

The selection of tools that were used in the implementation was based on the following criteria: Low cost, Ability to customize, High performance.

Low cost is an obvious prerequisite, while the ability to customize the underlying software platform was deemed critical since the needs and challenges of the computer vision part of the implementation could not have been anticipated beforehand. Finally, high performance would allow for real time measurements.

Based on the above criteria the use of Matlab™ platform was rejected since its cost and performance were incompatible with the specifications set. Amongst the many available software

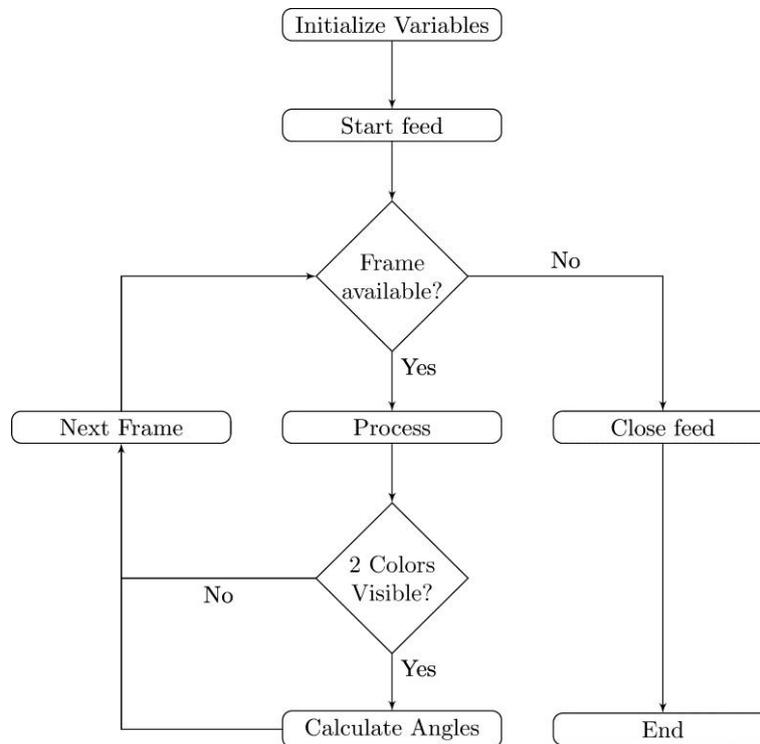


Fig. 7 Overall architecture flowchart

libraries for computer vision OpenCV was selected (Bradski, 2000). OpenCV is a multiplatform library that provides a set of high performance vision routines suitable for the development of vision applications. OpenCV is distributed under the BSD license making it ideal for commercial applications.

The programming language selected was C++ since it is powerful, feature-rich and has wide industry and community support. The toolchain used was gcc compiler and the GNU make buildsystem.

The robotic arm used was a Stäubli RX90 whose first joint has a rectangular cross section.

Two Logitech C525HD cameras were used. Those are typical mid-range commercially available web cameras with high-definition recording capabilities.

### 3.1 Architecture details

The system that was developed was separated into three parts. Those were: the main routine, the common subroutines and the map class. The main routine is responsible for the following: Input of the video, Image preprocessing, Color detection, Image post-processing, Results presentation and storage.

As the main routine does all of the above, it uses the common subroutines and the map class. The common subroutines deal with all vector data for each target, initialization functions and the functions that give the visible area per color and the maximum area per color. It should be noted that the maximum area depends only on the distance of the target from the camera.

The map class was initially implemented for trying various theoretical models, mostly unsuccessful, therefore it contained many functions that later ‘depreciated’ and were removed. However, the existence of this class is still important as it enables the use of multiple maps. This class contains functionality for both the selection of the solution range and the reverse numerical solution of the angle rotation map.

### 3.2 Image processing

Image processing is necessary for data extraction. The procedure is completed in three steps: pre-processing, color detection and post processing. The purpose of pre-processing is to prepare each frame of the video for the color detection. First a region of interest is defined, which speeds up the process since it discards many unnecessary pixels. Then, gauss blurring is applied for noise reduction.

Color detection is based on the values of the RGB components and is the most important processing routine, see Fig. 8. Each color and its shades are defined in the region of the RGB color space corresponding to certain RGB values. The RGB ranges are highly dependent on the lighting so they had to be defined by trial and error. The classification of a pixel to a certain color is accomplished by using a series of logical operations. For example, a pixel is considered to be green if its values are in the specified RGB ranges and G component is greater than B component.

Similarly for a color to be blue B component must be greater than G component. For colors other than R, G, B absolute differences of the combined colors can be used. Since colors depend very much on lighting the RGB ranges change in different lighting conditions. During the color detection phase the detected pixels are assigned the respective maximum values so that later processing is straightforward.

During the post processing phase, the image is split into its channels and thresholding is applied to each channel. Thresholding is easy because the detected pixels have been marked beforehand with maximal values. After the threshold operation, the produced binary images of each channel display only the detected color pixels. In the case where more colors have to be detected, more custom channels have to be added to the image. To remove any false positives that have remained from the detection routine a standard erosion is applied (Russ, 2011) with a kernel big enough to remove noise, but small enough so that no valuable information is discarded.

After post-processing the area for each color is

$$A = \frac{A_{\max} P}{P_{\max}} \quad (11)$$

where  $A_{\max}$  is the maximum projected image area on the camera sensor and  $P_{\max}$  is the maximum number of pixels that can be seen for each color. The  $P_{\max}$  value corresponds to the point where the visible surface area is maximum, i.e. when the surface (normal) vector is parallel to the camera vector. The values of  $P_{\max}$  can be found by performing an initial calibration recording before the system is actually set into operation.

### 3.3 Experimental setup and procedure

The experimental setup comprised of the robotic arm and the two cameras, see Fig. 9. The first camera is mounted on a wall, while the second camera is mounted on a tripod placed at the same

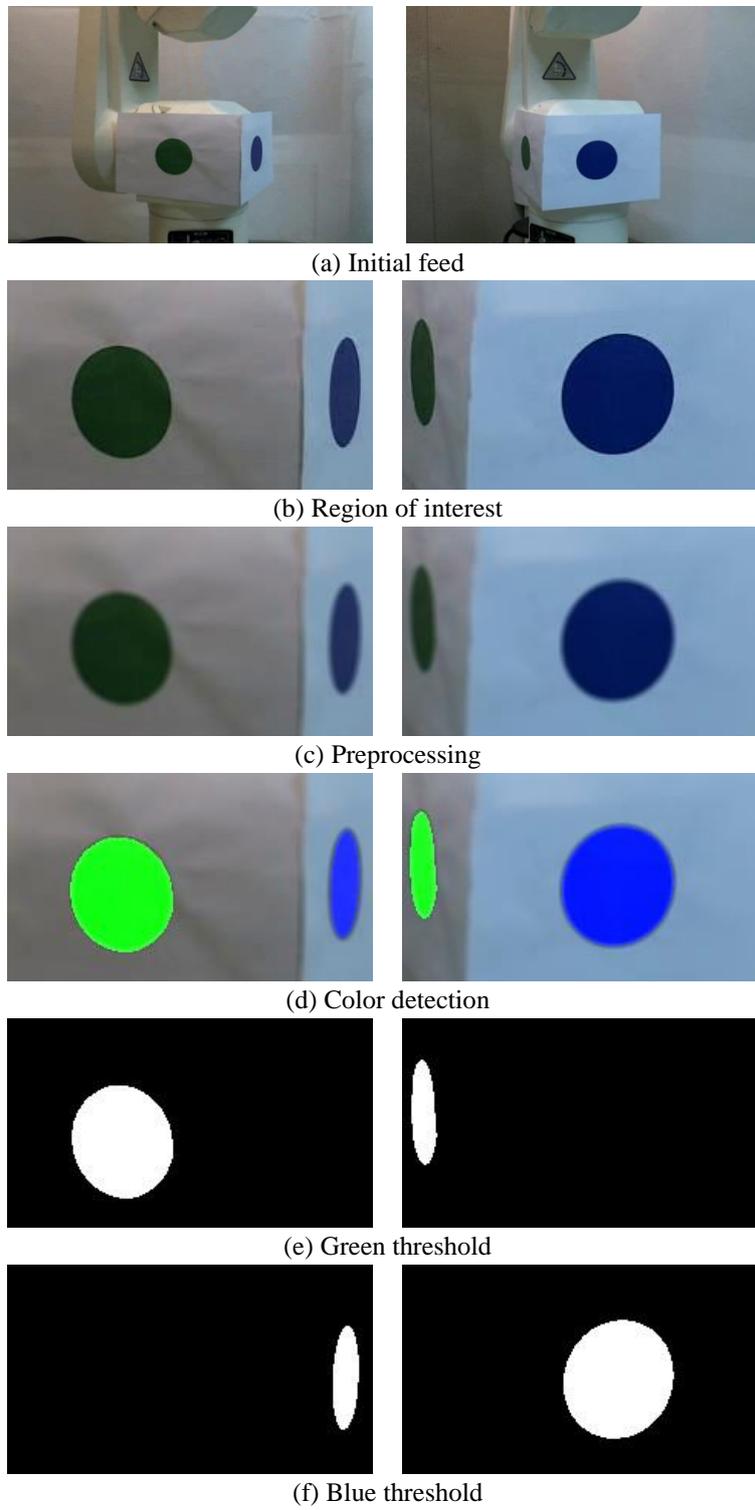


Fig. 8 Color detection from the left and right cameras



Fig. 9 Setup of the robotic arm.

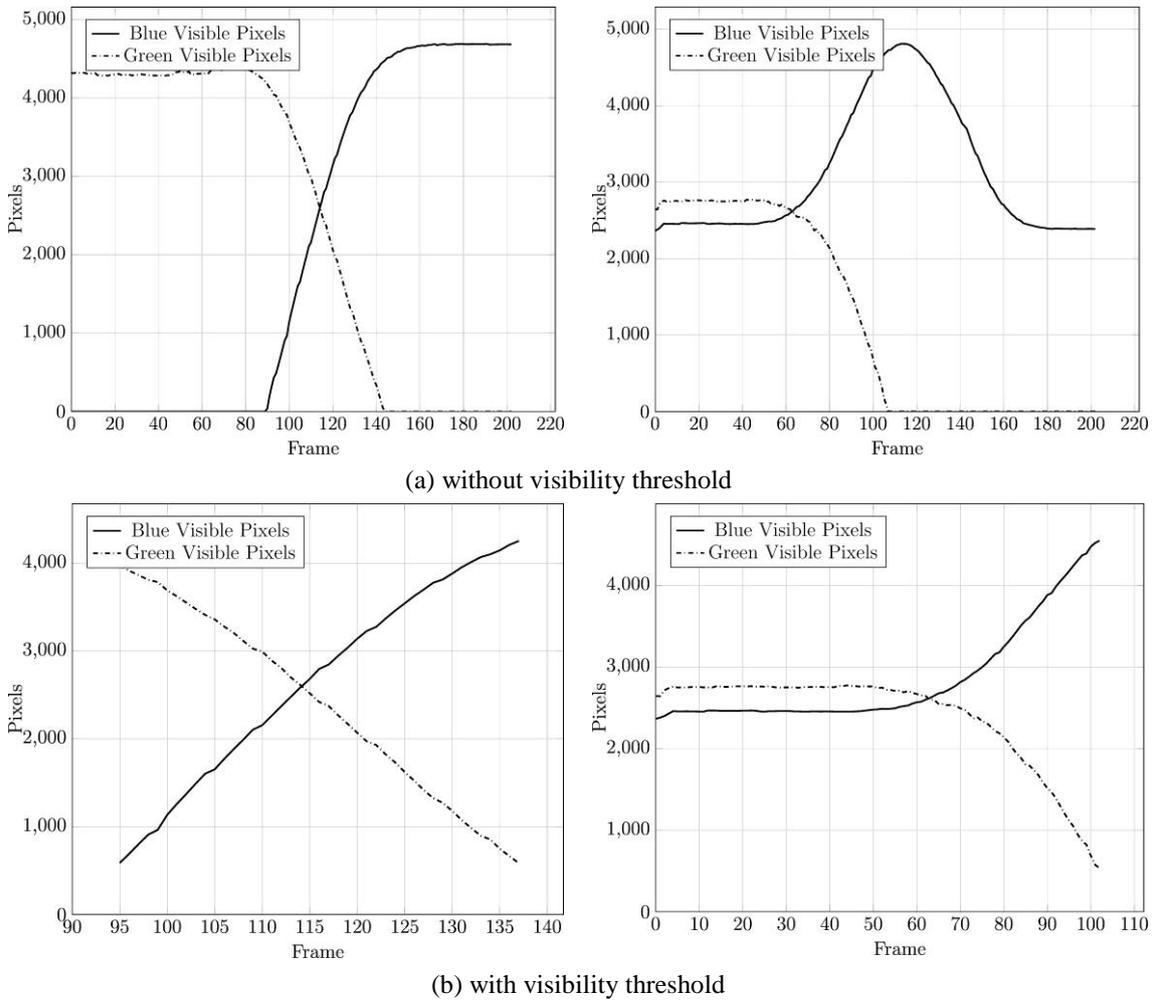


Fig. 10 Visible pixels from left and right camera

distance but at a different angle. The targets were printed on paper which was then glued on the corresponding sides of the robotic joint. Before the actual measurements take place the robot is commanded to take the starting position. The starting position is set to be where the green target has the maximum area facing one of the two cameras. This was chosen so as to accommodate a match of the zero angle of the model and the zero angle of the joint. Camera placement and distance measurement is achieved with the help of a custom-built low power distance measuring laser system of 1 mm accuracy, similar to the Leica Disto D5<sup>TM</sup>. After the hardware is set up an initial recording is made in order to determine the  $P_{max}$  values for each target.

#### **4. Results**

Measurements were taken for an angular range of  $90^\circ$  as a proof of concept. The rotation of the joint was counterclockwise therefore the range of the measurements is  $(-90^\circ, 0^\circ)$ . Initially the robot stays still and then it is rotated by  $90^\circ$ . Note that the two cameras are synchronized, i.e. each frame corresponds to the same joint position. Visibility threshold was arbitrarily set to 500 pixels for each target. For each frame of the video the pixels were counted, see Fig. 10. Subsequently the angle was calculated by reverse solving the angle area map, see Fig. 11.

In Fig. 11 it is obvious that there is some variance in the number of detected pixels that differ from color to color. This variance results in different measurements of the joint angle between the two targets.

It can also be observed in Fig. 11 that for some frames both cameras can take measurements and the corresponding measurement segments have the same slope but not the same values. This is seemingly caused by a different error applicable to each camera as discussed in the next section.

#### **5. Errors**

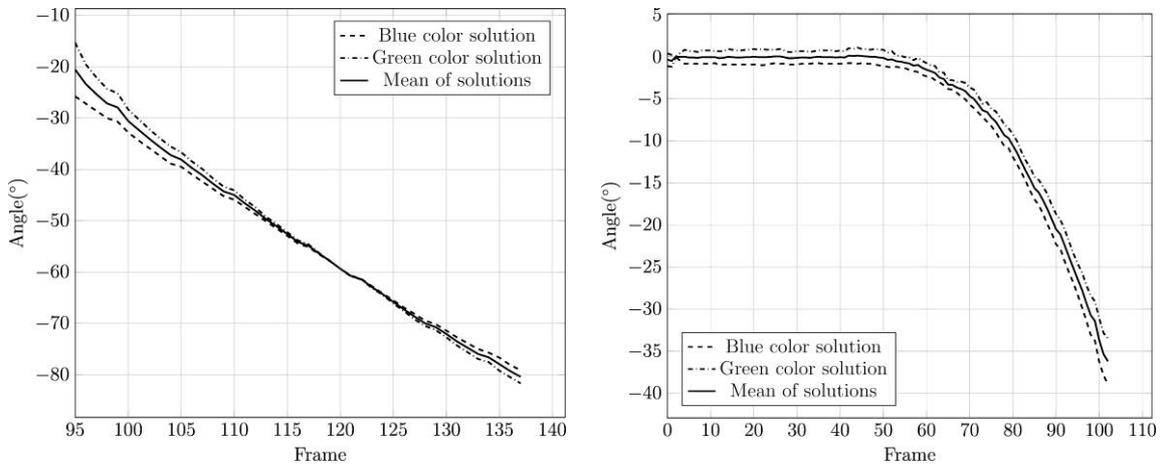
There are many noise sources that affect the measurements acquired but not all of them are subject to formal analysis. Lighting and target deformation are examples of the latter category, Errors in camera placement are a prime example where analysis can improve the results obtained by image processing.

##### *5.1 Lighting*

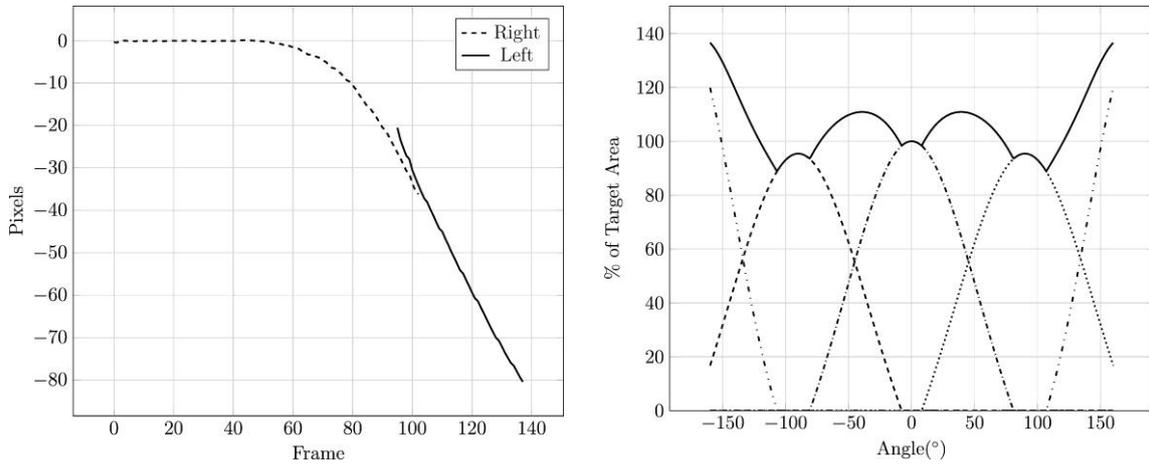
It is common for vision applications to need specific lighting conditions. In this application the conditions were not ideal. The existing lighting installation created shades and glaring on the targets. The problem is that the effect of lighting tends to be more or less ‘chaotic’. Achieving good, let alone optimal, lighting conditions requires experimentation. Overall ambient lighting conditions are needed to achieve color ranges that are stable throughout joint rotation.

##### *5.2 Target deformation*

Another ‘random’ phenomenon is the deformation of the target and its deviation from the ideal shape considered in the analysis. The deformations are caused by the adhesion of the target on the body of the robot. This can be eradicated by having the manufacturer paint the targets on the body



(a) Left and right camera solutions from each color and mean value of solutions



(b) Comparison of mean solutions both cameras

(c) Solution map employed

Fig. 11 Results

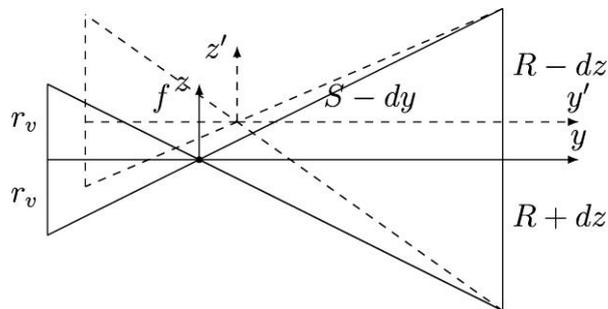


Fig. 12 Geometry of camera deviations in the calculation of the vertical radius

of the robot. Alternatively, hard targets made out of plastic or metal and coated by mat paint can be used.

### 5.3 Camera deviation from ideal position

This refers to the errors introduced by the deviation of the cameras from their ideal model positions that were assumed in the calculations of section 2. This type of error can be quantified. To investigate the effects that the camera deviations can have on the area-angle map, a computational simulation was carried out, see Fig. 12.

Supposing that the camera deviates from its ideal position by cartesian components  $dx$ ,  $dy$ ,  $dz$  the mathematical model can be modified as follows. For the horizontal radius Eq. (2) is modified to include camera displacements

$$\begin{aligned} P_{ux} &= x_u \cos(\theta) - y_u \sin(\theta) + c_x - dx & P_{uy} &= x_u \sin(\theta) + y_u \cos(\theta) + c_y - dy \\ P_{vx} &= x_v \cos(\theta) - y_v \sin(\theta) + c_x - dx & P_{vy} &= x_v \sin(\theta) + y_v \cos(\theta) + c_y - dy \end{aligned} \quad (12)$$

while for the vertical radius Eq. (7) is modified as

$$S = L - dy - r \cos(\theta) \quad (13)$$

It should be noted that  $dz$  also affects the vertical radius  $R$ . This effect is both negative and positive, if the radius is computed by the radius originating from the center upwards or from the center downwards, respectively. In this implementation the mean of those two radii is computed and therefore the effect of  $dz$  is canceled out.

In Fig. 13 the map of Fig. 11(c) can be observed for different values of  $dx$ ,  $dy$ ,  $dz$ . It can be observed that  $dz$  is the only parameter that seriously affects the produced map. Additionally, it can be seen that errors are bigger in the regions around  $\pm 45^\circ$  and  $\pm 160^\circ$  values. Another notable feature is that the eccentrically rotating target has reduced sensitivity as can be seen in the regions  $(-160^\circ, -110^\circ)$  and  $(110^\circ, 160^\circ)$ , where the corresponding target comes closer to the camera compared to the other targets; this is due to the reduced ratio of changing pixels to the max number of pixels.

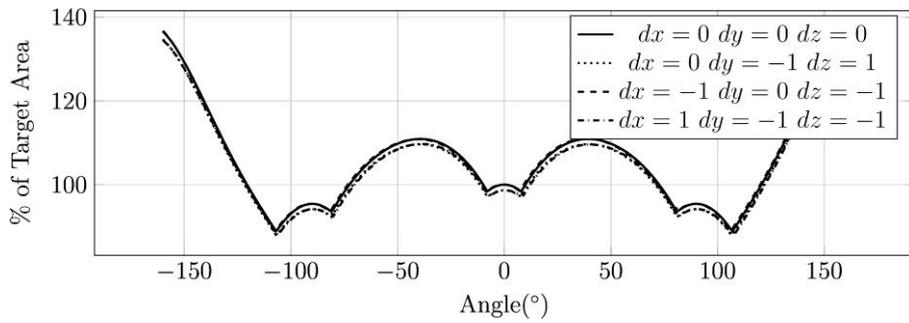
### 5.4 Experimental error measurement

To quantify the errors that exist in the system the robot was operated for known rotations of the measured joint. At every position 100 frames were taken from each camera, so that the typical noise can be manifested. Then, the mean of the measurement for each position was computed. In Fig. 14(a) the real position of the joint is plotted along with the measurements taken from the two cameras. The mean deviation from the real position of the joint was  $2^\circ$  for the right camera and  $4.07^\circ$  for the left camera. The max deviation is  $7^\circ$  and is observed in the left camera. The difference in errors between the two cameras can be attributed to the different lighting (as this is seen from the vantage points of the cameras) but also, and most notably so, to the different placement deviations amongst the two cameras.

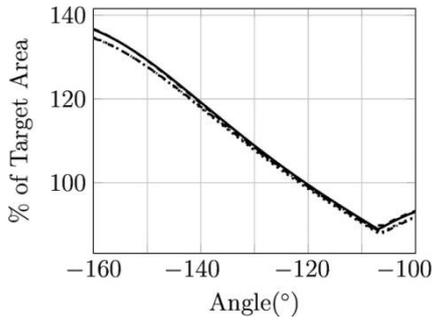
### 5.5 Estimation of camera deviations and measurement correction

Since the effect of camera deviations on the mapping can be formally described, those deviations can be solved for, if the results of measurements that are obtained from the model are compared with known positions of the joint.

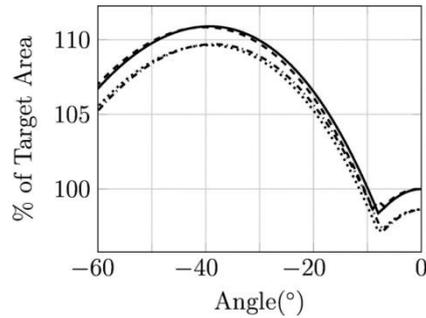
It must be noted that each camera has its own set of  $dx, dy, dz$  parameters. Normally those parameters should not change between different measurements since the camera position does not change. However, this would happen only on the premise that all other types of errors are zero. This is not true, of course. Moreover, those errors are also subject to variation, which, however, cannot be accounted for. Thus, it is as if  $dx, dy, dz$  parameters vary, too.



(a) Real map and effects of camera deviations from ideal positions  $d_i$  in cm

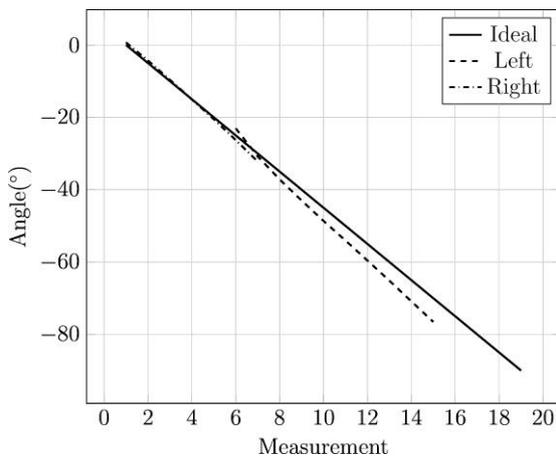


(b) Curve detail in the  $(-160^\circ, -100^\circ)$  range

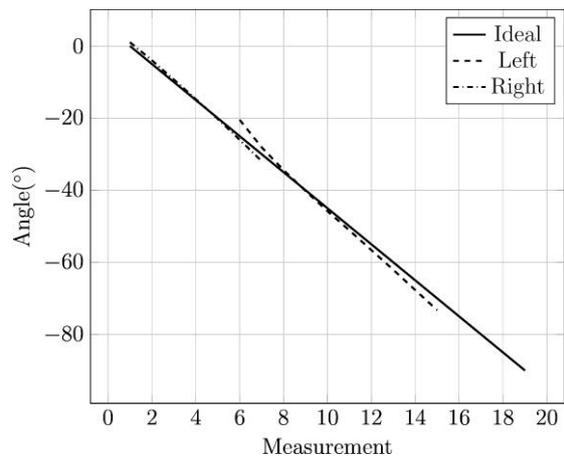


(c) Curve details in the  $(-60^\circ, 0^\circ)$  range

Fig. 13 Effects of camera deviations



(a) without compensation for camera deviations



(b) with compensation for camera deviations

Fig. 14 Experimental measurement of error with a step of  $5^\circ$

For the purpose of measurement correction mean values for  $dx, dy, dz$  were calculated. This can be formally described as follows. Let there be a set of camera deviations  $CD=\{dx, dy, dz\}$  different for each camera. In every position each camera takes  $n$  frames and produces  $n$  measurements all with different configurations of the  $dx, dy, dz$  parameters. Therefore for each position there is a set of measurements  $M$ . The parameters that are chosen for each position are placed in a set as follows

$$D_k = \{CD : \min(|m_i - m_o|)\} \quad m_i \in M \tag{14}$$

where  $m_o$  is the known rotation of the joint in each position and  $k$  the number of the known positions tested. From those sets it is possible to create a matrix with three columns, each one consisting of the corresponding elements of the  $D$  sets

$$A = \begin{bmatrix} dx_1 & dy_1 & dz_1 \\ \vdots & \vdots & \vdots \\ dx_k & dy_k & dz_k \end{bmatrix} \tag{15}$$

After this operation, the  $dx, dy, dz$  parameters for each camera can be calculated as

$$dx = \frac{\sum_{i=1}^k A[i,1]}{k} \quad dy = \frac{\sum_{i=1}^k A[i,2]}{k} \quad dz = \frac{\sum_{i=1}^k A[i,3]}{k} \tag{16}$$

The solution range for each set was selected intuitively in a reasonable range of 3 cm in each direction. The corrected measurements that were processed with the use of the calculated set of parameters for each camera can be seen in Fig. 14(b). After correction the mean deviation from the real position of the joint was  $1^\circ$  for the right camera and  $2.5^\circ$  for the left camera, while the max deviation is  $4^\circ$  and is observed in the left camera.

## 6. Conclusions

In this paper an optical computer vision based system was developed for the measurement of joint angles of robotic arms. The system enables the user to extract joint angle measurements without accessing the robot's programming environment. The physical principle on which the system was developed was the apparent deformation of targets placed on the body of the robot as those move in front of fixed cameras placed around the robot. Standard machine vision routines were used to acquire camera images and process them as to count pixels of respective targets. Target visibility issues were dealt with by assigning different colors to different targets and correspondingly duplicating cameras so as to never loose target visibility completely.

The results that were produced satisfied the initial specifications and this was verified by experimental quantification of the errors introduced by the system. A computational exploration of the errors introduced by camera positioning deviations was conducted that revealed that the mapping between the angle and visible area is sensitive only along the  $z$  axis.

Almost all sources of noise observed in the experimental process can be rectified or completely eliminated with the exception of lighting. More specifically, the error induced by possible – yet minute - deformations of targets can be eradicated by using hard targets made out of plastic or metal and coated by mat paint. Camera deviations can be completely modeled and solved for as shown in sections 5.3 and 5.5. Errors induced by the lighting conditions can be reduced by the use

of multiple unfocused lighting sources that create overall ambient lighting conditions which are ideal for this specific application.

Accuracy of the measurements can be improved by using cameras with higher resolution. High resolutions result in a more detailed depiction of a target and combined with better dynamic ranges can reduce the noise introduced by varying lighting conditions but at a higher computational cost. A particular improvement from higher resolutions can be achieved at the points of marginal target visibility since smaller pixels can more accurately approach the small visible curved shape of the targets. In cameras with bigger pixels and therefore smaller resolution, those small visible, but curved, patches would stimulate larger blocks of square or rectangular shaped pixels which, in turn, leads to greater deviations of measured surface that points to different mathematical solutions of the model. This is a matter of future research.

In addition, the present work will be extended to encompass all rotary joints of articulated industrial robots by making use of the forward transformation matrices from the first to the last joint onwards, thereby registering the full robot pose in real time.

## References

- Ballard, D.H. (1981), "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recogn.*, **13**, 111-122.
- Bradski, G. (2000), "The OpenCV Library", Dr. Dobb's Journal of Software Tools.
- Camarillo, D.B., Loewke, K.E., Carlson, C.R. and Salisbury, J.K. (2008), "Vision based 3-D shape sensing of flexible manipulators", *Proceedings IEEE International Conference on Robotics and Automation*, Pasadena, California, May, 2940-2947.
- Chen, S., Li, Y. and Kwok, N.M. (2011), "Active vision in robotic systems: a survey of recent developments", *Int. J. Robot. Res.*, **30**, 1343-1377.
- Di Leo, G. and Paolillo, A. (2011), "Uncertainty evaluation of camera model parameters", *Proceedings 28<sup>th</sup> IEEE Instrumentation and Measurement Technology Conference*, Hangzhou, P. R. China May, 6.
- Fung, J. and Mann, S. (2004), "Using multiple graphics cards as a general purpose parallel computer: applications to computer vision", *Proceedings of the 17<sup>th</sup> International Conference on Pattern Recognition*, Cambridge, UK, August, Vol.1, 805-808.
- Fung, J., Tang, F. and Mann, S. (2002), "Mediated reality using computer graphics hardware for computer vision", *Proceedings 6<sup>th</sup> International Symposium on Wearable Computers*, Washington, USA, October, 83-89.
- Golnabi, H. and Asadpour, A. (2007), "Design and application of industrial machine vision systems", *Robot. Comput. Integr. Manuf.*, **23**, 630-637.
- Heisele, B. and Ritter, W. (1995), "Obstacle detection based on color blob flow", *Proceedings of the Intelligent Vehicles '95 Symposium*, Detroit, USA, September, 282-286.
- Jungel, M., Mellmann, H. and Spranger, M. (2008), "Improving vision-based distance measurements using reference objects", in: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (Eds.), *RoboCup 2007: Robot Soccer World Cup XI*, Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 89-100.
- Li, W., Jin, J., Li, X. and Li, B. (2010), "Method of rotation angle measurement in machine vision based on calibration pattern with spot array", *Appl. Optics*, **49**, 1001-1006.
- Ming, A. and Ma, H. (2007), "A blob detector in color images", *Proceedings of the 6<sup>th</sup> ACM International Conference on Image and Video Retrieval*, Amsterdam, Netherlands, July, 364-370.
- Mirghasemi, S. and Banihashem, E. (2009), "Sea target detection based on SVM method using HSV color space", *Proceedings of 2009 IEEE Student Conference on Research and Development*, Serdang, Malaysia, November, 555-558.
- Russ, J.C. (2011), *The image processing handbook*, Taylor & Francis, 6<sup>th</sup> edition, Boca Raton.

- Su, C.T., Chang, C.A. and Tien, F.C. (1995), "Neural networks for precise measurement in computer vision systems", *Comput. Ind.*, **27**, 225-236.
- Tyrrell, J.A., Lapre, J.M., Carothers, C.D., Roysam, B. and Stewart, C.V. (2004), "Efficient migration of complex off-line computer vision software to real-time system implementation on generic computer hardware", *IEEE T. Inf. Technol. B.*, **8**, 142-153.
- Wang, W.G., Hu, Z.Y. and Shun, Z.Y. (2002), "3D shape measurement based on computer vision", *Proceedings 2002 International Conference on Machine Learning and Cybernetics*, Beijing, P.R. China, November, Vol. 2, 910-914.
- Wu, J. and Xiao, Z. (2010), "Video surveillance object recognition based on shape and color features", *Proceedings 3<sup>rd</sup> International Congress on Image and Signal Processing*, Yantai, P.R. China, October, 451-454.

CC