FIRA Robot World Congress 2003
October 1 – 3, 2003, Vienna, Austria

CALIBRATION AND AUTO-TUNING OF A REAL-TIME RLE VISION SYSTEM

Chris Messom[1], Nazeer Ahmed[2], Gourab Sen Gupta[2,3]

[1] *Institute of Information and Mathematical Sciences,*
Massey University, New Zealand.
C.H.Messom@massey.ac.nz
[2] *School of Electrical and Electronic Engineering,*
Singapore Polytechnic, Singapore.
{nazeer@sp.edu.sg, SenGupta@sp.edu.sg}
[3] *Institute of Information Sciences and Technolgy,*
Massey University, New Zealand.
G.SenGupta@massey.ac.nz

Abstract

This paper introduces three techniques for auto-tuning a run length encoded (RLE) robotic vision system. These are a noise limiting technique and two iterative enlargement techniques, an incremental one and a global one. The incremental iterative enlargement update process is applied after each pixel is processed while the global iterative enlargement update process is applied after processing the whole image. A calibration technique for auto-positioning of the vision system's camera is also presented as well as an automatic tracking algorithm for camera pan, tilt and zoom control. The image processing system discussed operates in real-time (16.67ms), within the 60 fields per second field rate of an NTSC video signal. The auto-tuning can take place online but concurrently in an asynchronous manner satisfying soft real-time constraints (66ms). The colour signatures of the objects to be identified are YUV range values since they are more robust to light intensity variation than the RGB values given by the image capture card. Experiments show that the calibration and auto-tuning is reliable and robust, performing well in variations of light intensity of 800 - 2000 lux.

Keywords : Real-time vision systems, calibration and auto-tuning

1. Introduction

Robot soccer is a research platform for the testing and development of systems ranging from high-level software down to low-level hardware. There has been significant development in the artificial intelligence approaches applied to robot soccer (especially in the simulation leagues [1, 2]). However the real robot leagues still suffer from the significant challenge of creating robust hardware platforms to support the intelligent software systems, which drive the planning and motion control algorithms [3, 4]. One area that remains a challenge is the development of a reliable high performance real-time vision system [5]. In ideal lighting conditions and large setup times good progress has been made, but the robustness of these vision systems is questionable when the lighting requirements are not met, such as exhibition venues with low ambient light levels and non uniform lighting across the field due to spot lighting [6].

Industrial vision systems solve the calibration problem by controlling the lighting conditions to the point that setup can be completed offline. Laborious and time consuming calibration methods may be suitable as they only need to be carried out on installation and perhaps modified during servicing. This approach is not suitable for robotic systems that operate in several different environments with differing lighting conditions. During a game in robot soccer competition the lighting conditions are fixed but they can vary between different venues. This leaves most researchers with the problem of having a perfectly operating vision system in their own labs, but a less than perfect system during competitions. This can be overcome by efficient and optimal calibration and auto-tuning techniques.

The research presented in this paper shows that much of the manual calibration and tuning of vision systems that currently occurs can be automated. The techniques discussed are transferable to robotic vision systems in industry where auto-calibration and tuning will improve quality and decrease the implementation and maintenance costs. The techniques presented can be extended to slowly varying lighting conditions since the auto-tuning algorithm can execute within soft real-time constraints (66 ms).

2. Run Length Encoding

Image processing algorithms are computationally extremely expensive since each pixel in the image must be processed. For this reason real-time performance has often only been possible on dedicated hardware implementations [7]. Vision systems are the primary sensor input to advanced robotic applications such as mobile robotics, agricultural robotics and inspection systems [8]. Most commodity vision systems use video signals as input to the image capture subsystems and so provide frame rates of approximately 30Hz and field rates of 60Hz for interlaced images [9]. Processing these images at useful resolutions (240x320 and above) in the 33.3 ms and 16.67 ms sample times respectively represents a significant challenge. The image processing must not dominate the sample time as the intelligent control algorithms required to control the robots will have little chance to satisfy the real time constraint.

Run Length Encoding (RLE) has been used to compress images. Essentially sequences of pixels of the same colour are represented as just two numbers: the pixel value and the number of pixels. The position of this sequence of pixels is also stored. For black and white images the compression algorithm is very efficient, especially if there are only a few large objects in the image. RLE has been investigated as a real-time image processing technique as it is useful when processing a large high-resolution image [10, 11]. A large part of the computational effort is wasted with standard image processing techniques when most of the image is background, or the body of a large object. Using RLE techniques, these pixels can effectively be skipped over so that the computationally expensive image processing algorithms can process just relevant pixels resulting in a significant improvement in performance.

RLE systems have proven to be efficient real time image processing systems for robotic applications. This is the case since most robotic applications are in domains where there are large objects (such as mobile robots in factories) and/or large areas of background (such as security systems). The RLE algorithm effectively produces real-time image processing systems that can give timely response in highly dynamic environments [10,11].

The limitations of the RLE identification algorithms have been in the requirement that each object's colour signature be specified for particular lighting conditions. Each object needs to be identified as occupying a subspace (normally convex) of a three dimensional colour space (for example YUV or RGB), see figure 1. In this way a lossful compression of the image is created, since all run lengths of pixels within one object’s colour signature are compressed into a single item. This loss is not a problem since the particular shade of colour is not important, just the object size and position.

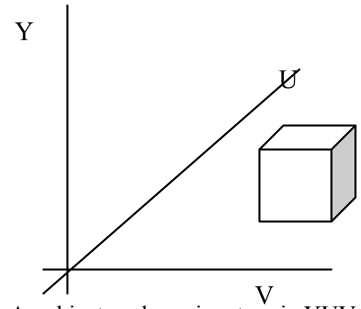

Figure 1. An objects colour signature in YUV colour space

A potentially laborious calibration process has to be completed before the RLE system can be used [12]. Every object’s colour signature has to specified so that each pixel in the image can be classified as being part of one object or part of the background. This paper introduces an auto-tuning method that optimises the colour signatures that the RLE system uses, so that the objects in the field of view are reliably and robustly identified.

3. Camera Calibration

Before the colour auto-tuning can occur the camera system must be calibrated. The camera is positioned approximately 2 to 2.5m above the robot soccer field. Ideally the soccer field should fill the image, but due to differing aspect ratios between the field and image a best-fit approach has been adopted. Finally a transformation from image to world coordinates must be provided but this is normally a simple scaling, translation and rotation operation if the camera is correctly positioned and the lens distortion is minimal, see figure 2.

The Canon VC-C4 NTSC camera used in this system has zoom, pan-horizontal and pan-vertical functionality. This system has been augmented with an auto-calibration capability so that the setup time for the system is considerably reduced. For a fixed camera system, the camera must be mounted precisely in position and orientation. With auto-positioning the camera can be mounted with a large variation in position and later automatically realigned for optimal viewing of the field. This reduces the

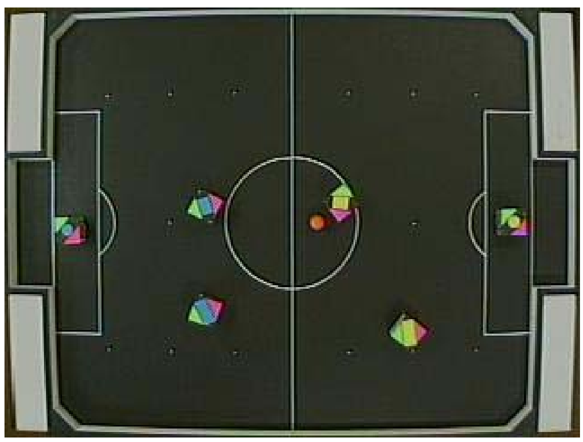

Figure 2. Overhead view of the robot soccer field

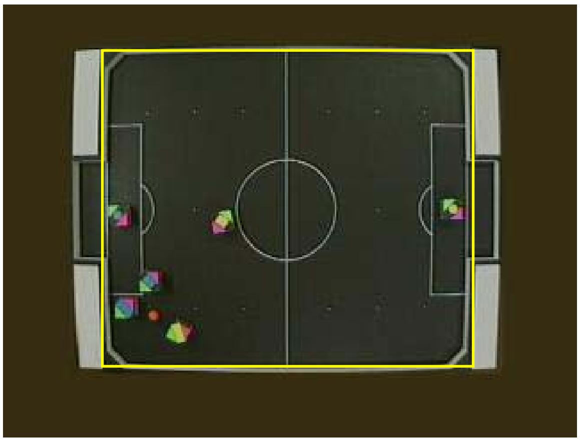

Figure 3. Specification of zoom

setup time required while still achieving an optimal image. The one limitation of the system is a lack of a rotation around the viewing axis. A rotation operation will be required in the image to world coordinate transformation of the current system if the field and image are not aligned.

The current system has a user controlled auto-positioning system. The camera's home position is in the centre of its range of motion for the horizontal and vertical pan and fully zoomed out. In the zoomed out position the field should be fully in view. The user must identify the boundaries of the field of view with a mouse click and drag on the image displayed on the user interface, refer figure 3. The system automatically centres the camera by panning horizontally and vertically the required amount based on the angular movement of the view angle that is required to place the camera image centre at the same position as the centre of the selected field of view, eqn 1.

$$\Delta Angle_H = \arctan((SC_x - IC_x)/CH)$$
$$\Delta Angle_V = \arctan((SC_y - IC_y)/CH) \qquad (1)$$

Where $SC_{x,y}$ is the coordinate of the centre of the selected region, $IC_{x,y}$ is the coordinate of the centre of the image, (both in world coordinates) and CH is the height of the camera above the field. The correct zoom setting is obtained from the user specified boundary of the field of view. The system then zooms the required amount, based on the minimum of the optical zoom available for the horizontal and vertical axis, refer figure 4.

$$ZoomPercentage = \min(IS_x/ FS_x, IS_y/ FS_y) * 100\% \qquad (2)$$

Where $IS_{x,y}$ is the size of the image and $FS_{x,y}$ is the size of the field, both in pixels.

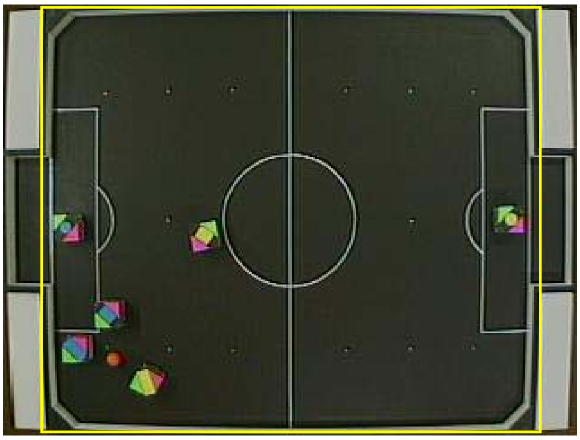

Figure 4. Zoomed image

The user interface also provides zoom and pan controls so that an expert operator can choose different zoom and pan positions. This is not essential but provides additional control for the expert user. Although this calibration procedure is not labour intensive a fully autonomous camera positioning system is also possible. This system will use edge detection to identify the lines on the field so that the centre of view and field boundaries can be automatically identified. The correct pan and zoom values can then be autonomously obtained without user intervention.

Once the camera has been correctly positioned the camera calibration process is then completed. The image to real world coordinate transformation is determined by the position of the field boundaries. In real world coordinates the origin is located at the bottom left hand corner of the field, the x axis is in line with the horizontal side line, while the y axis is in line with the vertical goal lines. The position of the top right hand corner is (WTR_x, WTR_y). Given the image coordinates of the top right (ITR_x, ITR_y) and bottom left position (IBL_x, IBL_y), assuming square pixels the standard rotation and translation transformation is applied to obtain the image to world coordinate transformation.

$$W_x = [\cos\theta *(I_x - IBL_x) + \sin\theta *(I_y - IBL_y)]*\gamma,$$
$$W_y = [-\sin\theta *(I_x - IBL_x) + \cos\theta *(I_y - IBL_y)]*\gamma,$$
$$\theta = \mathrm{atan}((ITR_y - IBL_y)/ (ITR_x - IBL_x)) - \mathrm{atan}(WTR_y/WTR_x)$$
$$\gamma = \sqrt{[(ITR_x - IBL_x)^2 + (ITR_y - IBL_y)^2]} / \sqrt{[WTR_y^2 + WTR_y^2]} \qquad (3)$$

(W_x, W_y) are the world coordinates and (I_x, I_y) the image coordinates.

4. Auto-tracking Objects

As robot soccer has been extended with a larger number of players, originally 3 players per side, now 7 per side and up to 11 per side in the future, the size of the field has increased. This has introduced the problem that a global vision system is not able to provide a high enough resolution to accurately calculate the robot's position and orientation.

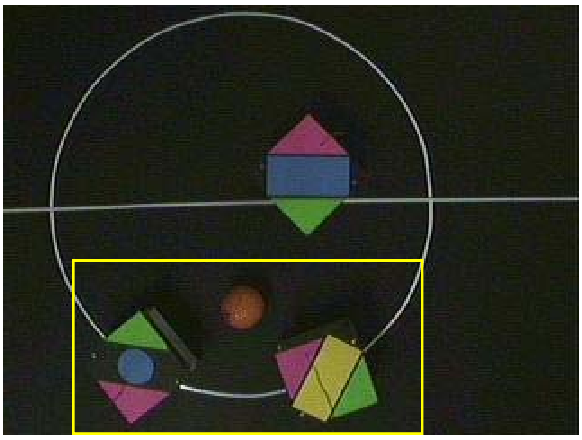

Figure 5. Identification of objects of interests

Multiple cameras in a grid formation over the field [11] offers a solution to this problem but a large number of cameras are needed to get the required accuracy for precision control. An alternative approach is to have a coarse grained global vision system that has an overview of the field and one or more fine grained cameras that can tilt, pan and zoom to any position of interest on the field.

The auto-tracking camera controller has two main modes, tracking and searching. In tracking mode the system follows the object of interest (normally the ball or a robot) as it moves within the image, refer figure 5. When the object is within the centre zone the zoom factor is increased until it reaches the maximum value (figure 6). The highest resolution is obtained in this situation resulting in a more accurate position calculation for the robots and ball. When the object is not within the centre region of the image the camera is panned and tilted until the object reaches the centre region. A tracking system that is continuously adjusted based on error (distance of the object from the centre of the image) will always be moving introducing additional sensor noise into the system. To reduce this effect a gain scheduled control technique is used in which the gain is zero when the object is within the centre zone, the gain is 20 within the intermediate zone and 50 within the extreme zone (refer figure 7).

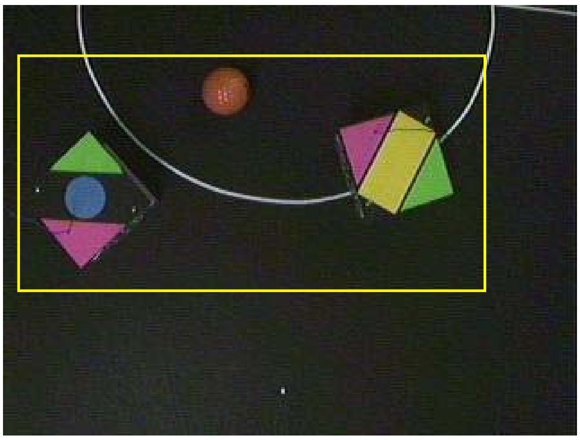

Figure 6. Pan, tilt and zoom to objects of interest

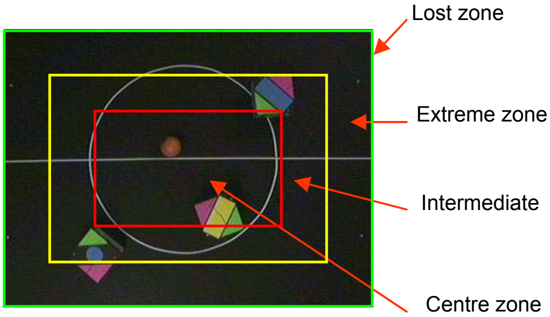

Figure 7. Gain scheduled panning speeds

In search mode the current object of interest is not in the image and so the camera must move to the position of the object. This position can be obtained from the global image system, which is then used to calculate the necessary pan and tilt angles. When the object is only just out of view it can be identified by zooming the image out.

5. Colour Calibration and Auto-tuning

Experiments with RGB and YUV colour spaces show that YUV is more stable. Table 1 shows that the range of RGB values are greater than the U and V ranges for the standard colour patches used in robot identification. Therefore YUV colour signatures were used in this RLE system. Note: most of the variation in the Y component is due to variation in light intensity rather than colour variation.

RGB Colour Recordings							
Yellow				*Purple*			
136	148	15	Min	112	64	80	Min
248	248	88	Max	200	128	152	Max
Green				*Blue*			
56	192	8	Min	64	104	112	Min
112	248	80	Max	80	120	176	Max
YUV Colour Recordings							
Yellow				*Purple*			
189	10	129	Min	106	122	135	Min
208	45	145	Max	131	151	166	Max
Green				*Blue*			
113	34	46	Min	88	143	103	Min
168	94	103	Max	111	163	117	Max

Table 1. Comparison of YUV and RGB colour signatures for standard robot colour tags

In order to calibrate the object's YUV signatures we need to specify the performance function of the calibration technique. For this application a seed point YUV value taken from a pixel on the object must then be expanded to a range of YUV values that are part of the object's signature. This seed value is specified by the user as shown in figure 8. The performance functions used to evaluate the calibration techniques are the number and size of noise artifacts introduced by the object signatures and whether distinct object's have overlapping colour signatures.

Ideally two colour signatures should not overlap, i.e

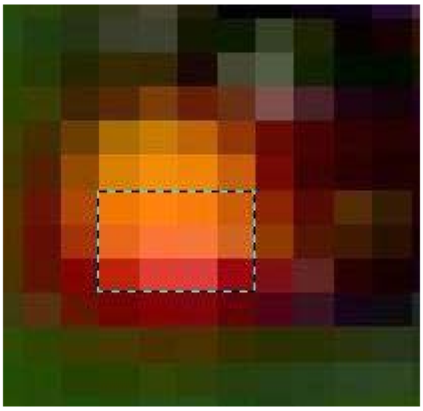

Figure 8. Selection of seed pixels

if CS_1 is the set of YUV values in the colour signature of object 1 and CS_2 is the set of YUV values in the colour signature of object 2 then;

$$CS_1 \cap CS_2 = \varnothing \quad (4)$$

Also the maximum size of the noise artifacts introduced by object 1's (O1) colour signature set CS_1, when run length encoding (RLE) an image I, should be less than a threshold value τ, which will normally be less than half the size of the smallest object in the field;

$$\mathrm{Max}(\ \mathrm{RLE}(\ I \mid CS_1) \otimes O1) < \tau \quad (5)$$

Where the $\otimes$ operator represents a removal of the pixels due to a particular object (in this case object 1). Three approaches have been investigated. The first approach is a noise limiting method that iteratively increases the range of YUV values until the noise artifacts reach the given threshold. The YUV range values are each consecutively expanded by the given increment. When noise artifacts are introduced or there is a colour signature overlap the change is undone. The remaining component's range values are iteratively expanded until the noise limit is again reached. This is an effective approach when τ is small as it tends to limit colour signature overlap with background pixels since RLE algorithm (using a colour signature e.g. CS_1) would treat classifications that are not part of the given object (O1) as noise artifacts, refer figure 9.

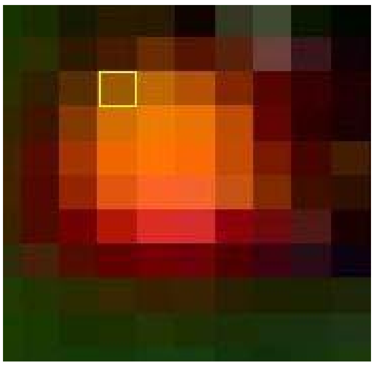

Figure 9. Comparison of selected pixel to colour signatures

The second and third approaches make use of a recursive method that identifies pixels μ that are in position (x,y) within a given area of the image (size Δ) around the object O1 and are within a given colour distance δ to the current colour signature CS_1. The current colour signature is then expanded to include this pixel's YUV value;

if $|\mu(x,y)-O1(x,y)| < \Delta$ **AND** $|\ CS_1-\mu(YUV)| < \delta$ **then** $CS_1 = CS_1 \oplus \mu(YUV)$ (6)

The operator $\oplus$ used in equation 6 expands the set CS_1 by adding the point $\mu(YUV)$ to the set CS_1 including any intermediate points so that the set CS_1 still remains convex.

The difference between the second and third approaches is the point at which the colour signatures are updated. The incremental approach changes the colour signatures as each pixel is processed while the global approach processes the whole image before updating the colour signatures. Although the two approaches are different it is possible to produce a similar result as the incremental recursive update approach by repeatedly applying the global recursive update algorithm.

6. Results

The performance evaluation of the RLE calibration has been carried out on a Pentium IV 1GHZ running Windows 98. The compiler used was Microsoft Visual C++ version 6. The performance of the noise limited tuning of the YUV RLE colour signatures was affected by the calculation of the RGB to YUV linear transformation given by a 3 by 3 floating point matrix multiplication. The conversion is required so that the RLE classification of the image can be completed so that the number and size of the noise artifacts introduced by the new colour signature can be identified. The calculation completed in 66 ms making it unsuitable for real-time auto-tuning of the colour signatures but suitable for offline tuning or auto-tuning in parallel on a separate processor with soft real-time constraints.

The recursive YUV colour tuning algorithm is very efficient, completing in 2ms for each object tuned. This gives the possibility of continuously auto-tuning the colour signatures of the objects being tracked by the system. However this algorithm relies on there being sufficient difference (the tolerance factor in the algorithm) between object colour signatures and background pixels. If the contrast between objects and background is sufficiently large this can be achieved, but noise, colour quantisation effects and other image artifacts can violate this condition. Since the stability of the recursive algorithm is not guaranteed only off-line auto-tuning is carried out on the system.

7. Conclusion

This paper presented a calibration and auto-tuning algorithm for a run length encoded (RLE) vision system. The calibration of the auto-positioning vision system offers quick setup and easy maintenance. The auto-tuning of the object colour signatures allows the system to operate in different environments with differing lighting conditions (800 - 2000), essential for any application outside of controlled industrial environments. The calibration is carried out offline and is not required to satisfy the real-time constraints. Future work will include modification of the auto-positioning system to include an actuated degree of freedom around the axis of view so that a rotation transformation from image to world coordinates is not needed.

A gain scheduled object tracking camera controller which makes use of both the position and velocity error is also under investigation.

8. References

[1] Riley P., Stone P., Veloso M. (2001), "Layered Disclosure: Revealing Agents' Internals", Seventh International Workshop on Agent theories, Architectures and Languages, Proceedings, **Lecture Notes in Artificial Intelligence**, Springer, Berlin, pp. 61–72.

[2] Veloso, M. and Stone, P. (1998), "Individual and Collaborative Behaviours in a Team of Homogeneous Robotic Soccer Agents", **Proceedings of International Conference on Multi-Agent Systems**, pp 309-316.

[3] Arkin R.C. and Balch T. (1998), **Cooperative Multiagent Systems, Artificial Intelligence and Mobile Robots,** D. Kortenkamp, R.P. Bonasso, and R. Murphy (eds), MIT Press.

[4] Kim Yong-Jae and Kim Jong-Hwan (2002), "Univector Filed Navigation Method for Fast Mobile Robots in Dynamic Environment", **Proceedings of the Fira Robot World Congress**, Seoul, Korea, pp 165-170.

[5] Bruce, J., Balch, T. and Veloso, M. (2000), "Fast and Inexpensive Color image Segmentation for Interactive Robots", **Proceedings IROS 2000**, San Francisco, pp 2061-2066.

[6] Xiong Rong and Chu Jian (2001), "A Global Vision System: using Hue Thresholds to Extract feature and recognize", **Journal of Harbin Institute of Technology**, Vol 8, No 3, pp 233-238.

[7] Kim Cheol-Taek, Song Eun-Young and Lee Ju-Jang (2002), "An Active Contour Model for Object Tracking using Prior Contour Information", **Proceedings of the Fira Robot World Congress**, Seoul, Korea, pp 146-149.

[8] Templer, R., Nicholls, H. and Nicolle, T. (1999), "Robotics for meat processing – research to commercialization", **Industrial Robot**, vol 26, number 4, pp 290 - 296.

[9] Ramesh Jain, M., Kasturi, R. and Schunck, B.G. (1995), **Machine Vision**, McGraw-Hill International Editions, Computer Science Series, International Edition.

[10] Sen Gupta, G., Messom, C.H., Demidenko, S., Lim (2003), "Identification and Prediction of a Moving Object using Real-Time Global Vision Sensing", **Proceedings IEEE Instrumentation and Measurement Technology Conference**, Vail, Colorado, U.S.A., pp 1402-1406.

[11] Messom, C. H., Demidenko, S., Subramaniam, K. and Sen Gupta, G. (2002), "Size/Position Identification in Real-Time Image Processing using Run Length Encoding", **Proceedings IEEE Instrumentation and Measurement Technology Conference**, Anchorage, AK, USA, pp 1055-1060.

[12] Luo Ronghua and Hong Bingrong (2002), "A Robust Color Object Tracking Method Based on Improved Seeded Region Growing", **Proceedings of the Fira Robot World Congress**, Seoul, Korea, pp 141- 145.