Abstract
Our research is based on Small Robot League of RoboCup, where a global vision system is applied to track eleven moving objects. To obtain the precise motion information of one ball and ten robots, two kinds of calibration are really important: the color calibration and the camera calibration. This paper investigates a K-Mean-clustering-based color calibration algorithm to obtain accurate color identification, and Tsai camera model [2] [5] to get the calibration parameters for image distortion compensation. Special techniques are implemented to apply these algorithms to achieve best results. Experimental results have shown the robustness of our approach.

1. Introduction
The Robot World Cup Initiative (RoboCup) is an international joint project to promote AI, robotics, and related field, which provides a standard platform for robotic soccer game [1]. A robotic soccer system for RoboCup Small Robot League has two five-robot teams playing a golf ball on a board the size of a table-tennis table. Our team is a fully autonomous system, which has the following two functional modules to perceive the environmental information and make decision for robots:

1) Vision Module
This module adopts global vision to track the moving objects, that is, only one camera is used to capture the entire view of the playing ground. Colors are used to differentiate the two teams of robots, and indicate their positions and orientations. The object detection is therefore a color-based image processing.

2) Intelligent Control Module
It receives the objects’ motion states from the vision module via LAN and analyzes them by the preset strategy to decide what to do in each step. The decisions are finally represented as motion commands broadcasted to robots through wireless communication. The system block diagram is illustrated in Fig.1:

2. Calibration in Vision System
The objective of a vision system is to extract interested information from the environment and provide it to high-level decision. As for Small Robot League in RoboCup, in order to simplify the perception task, the setting of the game is rigidly defined as follows [3]:

- Playing field: Dimensions of the playing surface are 152.5 cm by 274 cm. The floor material is green felt mat or carpet;
- Cameras positioned above the field will be mounted on a beam suspended from the ceiling. The beam will be positioned 3 meters above the field. If both teams agree, and the hosting facilities allow it, another height may be used. Cameras may not protrude more than 15cm below the bottom of the beam.
- Lighting: 700-1000 LUX uniform light is intended, but not guaranteed;
- Color: Six colors are used to differentiate objects (see Table 1).

A typical image captured by camera is shown as Fig.2.

The processing steps in our vision system are as follows:
1. Image acquisition;  
2. Color-based image segmentation;  
3. Blob analysis and position computation;  
4. Object association for two consequential images;  
5. Velocity and orientation computation;  
6. Data transmission via LAN to intelligent control module.

There are two main features in our vision system. The first is color-based object detection. It requires accurate color thresholds. From Fig. 2, we can easily find the following facts:

1) There are many weak yellow pixels scattered on the green playing ground. It is because our lighting has yellow color. Such noise is typical “slat-pepper” noise, that is, the number of noisy pixels is small, they disperse around the background rather than gather in several regions and their luminance values are relatively big.
2) There are many dark green patches around the playing ground board. It is caused by shadow and inconsistent lighting.

In addition, because of other random noise effects in the environment, the image will change a bit at every moment. All of these facts imply that it is necessary to do color calibration for finding proper color thresholds.

The second feature is that barrel distortion in the image should be compensated. According to the specifications of game setting, if assuming the camera is at the height of 2.6 m, the lens view angle is at least 55.57°, which will introduce obvious distortion as shown in Fig. 2. To compensate the distortion in the image, camera parameters are required, which make it necessary to do camera calibration.

In Section 3, we will describe a K-Mean-clustering-based algorithm for color calibration. Section 4 introduces Tsai’s model to compute camera parameters.

### 3. Color Calibration

In our vision system, captured images are in a 24-bit RGB true color format, that is, 24 bits are used to represent a pixel, and each 8 bits is for red, green and blue. There are eight colors used in the experiment (see Table 1).

<table>
<thead>
<tr>
<th>Color</th>
<th>Ideal RGB</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orange</td>
<td>(255,100,100)</td>
<td>Ball</td>
</tr>
<tr>
<td>Yellow</td>
<td>(255,255,0)</td>
<td>One Team</td>
</tr>
<tr>
<td>Blue</td>
<td>(0,0,255)</td>
<td>The other Team</td>
</tr>
<tr>
<td>White</td>
<td>(255,255,255)</td>
<td>Playing field markings</td>
</tr>
<tr>
<td>Green</td>
<td>(0,255,0)</td>
<td>Playing field surface</td>
</tr>
<tr>
<td>Magenta</td>
<td>(255,0,255)</td>
<td>Extra marker for robot ID and orientation</td>
</tr>
<tr>
<td>Cyan</td>
<td>(0,255,255)</td>
<td>Extra marker for robot ID and orientation</td>
</tr>
<tr>
<td>Black</td>
<td>(0,0,0)</td>
<td>Robot top cover</td>
</tr>
</tbody>
</table>

Table 1 Colors used in the experiment.

Our intention is to develop calibration software realizing autonomous detection of the above colors. Let’s first check their distribution in R-G-B color space (see Fig. 3). It is obvious that most of them are at the corner of RGB cube, which provides an opportunity to differentiate them by only using the color distance between them. If an object color in the real environment is as pure as that of its ideal case, the color threshold selection is a trivial matter. It is unnecessary to make any calibration. Unfortunately, it is not true.

There are many problems that need to be considered in the real physical environment:
- The color of lighting affects the objects’ colors.
• Shadows and reflections make the color values not uniform.
• Lighting is not ideally uniform everywhere.
• Noises inherent in camera.

The basic idea of our approach is to collect pixel samples for each object type, and compute the mean and variance of each color for threshold determination by using clustering algorithm. A threshold is deemed as the mean plus or minus several times of the standard deviation [4].

3.1. Color Clustering Algorithm
The K-Mean clustering algorithm is based on the minimization of the sum of square distances

\[ J_j = \sum_{X \in S_j(k)} \| X - Z_j \|^2, j = 1, 2, \ldots, K, \]

where \( S_j(k) \) is the cluster domain for class center \( Z_j \) at the \( k \)th iteration.

A general process of K-Mean clustering algorithm can be divided into four steps:
1. Select \( K \) samples arbitrarily as initial class centers, i.e. \( Z_1(k), Z_2(k), Z_3(k), \ldots, Z_K(k) \); where \( k = 1 \).
2. At the \( k \)th iteration, distribute the pattern sample \( X \) among the \( K \) classes according to the following rule: \( X \in S_j(k) \), if

\[ \| X - Z_j(k) \| < \| X - Z_i(k) \|, \text{for } i = 1, 2, \ldots K, i \neq j; \]

3. Compute \( Z_j(k+1) = \frac{1}{N_j} \sum_{X \in S_j(k)} X \);

\[ J_j = \sum_{X \in S_j(k)} \| X - Z_j(k+1) \|^2, \text{here } N_j \]

is the members 's number of the set \( S_j(k) \);
4. If \( Z_j(k+1) = Z_j(k) \), for \( j = 1, 2, \ldots K \), stop. Otherwise go to step 2.

Before applying the above algorithm to our application, many problems need to be considered and solved in advance.

1) Initial class center selection
Proper selection of initial cluster centers is important in this algorithm because K-Mean algorithm does not guarantee the global minimum of the objective function, that is, if the initial class center is unsuitable, clustering will be trapped into a local optimum.

It is assumed that there are only eight classes in the image because only eight colors are considered. The RGB value of a pixel near the color blob center is selected for each color as an initial class center. This method seems successful by the experimental results.

2) Color sample collection
Though in the case of real game, there is only one orange ball in use, the ball may move to everywhere on the playing field where the lighting may not be uniform. The reflection and shadow of the objects also affect the colors of different parts on the ball. Therefore, one color sample of the ball will have little influence in the sense of statistics. In order to obtain accurate result, many balls are distributed on the playing ground to collect clustering samples. Similar method is applied to analyze the colors of other objects.

3) The number of classes in clustering
This problem is a key factor for accurate clustering. The K-Mean clustering algorithm can merely decide which class a pixel belongs to, that is, a pixel belongs to one class just because it is nearer to its center than to those of any other’s. This may lead to errors in clustering when there are many salt-pepper noises, because those noisy pixels will form several new classes and they cannot be detected by observation alone.

Two steps are used to determine how many classes should be used in the clustering. First, a median filtering is imposed on the raw image. This step is expected to remove most of the noises. Second, pixels of the filtered image will be shown in a 3D RGB space. Many 3D blobs can be observed easily. The number of these blobs will determine the number of classes.

3.2. Experiment Results and Discussion
The source image is a clipped region within the image of Fig.1.

![Fig.4 Image used for clustering.](image_url)

1) Result of clustering with 8 classes
In this experiment, eight classes is used in clustering since only eight colors can be observed in Fig.4. We get $\sum J = 7420484$.

From the above table, we get $\sum J = 6112986$.

Comparing $\sum J$ of 10 classes and 8 classes, we find the result of color clustering with 10 classes is obviously better than that with 8 classes.

<table>
<thead>
<tr>
<th>Color</th>
<th>RGB</th>
<th>Total Num</th>
<th>J</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>253.9</td>
<td>1611</td>
<td>147101.0</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>255.0</td>
<td></td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>252.6</td>
<td></td>
<td>7.7</td>
<td></td>
</tr>
<tr>
<td>Yellow</td>
<td>244.7</td>
<td>338</td>
<td>327644</td>
<td>16.5</td>
</tr>
<tr>
<td></td>
<td>252.5</td>
<td></td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>167.2</td>
<td></td>
<td>25.6</td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td>65.9</td>
<td>30153</td>
<td>2551935.5</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>124.5</td>
<td></td>
<td>6.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50.0</td>
<td></td>
<td>3.8</td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>107.9</td>
<td>244</td>
<td>538782.1</td>
<td>22.7</td>
</tr>
<tr>
<td></td>
<td>174.4</td>
<td></td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>131.3</td>
<td></td>
<td>31.0</td>
<td></td>
</tr>
<tr>
<td>Orange</td>
<td>235.8</td>
<td>357</td>
<td>632428.1</td>
<td>25.4</td>
</tr>
<tr>
<td></td>
<td>147.4</td>
<td></td>
<td>28.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>62.4</td>
<td></td>
<td>17.5</td>
<td></td>
</tr>
<tr>
<td>Black</td>
<td>62.1</td>
<td>921</td>
<td>407620.7</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>68.7</td>
<td></td>
<td>15.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>39.2</td>
<td></td>
<td>8.6</td>
<td></td>
</tr>
<tr>
<td>Magenta</td>
<td>253.2</td>
<td>170</td>
<td>200766.2</td>
<td>8.5</td>
</tr>
<tr>
<td></td>
<td>165.1</td>
<td></td>
<td>19.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>153.6</td>
<td></td>
<td>26.9</td>
<td></td>
</tr>
<tr>
<td>Cyan</td>
<td>176.9</td>
<td>159</td>
<td>177032.4</td>
<td>24.9</td>
</tr>
<tr>
<td></td>
<td>250.9</td>
<td></td>
<td>10.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>241.8</td>
<td></td>
<td>19.8</td>
<td></td>
</tr>
<tr>
<td>Shadow1</td>
<td>123.0</td>
<td>520</td>
<td>851394.9</td>
<td>24.8</td>
</tr>
<tr>
<td></td>
<td>138.7</td>
<td></td>
<td>26.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64.9</td>
<td></td>
<td>18.6</td>
<td></td>
</tr>
<tr>
<td>Shadow2</td>
<td>180.5</td>
<td>207</td>
<td>278280.6</td>
<td>23.5</td>
</tr>
<tr>
<td></td>
<td>221.6</td>
<td></td>
<td>14.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>130.2</td>
<td></td>
<td>24.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Result of K-Mean clustering with 10 classes.

Note: Class Shadow1 may be the pixels around magenta orientation mark and orange ball. Class Shadow2 may be the line between the playing field white border and its green surface.

Some facts in Fig. 6 need to be explained further in order to understand the points described in Section 3.1.

i) A black patch can be seen in the upper right corner on the playing ground. It is the result of the shadow and uneven distribution of lighting.

ii) There are white spots in cyan markers. The reason is that the RGB values of some pixels within the cyan markers are really close to white, that is, the color of cyan patches on the playing ground is actually classified into two clusters.

iii) The centers of the blue patches seem a bit green. The reason is similar to ii): “Blue” is close to “Green” under the yellow lighting condition in our experiment.

Fig. 7 Class centers in color clustering with 10 classes

Fig. 7 shows the ten class centers (red ‘Θ’s) in color clustering. It is so different from the ideal case in Fig. 3. The color of lighting adds an offset to each of the ideal color; shadows and reflections breed new color classes.
classes since the $\sum J$. It is proved that differentiating colors by human observation is unreliable.

4. Camera Calibration

As mentioned before, there is barrel distortion in the captured image by the Camera. To compensate the calibration, we adopt Tsai’s camera model to obtain camera parameters. The following section is a brief description of Tsai’s algorithm.

4.1. Camera Calibration Algorithm by Tsai’s Model

Because the application of this model is a relative standard process, this section only summarizes the procedure of computation in Tsai’s model. Some mathematical signs are defined as:

- $(x_w, y_w, z_w)$ - The 3D coordinate of the object point $P$ in the 3D world coordinate system;
- $(x, y, z)$ - The 3D coordinate of the object point $P$ in the 3D camera coordinate system, which centered at point of optical center;
- $R, T$ - The transformation matrices from the world coordinate to the camera coordinate;
- $f$ - The distance between the image plane and the optical center;
- $k_1, k_2$ - The radial lens distortion parameters;
- $S_x$ - The uncertainty image scale factor.

First, the following four steps are taken [2][5]:

1. Rigid body transformation from $(x_w, y_w, z_w)$ to $(x, y, z)$. Parameters to be calibrated: $R, T$. This step translates the 3D world coordinates into 3D camera coordinates and computes the six extrinsic parameters of the camera model.
2. Perspective projection with pin-hole geometry. Parameter to be calibrated: $f$. The output of this step is the ideal undistorted image coordinates.
3. Radial lens distortion parameters to be calibrated: $k_1, k_2$. These parameters compensate for barrel distortion of camera, which make the field edge rounded. We can see this effect in the Fig. 1.
4. TV scanning, sampling, computer acquisition. Parameter to be calibrated: uncertainty scale factor $S_x$ for image’s $X$ coordinate. The last three steps compute five intrinsic parameters of the camera model (focal length, lens distortion, scale factor for the rows, and the origin in the image plane).

4.2. Acquire Matching Points

The Tsai’s approach requires a lot of, say 50, sample points for computation. Their world coordinates and the coordinates of their matching points in the image should be known first. It is generally a time-consuming task. To complete this work autonomously, a grid pattern as shown in Fig. 8 is designed and applied to finding the coordinates of the sample points in the image by blob analysis.

![Calibration Pattern](image)

The pattern cloth consists of 6 rows, 8 columns evenly placed squares, with each square’s dimension equals 10cm. The distance between 2 nearest squares in the $x$ direction is 22 cm., and is 15cm. in the $y$ direction.

At first we intend to do the matching work by corner detection as in [6], but the minimum precision by using the corner detection is 1 pixel, which is fairly large, and the corner results are not very satisfactory due to the shape of the blobs in the captured image. And there are some noise.

Compared with corner detection, blob analysis has no such shortcomings. The 48 centers of the squares in the pattern are selected as the samples of the camera calibration. The 8-connected-neighbour is used in the blob analysis, and parameters such as compactness and area are used to filter out some unwanted results. The result of the blob analysis contains the gravities of the blobs in the image. By this way, we get the image coordinates.

The world coordinates of the centers of the squares can be easily achieved. In the case, bilinear interpolation is used to compute all the other centers of the squares. Only the world coordinates of the 4 corner points are measured with $z_w = 0$.

The remaining work is to match the world coordinates with the image coordinates, i.e. to
match the results of the blob analysis, with the results of the bilinear interpolation. It is not so easy to directly match the points. We found an indirect way. Because the squares in the patterns cloth we designed are sparsely displaced, so you can imagine the distance between the centers of the gravity is fairly large. On the basis of this feature, we do the matching work in two steps: 1) Roughly measure the coordinates of the centers of the four corner squares in the image, then get the remaining center’s coordinate by the bilinear interpolation. 2) Match the rough results with the precise result by the blob analysis by distance. The Fig. 9 illustrates the matching algorithm.

Fig. 9 Illustration of the matching algorithm

Now the rough centers and the precise gravities of the blobs are matched, and because the order of the rough results is just the same as the one of the world coordinates, the sorted pairs of the world coordinates and the image coordinates are achieved.

4.3. Result and Evaluation

We do the Tsai’s Model camera calibration with full optimization by the matching data. Because we use subsample technique, the focal length of the camera is different from the original focal length.

The following are the evaluation data by the full optimization Tsai’ Camera Calibration,

- distorted image plane error:
  \( \text{mean} = 0.317313, \text{stdev} = 0.172131, \text{max} = 0.731821 \text{ [pix]}, \text{sse} = 6.225558 \text{ [pix}^2]\); 
- undistorted image plane error:
  \( \text{mean} = 0.332143, \text{stdev} = 0.179263, \text{max} = 0.754970 \text{ [pix]}, \text{sse} = 6.805672 \text{ [pix}^2]\); 
- object space error:
  \( \text{mean} = 3.198416, \text{stdev} = 1.727718, \text{max} = 7.291396 \text{ [mm]}, \text{sse} = 631.328942 \text{ [mm}^2]\); 
- normalized calibration error: 0.813582.

Compared with the result in [6], it is better. And while using the parameters achieved, we get steady states data of the robots and the ball.

5. Conclusion and Discussion

The calibration is an indispensable process in our vision system. It enables the success of color-based image segmentation and accurate blob positioning. “manual” calibration, such as differentiating colors by observation is rather unreliable. We presented a practical way to realize the color calibration and the camera calibration autonomously within about one hour.

This paper’s contribution in camera calibration is a good pattern for calibration work, and provide an easy way to match the points in the world coordinate with the image points.

In this paper, we have implemented a K-Mean-clustering-based color calibration algorithm. Three conditions in color calibration, which are initial class center selection, color sample collection, the number of classes in clustering, are discussed in detail. Experiment results have shown the effectiveness of our approach.

6. Reference