Plug & Play: Fast Automatic Geometry and Color Calibration for Cameras Tracking Mobile Robots

Alexander Gloye, Anna Egorova, Mark Simon, Fabian Wiesel, and Raúl Rojas

Freie Universität Berlin, Takustraße 9, 14195 Berlin, Germany
http://www.fu-fighters.de

Abstract. We have developed an automatic calibration method for a global camera hanging from the ceiling and tracking autonomous mobile robots. Firstly, we show how to define automatically the color maps we use for tracking the robots' markers. The color maps store the parameters of each important color in a grid superimposed virtually on the field. Secondly, we show that the geometric distortion of the camera can be computed automatically by finding the white lines on the field. The necessary geometric correction is adapted iteratively until the white lines in the image fit the white lines in the model.
Our method simplifies and speeds up significantly the whole setup process at RoboCup competitions. We will use these techniques in RoboCup 2004.

1 Introduction

Tracking colored objects is an important industrial application and is used in the RoboCup small-size league for locating robots using a video camera which captures the field from above.

The two most important problems which arise in this kind of object tracking are: a) the elimination of the geometric distortion of the cameras, and b) the calibration of the colors to be tracked. It is not possible to calibrate geometry and colors manually, once and for all, since the lighting conditions, the height and orientation of the cameras, the colors of the markers, and the field itself change from one place to another, and even from one hour to the next (for example, if daylight is present).

Color maps are needed, because the appearance of a color in RGB space is not uniform across the field. Some parts of the field can be more illuminated than others; different kinds of lightning can intermix (for example, light from a window and artificial light). Good reactive control of the robots requires the best possible computation of the robots' and ball's positions. In our system, we use RGB and HSV color parameters. Therefore, we need to register their differences over the field for each color used for tracking objects. A single lookup table for the whole field would not give good results under inhomogeneous lightning conditions.

Therefore, the correct calibration of the camera(s) is a very important task before a RoboCup game. With the right calibration, the position of the robots can be computed with small error. A manual setup, however, is time-consuming and error prone.

In this paper we describe the techniques we have developed for fast calibration of the global camera(s) used in the RoboCup small-size league. The paper is organized as follows. First we comment on some previously studied methods of autocalibration, for geometry and colors. Then we describe in detail our new semi-automatic color calibration method and compare its results to the hand-optimized ones. The next section deals with the calibration of the geometric transformation of the field and compare the automatic with the manual method. Finally, we describe our future plans towards a fully automated camera setup for our small-size team.

2 Related Work and Motivation

Much effort has been spent in the optics and robotics community to come up with reliable calibration methods for digital cameras. The main problem regarding color calibration, is the great variance of colors under different lightning conditions.

Zrimec and Wyatt have applied machine learning methods to the color calibration problem [10]. Their approach is to recognize regions delimited by edges and classify them according to certain features, such as average color hue, saturation, intensity, and others. A computer vision module for Sony robots uses those features to locate field landmarks. In the small-size league, the only partially constant feature we have is the size of the color markers, so such an approach is not feasible.

Another approach for automatic calibration is to compute global histograms of images, under different lightning conditions. Lookup tables for color segmentation are initialized in such a way as to make the new histogram equal to that found under controlled conditions [6]. In our case this approach would not work, since we do not use a single lookup table for the whole field. We recognize colors locally, using a color map which can change from one part of the field to the other. In general, any method which uses only one lookup-table for color segmentation will not perform as well as having color maps superimposed on the field [7]. Some authors have tried decision trees in order to segment colors independently from light. However, they focus on object localization robustness and do not deal with the automation of the calibration [1].

Regarding the correction of the geometric distortion introduced by a camera, the canonical approach relies on determining first intrinsic camera parameters (such as the distortion of the lens) and then extrinsic parameters (the projective distortion due to the 3D orientation of the camera [2]). The intrinsic parameters can be measured in the laboratory. The extrinsic can be fitted by least squares by identifying points on the image with points in a geometric non-distorted model. In actual RoboCup competitions zoom lenses are used to capture the field, since

the height at which the camera hangs can be variable. It is difficult to compute the intrinsic parameters for all zoom settings, and more so for wide angle lenses. Also, we do not want to identify points on the field by clicking on them (we used to do this before); the software should be able to automatically recognize the orientation of the camera and to select relevant points for matching them with the model. Our approach is therefore much more general and does not require the user to select calibration points by hand.

A paper by Whitehead and Roth describes an evolutionary optimization approach to camera autocalibration [9]. Their method does not apply to our case, because they optimize the fundamental calibration matrix directly, without considering the radial camera distortion. Projective geometry methods alone solve one part of our problem, but not the whole problem. Interestingly, the issue of self-calibration of cameras is very important in photogrammetry because satellite cameras can suffer mechanical misalignment during take-off which has to be corrected automatically [5]. Some authors have studied real-time distortion correction for digital cameras, but without handling the projective transformation correction [4].

3 Semi Automatic Color Calibration

In a previous paper we described our use of color maps for the implementation of robust color recognition of the robot's markers [7]. The color maps we use consist of a virtual grid superimposed on the field, one grid for each important color. At each grid node we store the RGB parameters of the color and the size of visible color markers for that part of the field, as seen by the camera. For each color, for example for red, we store around 100 different RGB measurements on the field, one measurement for each node of the grid.

During a game, we interpolate the RGB information contained in the nodes of the color maps in order to obtain a continuous function of RGB color and marker size over the whole field. The grid must be initialized before starting to track objects, that is, we need an initialization step for each individual color. The color and marker size maps are further adapted during play, so that the tracking system updates the grid information whenever a change of illumination occurs.

3.1 Initializing the Color Maps

The global camera captures images in RGB format. As it is well-known, this color space is not very useful when trying to achieve color constancy. Therefore, we operate in the HSV (hue/saturation/intensity value) color space. The transformation from RGB to HSV, and vice versa, can be computed easily. Fig. 1 shows the variability of the HSV components over the field as captured in our lab.

Fig. 2 shows the functional relationship between the intensity of the background pixels (green) and the color blobs on the field. The relationship between

Fig. 1. The HSV components for the field of play with blue test blobs on it. The first image represents the hue component, the second the saturation, and the third the intensity. As can be seen, the intensity changes significantly throughout the field.

them is an affine function, whose parameters can be computed from the background intensity and from two probes of the desired color positioned on the field. The function has two parameters: an additive constant and a scale factor. We compute both parameters twice, from two probes, and average them for subsequent computations. More probes could also be taken, resulting in better averaging.

Given b_1 and b_2, the intensity of the background color at the point P_1 and P_2, and given c_1 and c_2, the intensities of the color C whose map we want to derive at the points P_1 and P_2, respectively, the translation factor t is given by

$$t = \frac{(c_1 - b_1) + (c_2 - b_2)}{2} \tag{1}$$

and the scale factor s is given by

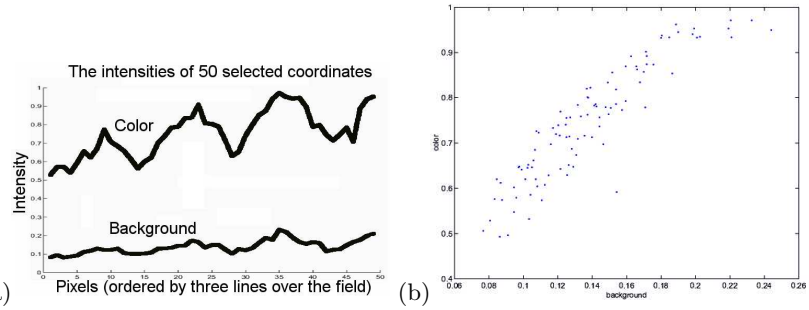
$$s = \frac{(\frac{c_1}{b_1}) + (\frac{c_2}{b_2})}{2} \tag{2}$$

When the background intensity b at a point P is known, the unknown intensity of the color C is given by

$$c = bs + (b + t) \tag{3}$$

The curves in Fig. 3 exhibit the same kind of relationship as in Fig. 2, but now using the results of the automatic computation of the intensity. One can see clearly the good approximation of the real values, even more since the values are not filtered in this figures.

We also want to estimate the size of the color blobs around the field. The apparent size of the markers varies with changing lightning conditions and with camera distortion. Our vision system uses this apparent marker size to improve the tracking of the robots. In this case, we cannot use the background information, because there is no information there about marker size. Therefore, we are limited to use only the color blobs' probes. We assume that their size is proportional to the intensity of the blob. We estimate the marker size by interpolating the sizes of the probes, according to the intensity.



(a) (b)

Fig. 2. The graph compares the intensity values of some random points on the green background and the color map. Illumination inhomogeneities produce different changes for different colors – scaling and translation factors are present: (a) shows the dependancy of the intensities as a function of the field coordinates (three lines over the field); (b) shows the intensity of the color blobs as a function of the intensity of the background at the same points. The correlation is clearly visible.

Fig. 3. The results from applying translation and scaling, computed for a sample blue blob. One can see, that using both scaling and translation yields a better approximation to the real, but unknown, intensities.

To reduce the influence of noise, a median color is computed for the blob and for the background color at the picked point. The radius for the area around the blob is predefined with respect to the blob-object which is identified by this color (for example the orange ball is a little bit smaller than a team marker). The size of the median filter for the background color, for its color grid, is exactly the size of a grid tile.

Fig. 4 shows the automatically computed orange color map for the ball. See Subsection 3.2 for an analysis of the results.

(a) (b) (c)

Fig. 4. A visual comparison of three different color maps. Subfigure (a) shows uniform color map, (b) shows a hand-optimized color map, and (c) shows the automatically initiliazed one.

3.2 Results and Comparison

The results of applying the automatic color calibration for the ball is shown in Fig. 4. It provides a visual feedback of the results and one can see how the sizes and colors at the different nodes of the color grid appear. Table 1 summaries our results. The different forms of initializing the color map before starting a game are compared to the manual adjustment method. As one can see, the automatic initialization gives better results and produces smaller errors, compared to a uniform initialization of the color map. The most important improvement is the reduction of the maximum error, the most relevant magnitude when trying not to lose track of an object in the field. The improvement in estimation of the ball size, as seen from the camera at different coordinates on the field, is also significant.

The obtained results are highly dependant on the choice of the two calibrating probes. One should try to choose points on the field with different light conditions, because this approximates best the affine transformation looked for.

Of course, the automatically initialized map is not as good as one optimized manually during ten or more minutes, but is a very good starting point for starting to track the robots. Its computation takes two user clicks, that is a few seconds. During the game, the maps are continuously updated and reach a stable state.

		Deviation from hand-optimized	
		maximum	mean
Uniform	hue	1.94%	0.85%
	saturation	7%	1.4%
	intensity	48%	9%
	size	31 pixel	13 pixel
	RGB distance	27.9%	5.2%
	HSV distance	48%	8.7%
Automatic	hue	1.28%	0.61%
	saturation	4.6%	1.7%
	intensity	25%	9.4%
	size	18 pixel	9.15 pixel
	RGB distance	14.5%	5.7%
	HSV distance	25.4%	9.73%

Table 1. Statistical results comparing initializations of color maps. The table compares the performance (relative to the hand-optimized color map) of a uniformly initialized color map and two different automatically initialized maps. The performance of one specific map is measured with the maximum and mean percentage deviation of computed values from those in the hand-optimized map.

4 Automatic Geometric Calibration

The second camera setup problem we consider here is the correction of the geometric distortion. Traditional methods for calibrating a camera require identifying pairs of points in the image and model space. The geometric transformation can be computed by finding a mapping with minimal error from image to model space. The quality of the transformation depends on the number of calibration points, their distribution over the field, and the precision of their localization. However, the small-size field provides only a few candidate points for such a match (corners of the field, goal, defense area, and intersections between lines), and they are located mainly at the boundary of the field. To avoid this problem, one could artificially define the necessary calibration points on the field by using a calibration carpet with marks. This allows an accurate calibration of the camera, but requires too much manual intervention.

In this paper we show how to use the white lines on the field to determine the parameters of the transformation. The correlation between the extracted, and transformed lines, and the lines of the model, is a measure for the quality of the transformation. This measure enables us to find the best transformation with conventional optimization algorithms.

Our method is divided in the following steps: First, we extract the contours of field regions from the video image. Next, we find a simple initialization, which roughly matches the lines of the field to the model. We then eliminate all contours which deviate too much from the model; this should retain only regions belonging to the field. Finally, we optimize the parameters of the geometric transformation.

4.1 Extraction of Contours

The non-white regions on the field are found by applying our region growing algorithm as described in [3]. The discrimination between white and non-white pixels is based on their relative intensity with respect to the local background. The local background intensity is interpolated from an intensity map. The map consists of several interpolation points uniformly distributed in a grid over the field, as explained in [7]. The value is bilinearly interpolated between nodes of the grid. To determine the background intensity at such an interpolation point, we average the background intensity in a small region around it. The smaller the region, the better the map adapts to local variations in intensity, but becomes more susceptible to disturbances as, for example white lines or robots, which do not belong to the background. To reduce this effect, we try to eliminate the foreground areas. Therefore, we first take a larger region, which is assured to contain a high percentage of background pixels (green field). The average in this region is a rough approximation for the background intensity. Relative to that approximation, we can now reject foreground pixels in the smaller region and locally determine the intensity by averaging the remaining pixels. The contours of the regions found are the borders of the field lines (see Fig. 5(a)).

(a) (b)

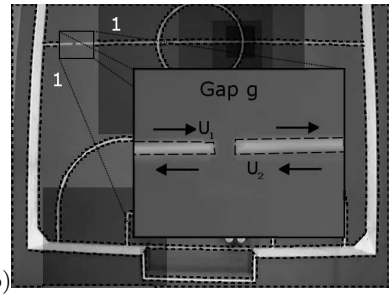


Fig. 5. (a)F-180 Field Padova 2003, with artificial shadows with strong edges. Dashed lines show the field and out-of-field contours. (b) Same field with a broken line (magnification). The gap cause two U-turns in the contour.

4.2 Quality Measure

Assume that we have a hypothetical geometric transformation T and that we want to measure its error. Since we cannot determine the exact location of a contour point p in the model, we approximate its error $E(T, p)$ with the distance between the transformed point $T(p)$ and the nearest white line in the model. For a set of points P, we compute the quality error $\hat{E}$ as

$$\hat{E}(T, P) := \bar{E}(T, P) + \sigma\left(E(T, P)\right) \tag{4}$$

where $\bar{E}$ is the mean error, and σ its standard deviation.

4.3 Initialization of the Transformation

To minimize the complexity of the optimization step, a smart initialization of the transformation T is useful. Furthermore, for the elimination of phantom contours it is even crucial.

We assume that the field has only limited distortion, radially and in the perspective. A linear transformation is a good initial approximation, except for the alignment of the field. Finding a matching pair of rectangles in model and field let us compute such a transformation. We use one camera for each side of the field. Think of each camera image as divided in four quadrants. Taking into account that only one region reaches into all quadrants of one image, a bounding box around its contour will result in a good approximation to a rectangle corresponding to one half of the field in the model. The association between the field sides in image and model is arbitrary and can be fixed by the camera enumeration (or can be detected by the goal box wall color).

In order to find such initial contour C_{init} , we find the bounding box of each contour. Bounding boxes, which do not reach into all four quadrants of the image are immediately rejected. The remaining ones are tested as initialization for the linear transformation, with all four orientations, and are evaluated using the quality measure described above, using the corresponding contour points. Since the region is not symmetric with respect to rotation, the best match identifies the required bounding box and the orientation of the field.

The transformation T can be then initialized accordingly to the linear transformation T_{init}. We use a biquadratic transformation as our standard transformation (see Section 4.5), but other transformation models are possible.

Since the initialization depends on the bounding box of C_{init}, it must be verified that the lines bordering C_{init} are not broken so, that the bounding box covers a larger region. See, for example, Fig. 5(b): Due to a gap, the former regions 1 and 5 (as numbered in Fig. 5(a)) have been fusioned into one region. Its bounding box is not a good approximation for one half of the field anymore. On the other hand, a gap between region 1 and 2 has no effect on the resulting bounding box. Such a gap in a line results in a contour, which has at least one U-turn as U_1 in Fig. 5(b). The gap g can be easily detected, by measuring the change in direction in the sequence of points in the contour as indicated by the arrows. The contour can be extended in the direction of the U-turn by prohibiting the region-grower to grow into the gap. This solves the problems with gaps.

4.4 Elimination of False Contours

Using the initial transformation T_{init} and the quality measure, we can now eliminate large phantom contours which do not have a correspondence in the model. This is necessary for the optimization step (Section 4.5). Phantom contours can come from robots on the field, from objects located outside of the field, etc. Since the robots are mostly black, they belong to the non-white region, except for their markers, which can be very bright, even surpassing the lines in brightness. But

due to their limited size, they can be easily discriminated from the lines on that basis. The main problem are large contours, which might be observed outside the field. The quality measure can be used as a reliable method for discriminating those contours. Depending on the quality of the initial transformation T_{init} and its corresponding contour C_{init}, we reject contours C_j, for which

$$\bar{E}(T_{init}, C_j) > \bar{E}\left(T_{init}, C_{init}\right) + 2\sigma\left(E\left(T_{init}, C_{init}\right)\right). \tag{5}$$

4.5 Optimization

We use for both coordinates in model space a biquadratic interpolation in a single 3×3 grid to determine the coordinates in model-space, giving us 18 parameters to be optimized. The initialization can be analytically derived from T_{init}. However, our approach does only rely on the convergence properties of the transformation function and not on the function itself.

Given the transformation T, gradient descent with linearly falling step-length suffices to optimize the placement of the vertices with sub-pixel accuracy to optimize the transformation T. To achieve faster convergence adaptive selection of step-length and/or conjugate-gradient methods can be used. Gradient descent only requires a non-degenerate initialization with the approximate orientation. In our experiments, even the whole image as rectangle for the initialization converged against the global minimum.

Other transformation-functions can be optimized similarly. Depending on its convergence properties, a global optimization method may be necessary, for example simulated annealing.

4.6 Calibration Results

We applied our algorithm to a set of real video images, captured with two different camera-systems (Basler 301fc, Sony XC-555). We presented the algorithm with two different fields, the field of the Padova World Cup 2003, built according to the F-180 2003 rules, and our lab-field, which has slightly other dimensions and several rough inaccuracies. The images had different brightness, and some were artificially modified to simulate shadows with strong edges (as can be seen in Fig. 5), which normally are not present on the field and strain region-extraction more than the soft shadows usually observed. Furthermore, some images were rotated up to 20^o.

For a correct field model, our algorithm could adjust the geometric biquadratic transformation for all images presented, given the constraints on the distortion, without further adjustments to the algorithm after the initial selection of the Padova images. Subsequent attempts to improve manually the parameters of the transform-function resulted only in worse results, both with respect to the average and standard-deviation of the error.

To speed up the algorithm, initially only a subset of the contour points is taken into account for the quality measure. The amount is gradually increased

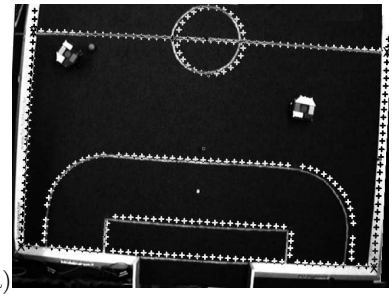
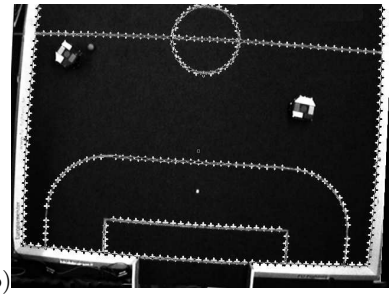
(a)  (b) 

Fig. 6. The model matched on the rotated field (a) after the initialization and (b) after the optimization step.

as the error decreases. We use an adaptive step-width, which slowly increases, unless the gradient step increases the error. Then the step-width is reduced. The step-width is independent of the norm of the gradient.

The implementation was not optimized for performance, so an improvement of several degrees can be expected. For instance, we computed the numerical gradient instead of using the analytical one.

We started the optimization with a step-width of 1 pixel and only using every 20th contour point in the measure. The optimization step required on our set of images at most 6 seconds (5 seconds with conjugated gradient) to adopt the parameters in 1/100 pixel accuracy and in average 4 seconds (4 seconds) on an Athlon XP -2400+ (2GHz).

5 Future Work

The color and geometric calibration methods described in this paper are being used now routinely by our team. They reduce the setup time for a competition to a minimum, and they also allow to recalibrate the system whenever the illumination changes. The color maps are adjusted continuously during a game, so that if the light is slowly dimmed, the robot tracking system is not affected.

The color map for a specific color is initialized by clicking on two markers during setup and computing the color and the size of this marker at any other coordinates on the field. This takes a few seconds, for all colors used, but we would like to avoid even this small manual intervention. In the future, our vision system will automatically detect colors differing from the background, and will initialize a color map for them.

For the geometric calibration, we want to evaluate other transformation-functions, which are probably more accurate, due to their better modeling of the distortion.

Since our region-growing algorithm can track regions in real time, our system could also continually check the camera calibration. Assuming that only slight displacements of the field occur, the algorithm does not need to be reinitialized,

since the current transformation is already a good initialization. The time required to adjust this initialization should be considerably shorter. If the field carpet moves below the camera or if the camera bar is displaced, the system would recalibrate automatically. Is also useful for the current setup method, to use the data over several frames to detect lines which are covered for a short time, for example by moving robots.

The improvements reported in this paper have the objective of achieving true "plug & play" capability. In the future it should be possible just to place robots on the field and start playing immediately against another team, whose colors and marker orientation will be determined automatically. This would speed RoboCup competitions significantly.

References

1. Brusey, J., and Padgham, L., "Techniques for Obtaining Robust, Real-Time, Colour-Based Vision for Robotics", *Proceedings IJCAI'99 - International Joint Conference on Artificial Intelligence, The Third International Workshop on RoboCup – Stockholm*, 1999.
2. Forsyth, D. A., and Ponce, A., *Computer Vision: A Modern Approach*, Prentice Hall., 1st edition, 2002.
3. von Hundelshausen, F., and Rojas, R., "Tracking Regions", in D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.): *RoboCup-2003 - Robot Soccer World Cup VII*, Springer-Verlag, 2004.
4. Gribbon, K. T., Johnston, C.T., Bailey, D.G., "A Real-time FPGA Implementation of a Barrel Distortion Correction Algorithm with Bilinear Interpolation", *Image and Vision Computing*, Palmerston North, New Zealand, pp. 408-413, November 26-28, 2003.
5. Jacobsen, "Geometric Calibration of Space Remote Sensing Cameras for Efficient Processing", *International Archives of Photogrammetry and Remote Sensing*, Vol.32, Part I, pp. 33-43.
6. Kulessa, T., and Hoch, M., "Efficient Color Segmentation under Varying Illumination Conditions", *Proceedings of the 10th IEEE Image and Multidimensional Digital Signal Processing Workshop*, July 12-16, 1998.
7. Simon, M., Behnke, S., Rojas, R.: "Robust Real Time Color Tracking" In: Stone, P., Balch, T., Kraetszchmar (eds): *RoboCup-2000: Robot Soccer World Cup IV*, pp. 239-248, Springer, 2001.
8. Rojas, R., Behnke, S., Liers, A., Knipping, L.: "FU-Fighters 2001 (Global Vision)", In: Birk, A., Coradeschi, S., Tadokoro, S. (eds): *RoboCup-01: Robot Soccer World Cup V*, Springer, 2001.
9. Whitehead, A., and Roth, G., "Evolutionary Based Autocalibration from the Fundamental Matrix", in S. Cagnoni, Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, Gnther R. Raidl (Eds.), *Applications of Evolutionary Computing – EvoWorkshops 2002, EvoCOP, EvoIASP, EvoSTIM/EvoPLAN*, Kinsale, Ireland, April 3-4, Springer-Verlag, 2002.
10. Zrimec, T., and Wyatt, A., "Learning to Recognize Objects - Toward Automatic Calibration of Color Vision for Sony Robots", *Workshop of the Nineteenth International Conference on Machine Learning (ICML-2002)*.