URN (Paper): urn:nbn:de:gbv:ilm1-2011iwk-104:8

56TH INTERNATIONAL SCIENTIFIC COLLOQUIUM
Ilmenau University of Technology, 12 – 16 September 2011
URN: urn:nbn:gbv:ilm1-2011iwk:5

APPLICATION OF IMAGE PROCESSING FOR AUTONOMOUS ROBOTS

Christoph Ußfeller

Technische Universität Ilmenau

ABSTRACT

For the task of object localisation for autonomous robots, an image processing system is applied. The essentials image decoding, color classification and image segmentation are covered. The Helmert transform is applied for robot localisation by matching image segments with a pattern model. The transform is adjusted to utilize additional data gathered by the image processing. A comparison between the original and the modified transform is done.

1. INTRODUCTION

In the field of autonomous robotics there exists the benchmark task of soccer playing robots. Due to the broad range of problems to be solved in order to become competitive against human players, there are specific leagues which deliver standardised general frameworks to focus on specific problematic issues. In the RoboCup small size league there compete two centralised controlled teams utilising an object tracking top vision system. This paper deals with the localisation problem.

One branch of localisation methods is the model based statistical evaluation of image data. Representatives of this branch are correlation analysis, performed in the spatial or in the frequency domain and the Hough transform [BBD+09]. Both methods find a maximum consensus of the input data within a predetermined, discrete solution space. The algorithm RANSAC [FB81] is related to the Hough transform in the way it determines the greatest consensus of input data with a model, but it saves computing time by generating the evaluated solutions out of the input data. While Hough transform and RANSAC are usually applied to binary data, correlation analysis also enables access to texture information. However, this statistical methods while being able to extract information from strongly degenerated signals, are quite computational expensive. A second branch of localisation methods is span by segmentation methods which decompose the image into regions of equal or similar manifestation of a feature. Subsequent this reagions are matched to a model. Segmentation methods may be reagion-oriented [BBV00, Ler08], contour-oriented [Ren] or take place in the frequency domain [MTJAL05, LW95]. Another perspective to the segmentation problem may be seen in the utilisation of cluster techniques [YGZ03, YSL09], which closely relate to reagion oriented approaches. As the biggest problem in the actual task is the required computing time, statistical methods are refrained. From the group of segmenting methods, the region oriented variant of run length encoding and subsequent interline union of connected runs was chosen, because the segmentation task itself as the implementation is comparatively simple and the expected computational cost is low. This approach has been proposed in [BBV00]. It is shown that this method is a specialised variant of the task underlying DBSCAN [M. 96], which serves as example for cluster-based image segmentation methods. In contrast to [BBV00], where a double thresholding color discriminating method was feasible due to the availability of a YUV-signal, a Voronoi classificator is used as adaption for RGB-signals.

The first step of the processing chain (Figure 1) includes bayer demosaicing to interpolate the color components for each pixel, localised amplification to level differences of illumination, color classification as a problem specific color reduction which matches the color space to expected colors and a run length encoding which presents the obtained data in a fast proccessable way to the next stage. In a second step, equal colored segments are formed. After matching corresponding segments to the robot patterns, the task of interpretation of segment properties as robot postitions arises. Therefore the regression method Helmert transformation is adapted for the context of image processing to make use of the redundancy of the robot pattern. The parameters of the transform allow easy calculation of the robot's location.

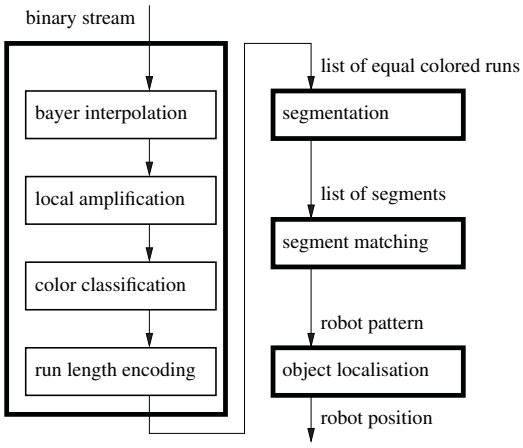


Fig. 1. Processing chain

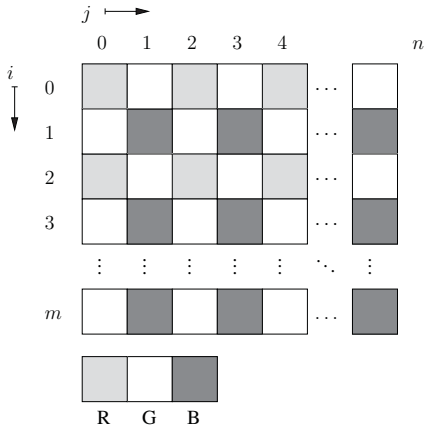


Fig. 2. Bayer pattern

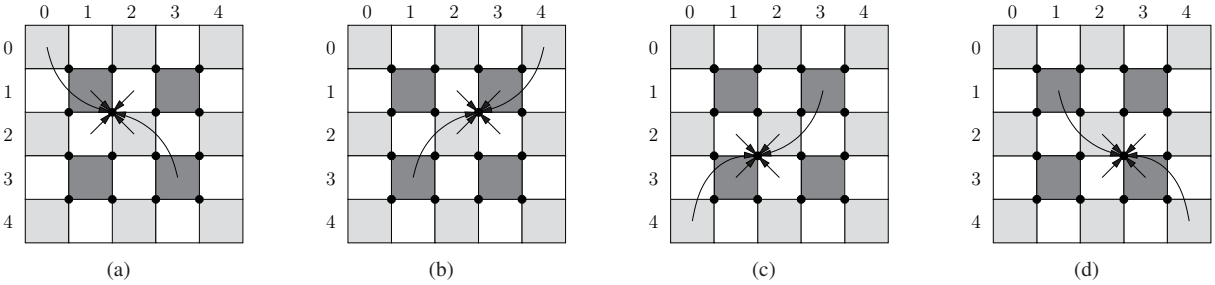


Fig. 3. Bayer interpolation

2. PREPROCESSING

2.1. Bayer interpolation

A common variant of digital imaging sensors is the 1-chip sensor with bayer pattern. It is a photosensitive matrix covered by a regular pattern of color filters, see Figure 2. The values of this matrix may be obtained as $V_{i,j} \in \mathfrak{X} \subset \mathbb{R}$ with row index i and column index j. The advantage of relatively inexpensive video hardware is countered by the drawback of incomplete color information. As every pixel detects the intensity only for one color channel, a reconstruction of the missing channels becomes necessary. There is a broad range of interpolation techniques for this problem, which usually trade-off between accuracy and processing time. See [GGA+05] for an instructive introduction. In the application at hand the dominating criterion is processing speed, thus a very simple demosaicing regime is implemented, see Figure 3.

A bayer primitice cell consists of one red-, one blue- and two green-sensitive elements which share one vertex. Let $\vec{v} := \begin{pmatrix} v_1 & \dots & v_n \end{pmatrix}^T \in \mathfrak{X}^n$ denominate the value of an image element of the channels $1, \dots, n$ and the corresponding values $v_1, \dots, v_n$ in the device dependent color space $\mathfrak{X}^n$, then a bayer decoding scheme $\mathrm{decode} : \mathfrak{X}^{N \times M} \times \mathbb{N}^2 \to \mathfrak{X}^3 : (V, i, j) \mapsto \mathrm{decode}(V, i, j)$ is desired, where for convenience $v_r := v_1$, $v_g := v_2$ and $v_b := v_3$. In the implemented regime the green value of a cell is the mean of both green-sensitive elements $g_1(i,j)$, $g_2(i,j)$ to $v_g(V,i,j) = (V_{g_1(i,j)} + V_{g_2(i,j)})/2$. The red and blue values are determined as weighted mean according to $v_r(V,i,j) = (3V_{r_1(i,j)} + V_{r_2(i,j)})/4$ and $v_b(V,i,j) = (3V_{b_1(i,j)} + V_{b_2(i,j)})/4$. Thereby $r_1(i,j)$ and $b_1(i,j)$ point to the respective dominant elements which are part of the bayer cell. Further $r_2(i,j)$ and $b_2(i,j)$ point to the respective recessive elements which are the first elements of the respective color found by mirroring the dominant cell on the center vertex. Thus the values v_r, v_g and v_b are reconstructed for the vertex that is center of the bayer cell. As the underlying pattern is regular, initialisation of an odd and an even indexing group $(r_1(i,j), r_2(i,j), g_1(i,j), g_2(i,j), b_1(i,j), b_2(i,j))_{o,e}$ at the beginning of a line leads to a scheme, where subsequent to image elements evaluation all indices are incremented by 2. This regime yields less color distortion at sharp edges than the common nearest neighbour approach but appears to be relatively blurry. Yet it is sufficient for the robot localisation in RoboCup as it is fast enough and does not affect the subsequent color classification perceptibly.

2.2. Localised amplification

Due to the big influence of vignetting and local disparities of illumination, the color classification initially was insufficient. A good recognised robot disappeared when it moved to a corner of the pitch. As this problem could not be solved adequately by adaption of the color classificator, a local amplification was put into the processing chain. Figure 4 shows the mean intensity of the empty pitch with a severe sag in the pitchs corners. By inversion of those local means, a localised amplification table is constructed. It is applied to all bayer decoded elements prior to color classification. As it is a pixel operation, it is computationally expensive and leads to an increase of the image processing time by factor 1.4-1.5. Yet the achieved frame rate is sufficient and color classification becomes reliable.

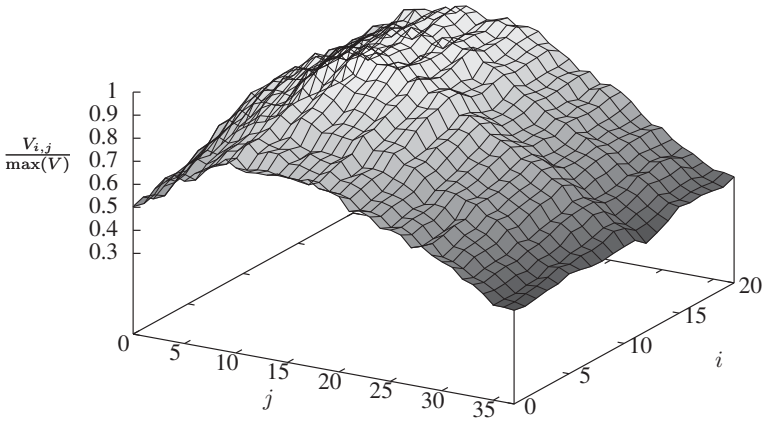


Fig. 4. Local itensity of the pitch's illumniation: Mean illumination of a cell per maximum of the mean illumination of all cells.

2.3. Color classification

Robot localisation by identifying and evaluation of marker patterns is the objective for image segmentation. A marker features a distinct color of a set $\mathfrak{C}$. In the domain of RoboCup, the color set is given by $\mathfrak{C} := \{\text{White}, \text{Orange}, \text{Yellow}, \text{Green}, \text{Cyan}, \text{Blue}, \text{Magenta}, \text{Residue}\}$, where the sets elements are a human description for an impression consent. However the element Residue is reserved for any $\vec{v}$ of indeterminate class membership. A sample $S_C \subset \mathfrak{X}^n$ for each color class $C \in \mathfrak{C}$ is used to determine a classificator as function $c : \mathfrak{X}^n \to \mathfrak{C}$. The samples are gathered and labeled externally. Figure 5 gives an exemplary scatter plot of those collections, where in feasible cases each collection forms a discriminable cluster. There are many different approaches for the classification problem, e.g. simple double thresholding [BBV00], support vector machines [Wan05], self organizing maps [Koh90, ZH09] and multilayer perceptrons [PBC05]. Due to the small and discrete feature space, the approach of a discrete Voronoi decomposition as sample generalisation is chosen. For this purpose a metric $d : \mathfrak{X}^n \times \mathfrak{C} \to \mathbb{R}_+$ is used to determine the distance of one element $\vec{v}$ of the RGB color space to a respective color class C. With this the color space is discretely decomposed into Voronoi cells $\mathcal{V}_C = \{\vec{v} \in \mathfrak{X}^n \mid d(\vec{v}, C) < d(\vec{v}, i);\ i, C \in \mathfrak{C} \wedge i \neq C\}$ (confer [Sud06][1]) which is equivalent to a generalisation to

$$\begin{aligned} G &: \mathfrak{X}^n \to \mathfrak{C} \\ \vec{v} &\mapsto G(\vec{v}) = \underset{C \in \mathfrak{C}}{\arg\min}(d(\vec{v}, C)) \ \cdot \end{aligned} \tag{1}$$

and yields an implementation of the desired classification c.

The Mahalanobis distance [Mah36] and a minimum sample distance are examined as considered of metric d. For the application of the Mahalanobis distance, the distribution of a class is considered as a spheroid with the sample based estimation of the center

$$\overrightarrow{S_C} = \frac{\sum\limits_{\vec{v} \in S_C} \vec{v}}{|S_C|} \tag{2}$$

[1] In [Sud06] "$\leq$" is used as affinity criterion. As this leads to ambiguity on the cell borders, "$<$" is used instead.

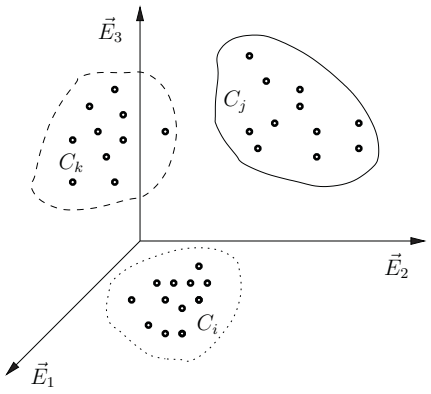


Fig. 5. Exemplary scatter plot of three samples in a $\mathbb{R}^3$ feature space. As samples are easily seperable, a plausible generalisation of the class membership of elements of the feature space seems to be feasible.

and the estimation of the covariance matrix

$$\mathrm{Cov}(S_C) = \frac{\sum_{\vec{v}\in S_C} \begin{pmatrix} \Delta v_1 \Delta v_1 & \dots & \Delta v_1 \Delta v_n \\ \vdots & \ddots & \vdots \\ \Delta v_n \Delta v_1 & \dots & \Delta v_n \Delta v_n \end{pmatrix}}{|S_C| - 1}, \tag{3}$$

where $\vec{\Delta v} := \vec{v} - \overrightarrow{S_C}$. The mahalanobis distance between a $\vec{v}$ and a color class C represented by its sample S_C then is

$$d_m(\vec{v}, C) := \sqrt{\vec{\Delta v}^{\,T} \left(\mathrm{Cov}(S_C)\right)^{-1} \vec{\Delta v}}, \tag{4}$$

see [ITM01]. This approach has been utilised for RoboCup color classification in [Ler08]. [KO05] shows that a classificator for multivariate normal distributions under (1) and (4) serves as Bayes classificator [2].

In the case that the sample S_C of a color class C is not well approximable by a multivariate normal distribution, the second attempt for a definition of a metric is the sample based minimum distance

$$d_S(\vec{v}, C) := \min_{\vec{s}\in S_C} (\mathfrak{d}(\vec{v}, \vec{s})) \tag{5}$$

where $\mathfrak{d} : \mathfrak{X}^n \times \mathfrak{X}^n \to \mathbb{R}_+$ is a problem specific metric, for instance euclidean or manhattan. It may be utilised, when no applicable knowledge about the distributions is available. As this 1-nearest-neighbour method is prone to noise, the color classificator c may also be implemented as a k-nearest-neighbour method. When a Residue sample is gratuitous, in both cases (4), (5) a hint $d_{m,S}(\vec{v}, \mathrm{Residue}) :=$ threshold may be used, where threshold is an application dependent maximum distance for a well recognized color.

In practical tests, both variants have been insufficient without localised amplification. With introduction of localised amplification, both approaches yield sufficient results with insignificant distinctions of classification quality.

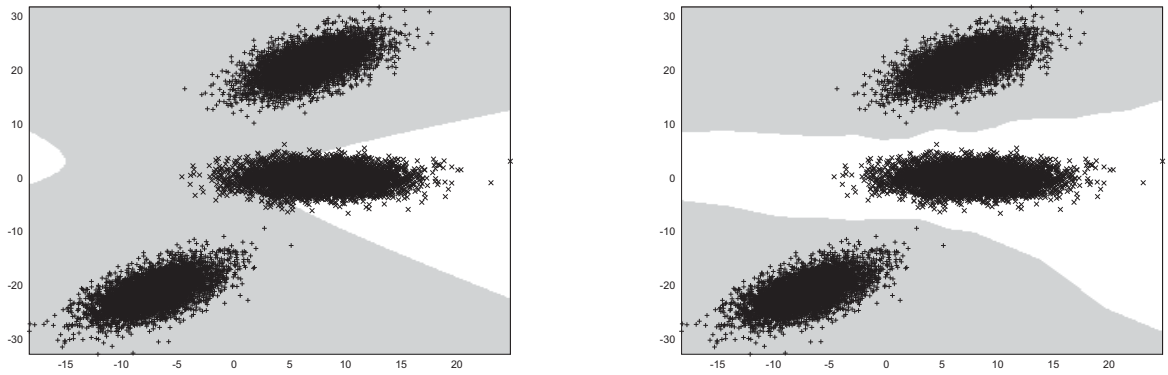
3. COLOR IMAGE SEGMENTATION

3.1. Segment definition

The objective of image segmentation is to identify the image regions with equal manifestation of a feature. This is applicable to the object localisation in the domain of RoboCup which its task is to detect the ball and marker patterns. Combined with prior knowledge (geometric model of the pattern, parameters of the optical system) the determination of the environment's state is allowed.

A segment is understood as a connected set of image elements of equal color. Its elements have no direct neighbours of the same color class, which are not element of the set. This property of a segment suggests to understand a segment as DBSCAN-cluster [M. 96] with respect to Definition 5. Let $\varepsilon \in \mathbb{R}_+$ denote a critical distance, MinPts $\in \mathbb{N}$ a minimal cardinality, $\mathfrak{D}$ a data domain, $D \subset \mathfrak{D}$ a data set and $d : \mathfrak{D}^2 \to \mathbb{R}_+$ a distance function. Definitions 1-5 are taken from [M. 96] with the substitutions $\varepsilon :=$ Eps and $d :=$ dist.

[2] As a color class C may be considered as generator of meassurements $\vec{v_i}$, the distribution of those meassurements is the probability $P_C(\vec{v})$ of C realisating $\vec{v}$, where the distribution itself is unknown but represented by the sample S_C (training set). In a multiclass setup $P_C(\vec{v})$ is refered to as $P(C|\vec{v})$, the probability of $\vec{v}$ under C. Utilisation of Bayes Theorem leads to the probability $P(\vec{v}|C)$ of realisation of $\vec{v}$ being preceded by C, shown in [KO05]. A Bayes classifier is a classifier that chooses $C(\vec{v})$ under $P(\vec{v}|C)$ most plausible, see [KO05].



(a) Mahalanobis distance

(b) Sample based minimum euclidean distance

Fig. 6. Discrete Voronoi decomposition of a plane by two synthetic samples. While the central sample (white, $\times$) is unimodal, the second sample (grey, $+$) is bimodal. Thus the Mahalanobis distance yields misclassification for some points.

Definition 1 *(ε-neighborhood of a point) The ε-neighborhood of a point p, denoted by $N_\varepsilon(p)$, is defined by $N_\varepsilon(p) = \{q \in D \,|\, d(p,q) \leq \varepsilon\}$.*

Definition 2 *(directly density-reachable) A point p is directly density-reachable from a point q wrt. ε, MinPts if*

1. *$p \in N_\varepsilon(q)$ and*
2. *$|N_\varepsilon(q)| \geq$ MinPts (core point condition).*

Definition 3 *(density-reachable) A point p is density-reachable from a point q wrt. ε and MinPts if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.*

Definition 4 *(density-connected) A point p is density-connected to a point q wrt. ε and MinPts if there is a point o such that both, p and q are density-reachable from o wrt. ε and MinPts.*

Definition 5 *(cluster) Let D be a database of points. A cluster C wrt. ε and MinPts is a non-empty subset of D satisfying the following conditions:*

1. *$\forall p, q$: if $p \in C$ and q is density-reachable from p wrt. ε and MinPts, then $q \in C$. (Maximality)*
2. *$\forall p, q \in C$: p is density-connected to q wrt. ε and MinPts. (Connectivity)*

In the case at issue, the input data $D \in \mathfrak{C}^{N \times M}$ is presented as locally referenced two dimensional array of unique class membership of the data domain $\mathfrak{D} = \mathfrak{C}^{N \times M}$. An element $p \in D$ may be described $p := (y, x, c) = (y, x, D_{y,x})$. By fixating the minimum distance to $\varepsilon := 1$, the minimal cardinality of an epsilion neighbourhood to MinPts $:= 1$ and defining d according to

$$d : \mathfrak{D}^2 \to \mathbb{R}$$
$$(p_a, p_b) \mapsto d(p_a, p_b) = \begin{cases} \varepsilon + 1, & \text{if } c_a \neq c_b \\ |x_b - x_a| + |y_b - y_a|, & \text{else} \end{cases} \tag{6}$$

as a modified manhattan metric, a definition of image segments arises:

Definition 6 *Let D denote the set of classified image elements. Under* (6) *all DBSCAN-cluster C_i in D with respect to $\varepsilon = 1$ and MinPts $= 1$ are segments.*

3.2. Run length encoding

This leads to an algorithm for image segmentation as a special case of the task underlying DBSCAN which suffices the demand for high processing speed in the given application. The pursued approach is to build a run length encoding of class membership and subsequent union of neighboured groups of the same class membership which represent the segments, as proposed in [BBV00, MDSG02].

Run length encoding is a technique for image compression. Employed for image segmentation, two beneficial effects arise: The reduction of required memory bandwith and the decrease of redundant tests for connectedness.

While scanning the input data set line by line a set of groups $\mathcal{G} := \{G_i\}$ with

$$G := (y, x, w, c, s, r) \tag{7}$$

is built, where y, x, w, c are a compressed description of the density-connected elements $\{ (y, x, c), (y, x + 1, c), \ldots, (y, x + w, c) \}$. Label s and index of the root group r are required for the generation of segments.

The subtupel n_i with

$$n_i := n(G_i) = (y(G_i), x(G_i), w(G_i), c(G_i), s(G_i)) \tag{8}$$

may be recognised as node and the index of the root group as edge $e_{i,j}$:

$$r(G_i) = e_{i,r(G_i)}. \tag{9}$$

The set of groups $\mathcal{G}$ consequently realizes a graph $(\{n_i\}, \{e_{i,j}\})$, where the elements of $\mathcal{G}$ may be ordered with respect to

$$G_a < G_b \iff \begin{cases} x(G_a) < x(G_b), & \text{if } y(G_a) = y(G_b) \\ y(G_a) < y(G_b), & \text{else.} \end{cases} \tag{10}$$

Algorithm 1 is applied for construction of the set $\mathcal{G}$ by the line. Due to the lack of interline neighbourhood information at this point, the initialisations $s(G_i) \leftarrow i$ and $r(G_i) \leftarrow s(G_i)$ are made. The result therefore is a disjunct forest [BBV00] (confer [Die06]). The subsequent procedure of interline group connection delivers the trees of this forest which are the segments to be found [BBV00]. Their elements encode all picture elements part of the respective segment. As the input data is processed by the line, $L_i \subset \mathcal{G}$ are sustained to

$$L_i := \{G_q \in \mathcal{G} | y(G_q) = i\}, \tag{11}$$

by indexing a lazy garbage collected structure keeping $\mathcal{G} = L_1 \cup \ldots \cup L_N$. All L_i may be ordered with respect to (10).

In an actual implemenatation, Algorithm 1 is modified to take the undecoded image V instead of the already preprocessed D. Thus the assignment $c_c \leftarrow D_{i,1}$ implies full preprocessing integrated into the run length encoding, which significantly reduces index operation costs. Confer Figure 1.

3.3. Union of reachable groups

The basis of the segmentation by run length encoding with respect to Definition 6 is the direct density-reachability between two groups G_a, G_b of different rows $i(G_a) \neq i(G_b)$: With respect to definition 6 two groups G_a, G_b of different rows $|y(G_a) - y(G_b)| = 1$ are direct density-reachable in terms of definition 2, when the conditions $x(G_a) < x(G_b) + w(G_b)$ and $x(G_b) < x(G_a) + w(G_a)$ are met. Algorithm 2 may be used to unite direct density-reachable groups G_a, G_b of equal class membership. It utilises incremental deep update of equivalences by adapting the labels of the respective root groups. Algorithm 3 finds the segments in D by linewise unions of reachable groups. Figure 7 gives an example of this.

3.4. Segment matching

As Algorithm 3 has processed the run length encoding $\{L_i\}$, the segments in D are implicitly determined and may be extracted into a form, where each segment is characterised by an area A_i, a center of gravity $\vec{r_i}$ and a color class C_i. If the RoboCup standard pattern [MAN+11] (see Figure 8(a)) is to be identified, it is sufficient to represent a robot by its center segment, which is identified by dedicated color classes. The remaining pattern segments are then arranged by their distances to the center. If an alternative pattern with better utilisation of the available area (see Figure 8(b)) is to be found, peripheral markers of close robots may become indistinguishable. To avoid this, a circular region of interest is put arround the center segment. For every remaining color class reserved for robot

Input: Picture $D \in \mathfrak{C}^{N \times N}$
Output: Linewise run length encoding $\{L_i\}$
begin
  // Number of groups
  $k \leftarrow 0$
  foreach $i \in [1, N]$ **do**
    // Set of groups in current line
    $L_i \leftarrow \emptyset$
    // Current open group: c_c, j_c
    $c_c \leftarrow D_{i,1}$
    $j_c \leftarrow 1$
    foreach $j \in [2, M]$ **do**
      if $c_c \neq D_{i,j}$ **then**
        // Finalize open group
        if $c_c \in$ *set of inspected color classes* $\subseteq \mathfrak{C}$ **then**
          $k \leftarrow k + 1$
          $L_i \leftarrow L_i \cup (i, j_c, j - j_c, c_c, k, k)$
        end
        // Open new group
        $c_c \leftarrow D_{i,j}$
        $j_c \leftarrow j$
      end
    end
    // Finalize open group at the end of line
    if $c_c \in$ *set of inspected color classes* $\subseteq \mathfrak{C}$ **then**
      $k \leftarrow k + 1$
      $L_i \leftarrow L_i \cup (i, j_c, M - j_c, c_c, k, k)$
    end
  end
  return $\{L_1, \ldots, L_N\}$
end

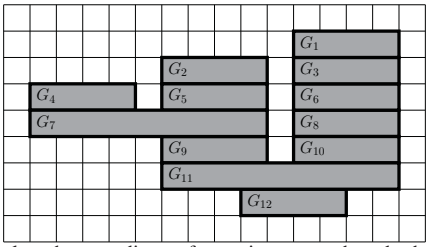
Algorithm 1: Run length encoding

Input: G_a, G_b
begin
  // Follow label of root group
  if $s(G_{r(G_a)}) < s(G_{r(G_b)})$ **then**
    // Update label of root group
    $s(G_{r(G_b)}) \leftarrow s(G_{r(G_a)})$
    // Update index of root group
    $r(G_b) \leftarrow r(G_a)$
  else
    Unite(G_b, G_a)
  end
end

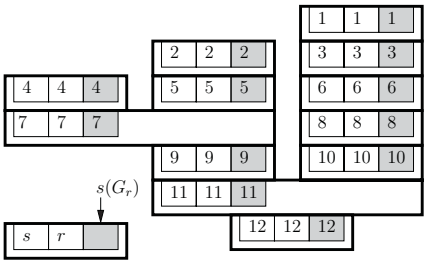
Algorithm 2: Unite(G_a, G_b)

```
Input: Run length encoding {L_i}
begin
    for l ← 2; l ≤ |{L_i}|; l ← l + 1 do
        // A, B are the sets of groups of the last and current line
        A ← L_{l−1}, B ← L_l
        // Indices of the currently examined groups in the respective lines
        i ← 1, j ← 1
        while (i ≤ |A|) ∧ (j ≤ |B|) do
            if x(B_j) > x(A_i) + w(A_i) then
                // A_i ends before B_j starts, no reachability
                i ← i + 1
                continue
            end
            if x(A_i) > x(B_j) + w(B_j) then
                // B_j ends before A_i starts, no reachability
                j ← j + 1
                continue
            end
            // A_i and B_j are reachable according to definition ??
            if c(A_i) = c(B_j) then
                Unite(A_i, B_j)
            end
            // Update group indices
            if x(B_j) + w(B_j) > x(A_i) + w(A_i) then
                i ← i + 1
            else
                j ← j + 1
            end
        end
    end
end
```

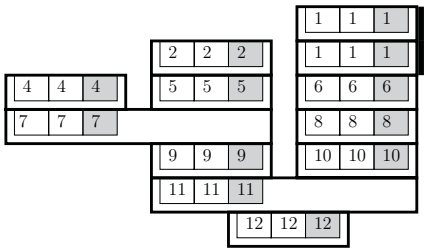
Algorithm 3: Segmentation as specialised union find



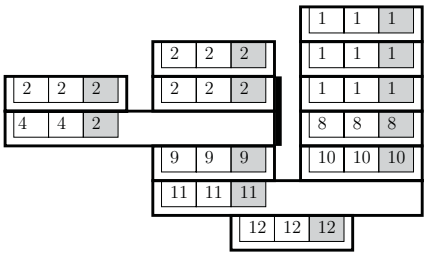
(a) Run length encoding of an image gathered the groups $G_1, \ldots, G_{12}$ of equal color class. By applying Algorithm 3 they are joined. In this expample the proceeding of unifying by Algorithm 2 is shown, reachability of the respective groups is premised.



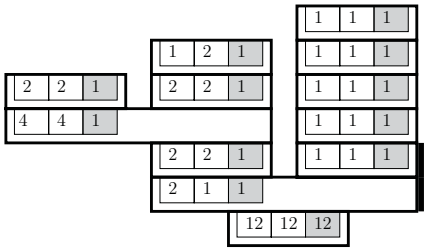
(b) Inititialisation of the labels (first field of a group) with $s(G_i) \leftarrow i$ and index of the root group (second field of the group) $p(G_i) \leftarrow s(G_i)$. The grey field always shows the label $s(G_{r(G_i)})$ of the root group.



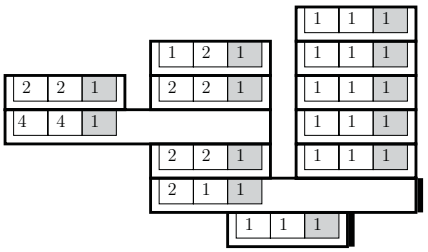
(c) The black token signals the union of the groups G_1 and G_3: Due to $s(G_{r(G_1)}) = s(G_1) = 1 < s(G_{r(G_3)}) = s(G_3) = 3$, G_1 becomes new root group of G_3. $s(G_{r(G_3)}) \leftarrow s(G_{r(G_1)})$, $r(G_3) \leftarrow r(G_1)$. The figure shows the state after this union.



(d) Union of the groups G_5 and G_7: Due to $s(G_{r(G_5)}) = s(G_2) = 2 < s(G_{r(G_7)}) = s(G_4) = 4$, G_2 becomes new root group of G_4. $s(G_{r(G_5)}) \leftarrow s(G_{r(G_7)})$, $r(G_5) \leftarrow r(G_7)$. The former values $s(G_7)$ and $p(G_7)$ are not modified. Already modification of the root group validates $s(G_{r(G_7)})$.

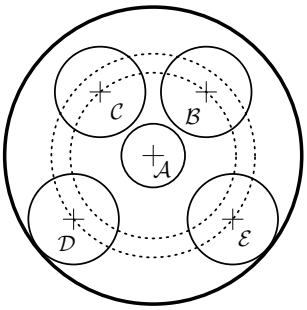


(e) Union of the groups G_{10} and G_{11}: Due to $s(G_{r(G_{10})}) = s(G_1) = 1 < s(G_{r(G_{11})}) = s(G_2) = 2$, G_1 becomes new root group of G_2. $s(G_{r(G_{11})}) \leftarrow s(G_{r(G_{10})})$, $r(G_{11}) \leftarrow r(G_{10})$. The segment membership of the left branch swaps to 1.

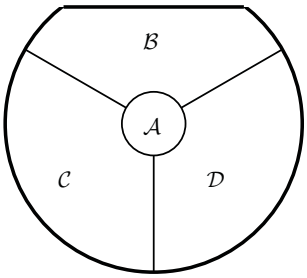


(f) The union of the segment 1 is finished.

Fig. 7. Important steps of the segment union by Algorithm 3.



(a) Standard pattern for RoboCup small size robots



(b) Pattern with better utilisation of available area

Fig. 8. Robot identification patterns

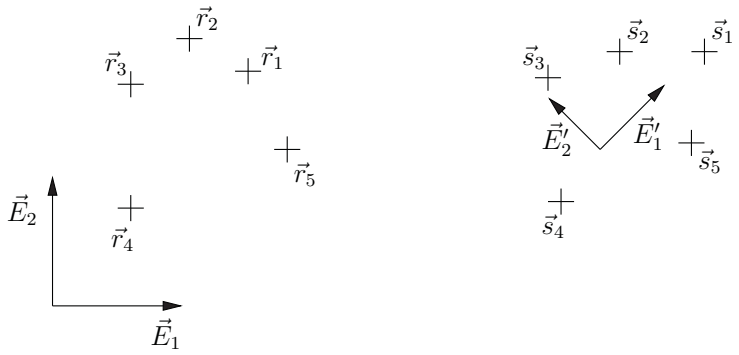


Fig. 9. Typical utilisation of the Helmert transform to determine the position and the rotation of the observer frame $(\vec{E}'_1, \vec{E}'_2)$ within the reference frame $(\vec{E}_1, \vec{E}_2)$ by matching ident points.

identification, an accumulator is filled with the intersection of the run length encoding and this region of interest. Thus as areas as centers of gravity are determined.

The first step of the evaluation of the RoboCup standard pattern is the arrangement of the segments to inner or outer circle. The feature for marker identification at a respective circle is the azimuthal gap with respect to the center. For the alternative pattern, a color class is reserved as start feature. The second step in both cases is to identify the respective segments by exploiting the azimuthal gaps. For this a segment's center of gravity is considered as $\vec{r} = \begin{pmatrix} r_x & r_y & 0 \end{pmatrix}^T$. Relative order of two segments i, j with respect to a center segment c then is found by evaluation of $\operatorname{sign}\left(((\vec{r}_i - \vec{r}_c) \times (\vec{r}_j - \vec{r}_c))_z\right)$. Now the color classes of the identified segments allow robot attribution according to an explicit encoding scheme.

4. OBJECT LOCALISATION UTILISING HELMERT TRANSFORM

4.1. 4-parameter transform for localisation

A basic problem of geodesy is the accurate determination of object positions for map construction applying methods of distance and angle measurement. For this purpose the positions of well known reference points in the yet unknown observer system is determined by a polar to cartesian transform of the corresponding measured values. Those positions of the known reference points in the observer system allow the calculation of the observer systems parameters. Thus positions of new objects may be transferred to the reference system. One approach to this problem is the Helmert transform, usually exercised as 7-parameter transform for position determination problems in $\mathbb{R}^3$, but also used as 4-parameter transform for problems in $\mathbb{R}^2$ which may attain focus in this contribution. The 4-parameter Helmert transform consists of simple shifting, scaling and rotating operations in the form of

$$m \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} (\vec{s}_i - \vec{s}_0) = (\vec{r}_i - \vec{r}_0) + \vec{\varepsilon}_i, \tag{12}$$

where $\vec{r}_i$ and $\vec{s}_i$ are the positions of ident points in the reference respective in the observer frame, see Figure 9.

Points may be transferred from one system to the other using the translations $\vec{r}_0$ and $\vec{s}_0$ in the corresponding frames, a scaling factor m and the rotation angle φ between the x-axes of the systems. The vector $\vec{\varepsilon}_i$ denotes the residual gap of the position of an ident point in both systems induced by measurement effects. The 7-parameter transform utilises the same operations but accounts the higher degree of freedom in $\mathbb{R}^3$.

If $\vec{\varepsilon}_i = \vec{0}\ \forall i$, the evaluation of two points' positions in both systems would be sufficient to calculate the transform's parameters. Since generally $\vec{\varepsilon}_i \neq \vec{0}$, a least squares method is used to determine the parameters of the best fitting transform. Underlying optimisation criterion is the total error as the sum of the squares of the residual gaps

$$E = \sum_{i=1}^{n} \vec{\varepsilon}_i^{\,2} \rightarrow \min, \tag{13}$$

which is constituted [GJ07] by shifting about the arithmetic averages of the points' positions

$$\vec{s}_0 = \frac{1}{n} \sum_{i=1}^{n} \vec{s}_i, \tag{14}$$

$$\vec{r}_0 = \frac{1}{n} \sum_{i=1}^{n} \vec{r}_i \tag{15}$$

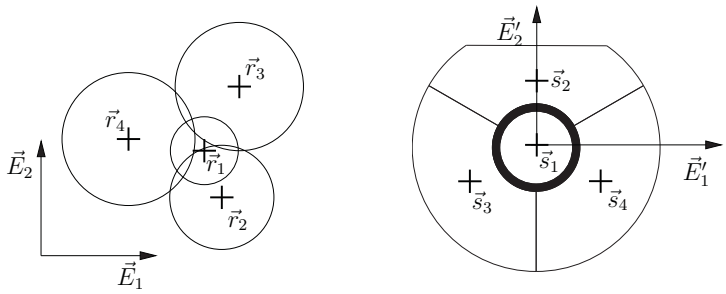


Fig. 10. Application of the Helmert transform for determination of pattern position and orientation: The model based centers of gravity of the markers act as local observation of the reference points, while the centers of segments detected by image processing determine their positions in the reference frame.

and a scaling rotation following

$$m = \frac{\sqrt{\left(\sum_{i=1}^{n}\left(R_{i_x}S_{i_x} + R_{i_y}S_{i_y}\right)\right)^2 + \left(\sum_{i=1}^{n}\left(R_{i_y}S_{i_x} - R_{i_x}S_{i_y}\right)\right)^2}}{\sum_{i=1}^{n}\left(S_{i_x}^2 + S_{i_y}^2\right)} \text{ and} \tag{16}$$

$$\varphi = \arg\left(\begin{pmatrix}\sum_{i=1}^{n}\left(R_{i_x}S_{i_x} + R_{i_y}S_{i_y}\right)\\ \sum_{i=1}^{n}\left(R_{i_y}S_{i_x} - R_{i_x}S_{i_y}\right)\end{pmatrix}\right), \tag{17}$$

where $\vec{R}_i := \vec{r_i} - \vec{r_0}$ and $\vec{S}_i := \vec{s_i} - \vec{s_0}$ are the reduced positions. The function $\arg$, often depicted as "$\mathrm{atan2}$" in common programming languages, delivers the quadrant corrected orientation of a vector $\in \mathbb{R}^2$:

$$\begin{aligned}
&\arg : \mathbb{R}^2 \to \mathbb{R}\\
\vec{r} = \begin{pmatrix} r_x \\ r_y \end{pmatrix} \mapsto \arg(\vec{r}) = &\begin{cases}
\begin{cases}
\arctan\left(\frac{r_y}{r_x}\right)\operatorname{sgn}(r_y) & , r_x > 0\\
\frac{\pi}{2}\operatorname{sgn}(r_y) & , r_x = 0 \quad , r_y \neq 0\\
\left(\pi - \arctan\left(\frac{r_y}{r_x}\right)\right)\operatorname{sgn}(r_y) & , r_x < 0
\end{cases}\\
\begin{cases}
0 & , r_x > 0\\
\text{undefined} & , r_x = 0 \quad , r_y = 0.\\
\pi & , r_x < 0
\end{cases}
\end{cases}
\end{aligned} \tag{18}$$

If the origin of the observer's system is transferred into the reference system by (12) under (14)-(17), the task of position determination with respect to minimal sum of the squares of the residual gaps is accomplished. Figure 10 illustrates the application of the Helmert transform for image processing.

4.2. 4-parameter determination with weights

Visual confirmation of the color classification as base of the image segmentation revealed the frequent error of area loss which may distort localisation result. Figure 11 shows an example of misclassification which makes clear, that the original approach, while being an optimal one, does not make use of all available data.

This encourages exploitation of additional data gathered in the process of image evaluation by introduction of weights, which could for example be derived from predicted and measured segment areas. A weight w_i is applied to the corresponding residual gap $\vec{\varepsilon}_i$ to reduce the influence of poorly detected segments. Weighting of a single residual gap leads to the adapted error sum

$$\begin{aligned}
E &= \sum_{i=1}^{n}\left(\omega_i\vec{\varepsilon}_i\right)^2\\
&= \sum_{i=1}^{n}\omega_i^2\left(m\cos(\varphi)\left(s_{i_x} - s_{0_x}\right) - m\sin(\varphi)\left(s_{i_y} - s_{0_y}\right) - \left(r_{i_x} - r_{0_x}\right)\right)^2\\
&\quad + \sum_{i=1}^{n}\omega_i^2\left(m\sin(\varphi)\left(s_{i_x} - s_{0_x}\right) + m\cos(\varphi)\left(s_{i_y} - s_{0_y}\right) - \left(r_{i_y} - r_{0_y}\right)\right)^2.
\end{aligned} \tag{19}$$

Necessary condition for a minimum of this error are

$$\frac{\partial E}{\partial r_{0_x}} = 2\sum_{i=1}^{n} m\omega_i^2 \left(\cos(\varphi)(s_{i_x} - s_{0_x}) - \sin(\varphi)(s_{i_y} - s_{0_y}) - \left(\frac{r_{i_x}}{m} - \frac{r_{0_x}}{m}\right)\right) \stackrel{!}{=} 0 \tag{20}$$

and

$$\frac{\partial E}{\partial r_{0_y}} = 2\sum_{i=1}^{n} m\omega_i^2 \left(\sin(\varphi)(s_{i_x} - s_{0_x}) + \cos(\varphi)(s_{i_y} - s_{0_y}) - \left(\frac{r_{i_y}}{m} - \frac{r_{0_y}}{m}\right)\right) \stackrel{!}{=} 0. \tag{21}$$

Consequence of (20) is

$$\begin{aligned} m\cos(\varphi)\sum_{i=1}^{n}\omega_i^2 s_{i_x} - m\sin(\varphi)\sum_{i=1}^{n}\omega_i^2 s_{i_y} - \sum_{i=1}^{n}\omega_i^2 r_{i_x} & \\ = m\cos(\varphi)s_{0_x}\sum_{i=1}^{n}\omega_i^2 - m\sin(\varphi)s_{0_y}\sum_{i=1}^{n}\omega_i^2 - r_{0_x}\sum_{i=1}^{n}\omega_i^2 & \end{aligned} \tag{22}$$

and equating coefficients leads to

$$\cos(\varphi) : m\sum_{i=1}^{n}\omega_i^2 s_{i_x} = m s_{0_x}\sum_{i=1}^{n}\omega_i^2 \tag{23}$$

$$\sin(\varphi) : -m\sum_{i=1}^{n}\omega_i^2 s_{i_y} = -m s_{0_y}\sum_{i=1}^{n}\omega_i^2 \tag{24}$$

$$\text{absolute term} : -\sum_{i=1}^{n}\omega_i^2 r_{i_x} = -r_{0_x}\sum_{i=1}^{n}\omega_i^2. \tag{25}$$

Analogously (21) implies

$$\begin{aligned} m\sin(\varphi)\sum_{i=1}^{n}\omega_i^2 s_{i_x} + m\cos(\varphi)\sum_{i=1}^{n}\omega_i^2 s_{i_y} - \sum_{i=1}^{n}\omega_i^2 r_{i_y} & \\ = m\sin(\varphi)s_{0_x}\sum_{i=1}^{n}\omega_i^2 + m\cos(\varphi)s_{0_y}\sum_{i=1}^{n}\omega_i^2 - r_{0_y}\sum_{i=1}^{n}\omega_i^2 & \end{aligned} \tag{26}$$

and therefore

$$\sin(\varphi) : m\sum_{i=1}^{n}\omega_i^2 s_{i_x} = m s_{0_x}\sum_{i=1}^{n}\omega_i^2 \tag{27}$$

$$\cos(\varphi) : m\sum_{i=1}^{n}\omega_i^2 s_{i_y} = m s_{0_y}\sum_{i=1}^{n}\omega_i^2 \tag{28}$$

$$\text{absolute term} : -\sum_{i=1}^{n}\omega_i^2 r_{i_y} = -r_{0_y}\sum_{i=1}^{n}\omega_i^2. \tag{29}$$

Hereby the translations are

$$\vec{s_0} = \frac{\sum_{i=1}^{n}\omega_i^2\vec{s_i}}{\sum_{i=1}^{n}\omega_i^2} \tag{30}$$

and

$$\vec{r_0} = \frac{\sum_{i=1}^{n}\omega_i^2\vec{r_i}}{\sum_{i=1}^{n}\omega_i^2}. \tag{31}$$

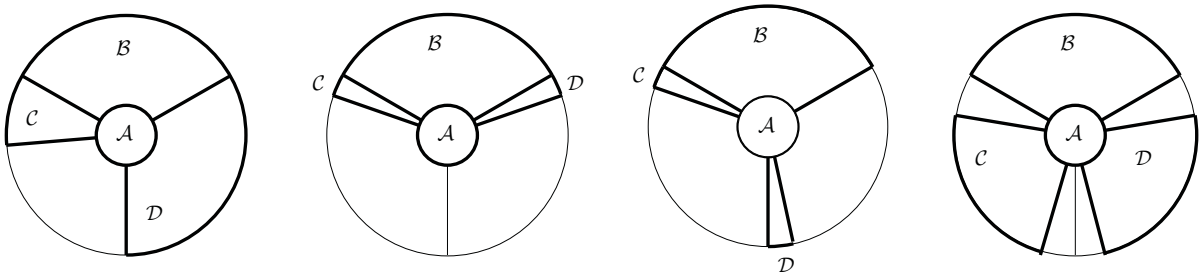


Fig. 11. Examples of misclassification: (a) shows a common case which motivated for parameter determination with weights; In (b) a high position error, but no angle error will occur; (c) will introduce high angle error; In (d) misclassification will not affect localisation.

Substitution of $\vec{R}_i := \vec{r}_i - \vec{r}_0$ and $\vec{S}_i := \vec{s}_i - \vec{s}_0$ as well as $a := m\cos(\varphi)$ and $o := m\sin(\varphi)$ leads to the partial derivatives

$$\frac{\partial E}{\partial a} = 2\sum_{i=1}^{n} \omega_i^2 \left((aS_{i_x} - oS_{i_y} - R_{i_x})S_{i_x} + (oS_{i_x} + aS_{i_y} - R_{i_y})S_{i_y}\right) \stackrel{!}{=} 0 \tag{32}$$

and

$$\frac{\partial E}{\partial o} = 2\sum_{i=1}^{n} \omega_i^2 \left(\left(-aS_{i_x} + oS_{i_y} + R_{i_x}\right) S_{i_y} + \left(oS_{i_x} + aS_{i_y} - R_{i_y}\right) S_{i_x}\right) \stackrel{!}{=} 0. \tag{33}$$

Equation (32) is suited to determine parameter a:

$$\sum_{i=1}^{n} \omega_i^2 \left(R_{i_x}S_{i_x} + R_{i_y}S_{i_y}\right) = \sum_{i=1}^{n} \omega_i^2 \left((aS_{i_x} - oS_{i_y})S_{i_x} + (oS_{i_x} + aS_{i_y})S_{i_y}\right) = a\sum_{i=1}^{n} \omega_i^2 \left(S_{i_x}^2 + S_{i_y}^2\right), \tag{34}$$

analogously parameter o may be provided by (33):

$$\sum_{i=1}^{n} \omega_i^2 \left(R_{i_y}S_{i_x} - R_{i_x}S_{i_y}\right) = \sum_{i=1}^{n} \omega_i^2 \left((-aS_{i_x} + oS_{i_y})S_{i_y} + (oS_{i_x} + aS_{i_y})S_{i_x}\right) = o\sum_{i=1}^{n} \omega_i^2 \left(S_{i_x}^2 + S_{i_y}^2\right). \tag{35}$$

With $a^2 + o^2 = m^2(\cos^2(\varphi) + \sin^2(\varphi)) = m^2$ arises

$$m = \frac{\sqrt{\left(\sum\limits_{i=1}^{n} \omega_i^2 \left(R_{i_x}S_{i_x} + R_{i_y}S_{i_y}\right)\right)^2 + \left(\sum\limits_{i=1}^{n} \omega_i^2 \left(R_{i_y}S_{i_x} - R_{i_x}S_{i_y}\right)\right)^2}}{\sum\limits_{i=1}^{n} \omega_i^2 \left(S_{i_x}^2 + S_{i_y}^2\right)}. \tag{36}$$

Conforming to the substititions a and o holds $\frac{a}{m} = \cos(\varphi)$ and $\frac{o}{m} = \sin(\varphi)$ which is satisfied by

$$\varphi = \arg\left(\begin{pmatrix} \sum\limits_{i=1}^{n} \omega_i^2 \left(R_{i_x}S_{i_x} + R_{i_y}S_{i_y}\right) \\ \sum\limits_{i=1}^{n} \omega_i^2 \left(R_{i_y}S_{i_x} - R_{i_x}S_{i_y}\right) \end{pmatrix}\right). \tag{37}$$

Again player postion may be determined by transforming the players origin of coordinates into the reference system with (12). Thereby the parameters of the transform following (30), (31), (36) and (37) proceed to the results of (14), (15), (16) and (17) for weights $w_i = 1$.

4.3. Application of the transform

For examination whether errors in position determination may be decreased by introduction of weights into the optimisation criterion of Helmert's transform, a synthetic test based on an application in image processing for

Marker	x_0	y_0	r_i	r_a	θ_0	θ_1
$\mathcal{A}$	0	0	0	25mm	0	2π
$\mathcal{B}$	0	0	25mm	90mm	$\pi/6$	$\theta_0 + 2\pi/3$
$\mathcal{C}$	0	0	25mm	90mm	$5\pi/6$	$\theta_0 + 2\pi/3$
$\mathcal{D}$	0	0	25mm	90mm	$3\pi/2$	$\theta_0 + 2\pi/3$

Table 1. Parameters of the marker pattern

RoboCup is suggested. A robot of RoboCup small-size league is identified and located by a unique marker pattern perceived with the help of a video camera mounted above the pitch. The estimation of the influences of classification errors to localisation errors requires the definition of a marker pattern: A player's pattern consists of the markers $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ which all are circle sectors of the parameters $(x_0, y_0, r_i, r_a, \theta_0, \theta_1)$ with the sectors underlying circles center (x_0, y_0), inner radius of the sector r_i, outer radius of the sector, start angle θ_0 and end angle θ_1. The definition of the according parameters with respect to the robot system is shown in Table 1.

The area A of a marker is

$$A = \int_{r_i}^{r_a} \int_{\theta_0}^{\theta_1} r d\varphi dr = \frac{1}{2}(\theta_1 - \theta_0)({r_a}^2 - {r_i}^2) \tag{38}$$

and the center of gravity $\vec{s}$ follows

$$\begin{aligned} \vec{s} &= \frac{1}{A} \int_{r_i}^{r_a} \int_{\theta_0}^{\theta_1} \begin{pmatrix} x_0 + r\cos(\theta) \\ y_0 + r\sin(\theta) \end{pmatrix} r d\theta dr \\ &= \frac{1}{A} \left(\frac{r_a^3 - r_i^3}{3} \begin{pmatrix} sin(\theta_1) - \sin(\theta_0) \\ -\cos(\theta_1) + \cos(\theta_0) \end{pmatrix} + \frac{r_a^2 - r_i^2}{2}(\theta_1 - \theta_0) \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \right). \end{aligned} \tag{39}$$

For the simulation of disturbance by image processing, a marker pattern may be manipulated directly by its sector's parameters as shown in Algorithm 4 and Algorithm 5, see Figure 12. Thus artefacts of the color classification's false-negatives can consistently be emulated, which in the case of direct manipulation of the sector's centers of gravity and areas would not be guaranteed.

Input: Position (x, y),orientation φ
Output: Marker parameters $(x'_0, y'_0, r'_i, r'_a, \theta'_0, \theta'_1)$
begin
`// Markers quality states the ratio of disturbed and undisturbed markes area.`
Define a markers quality $\frac{A'}{A} \in (0, 1]$.
`// This quality finalises the radius of the center marker`
Assign radius of the center marker $r'_a = r_a\sqrt{\frac{A'}{A}}$.
`// The disturbed central marker should lay within the undisturbed marker,`
`thereby codomains of the center of gravity arise with` $x_{\max} = r_a - r'_a$.
Define $x'_0 \in [-x_{\max}, x_{\max}]$.
`// and via Pythagorean theorem follows` $y_{\max} = \sqrt{x_{\max}^2 - {x'_0}^2}$.
Define $y'_0 \in [-y_{\max}, y_{\max}]$.
return $(x'_0 + x,\ y'_0 + y,\ 0, r'_a,\ 0,\ 2\pi)$
end

Algorithm 4: Emulation of a disturbed image processing result for the center marker

The areas and centers of gravity of the pattern's markers are determined for use as observer-sided arguments of (30), (31), (36) and (37). Applying algorithms 4 and 5 to all markers of the pattern, disturbed marker parameters are calculated for arbitrary (x, y, φ). The areas and centers of gravity of those disturbed marker parameters are determined as arguments for the reference-sided parts of (30), (31), (36) and (37). Equation (12) leads to the determined position (x', y') and orientation φ' of the pattern. The disturbance induced error then is $\vec{e} := \begin{pmatrix} \Delta x & \Delta y & \Delta\varphi \end{pmatrix}^T := \begin{pmatrix} x' - x & y' - y & \varphi' - \varphi \end{pmatrix}^T$.

As algorithms 4 and 5 do not specify distributions for the respective parameters, any appropriate distribution may be used. Attention should be paid to the fact, that the chosen distributions as the definitions of parameters in

Input: Position (x, y), orientation φ
Output: Marker parameters $(x'_0, y'_0, r'_i, r'_a, \theta'_0, \theta'_1)$
begin
Define marker's quality $\frac{A'}{A} \in (0, 1]$.
// The covered angle $\Delta\theta = \theta_1 - \theta_0$ of the undisturbed sector determines the codomain for the covered angle $\Delta\theta' = \theta'_1 - \theta'_0$ of the disturbed sector.
Define $\Delta\theta' \in [\frac{A'}{A}\Delta\theta, \Delta\theta]$.
// For this covered angle $\Delta\theta'$, the minimal outer radius of the sector $r_{\min} = \sqrt{2\frac{A'}{\Delta\theta'} + r_i^2}$ will permit to achieve markers quality $\frac{A'}{A}$.
Define outer radius $r'_a \in [r_{\min}, r_a]$.
Assign the fitting inner radius $r'_i = \sqrt{{r'_a}^2 - 2\frac{A'}{\Delta\theta'}}$
// Determine the start angle θ'_0 by scattering the gap $\Delta\theta - \Delta\theta'$ with a gap parameter $z \in [0, 1]$.
Define $\theta'_0 = \theta_0 + z(\Delta\theta - \Delta\theta')$.
return $(x'_0 + x,\ y'_0 + y,\ r'_i,\ r'_a,\ \theta'_0 + \varphi,\ \theta'_0 + \Delta\theta + \varphi)$
end

Algorithm 5: Emulation of a disturbed image processing result for an outer marker $\in \{\mathcal{B}, \mathcal{C}, \mathcal{D}\}$

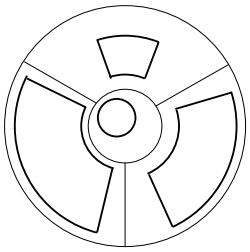


Fig. 12. Example of pattern disturbance by application of Algorithm 4 and Algorithm 5: All disturbed markers lay within their undisturbed primitives – the simulated misclassification lost some pixels with the probability of false negative greater zero, but did not gain pixels, the probability of false positve equals zero.

the given disturbance model itself may induce a bias into the results. Also, the specific drafting of an error scoring norm will affect the results.

To score the error $\vec{e}$ an arbitrarily chosen failure case from the application domain is considered:

1. As the robot's kicker's width is about 80mm, an error of 40mm in x-direction will prohibit a reliable shoot.
2. Errors in x- and y-direction should be scored equally.
3. Rating of the orientation error is done by a distance based targeting accuracy: For a distance of 3m a perpendicular accuracy of 0.2m is required.

This failure case is used to construct a simple weighted square norm for error scoring:

$$|\vec{e}| := \sqrt{\left(\frac{\Delta x}{40\text{mm}}\right)^2 + \left(\frac{\Delta y}{40\text{mm}}\right)^2 + \left(\frac{\Delta\varphi}{\arctan\left(\frac{0.2\text{m}}{3\text{m}}\right)}\right)^2}, \tag{40}$$

where $|\vec{e}| > 1$ signals intolerable localisation errors.

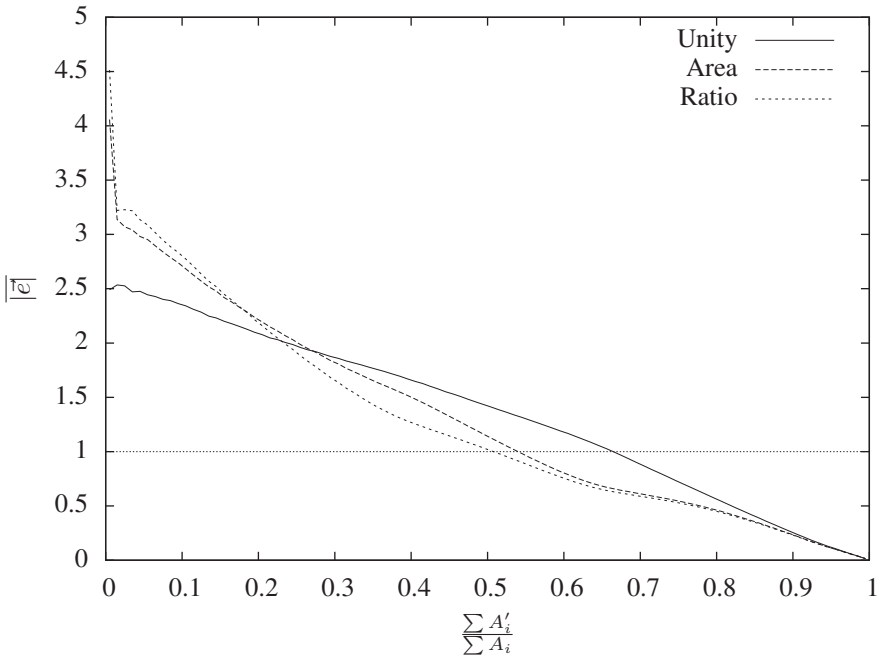
The influence of weighting approaches to position determination accuracy was examined empirically for

Unity $w_i = 1$ – Basic case underlying equations (14)-(17),

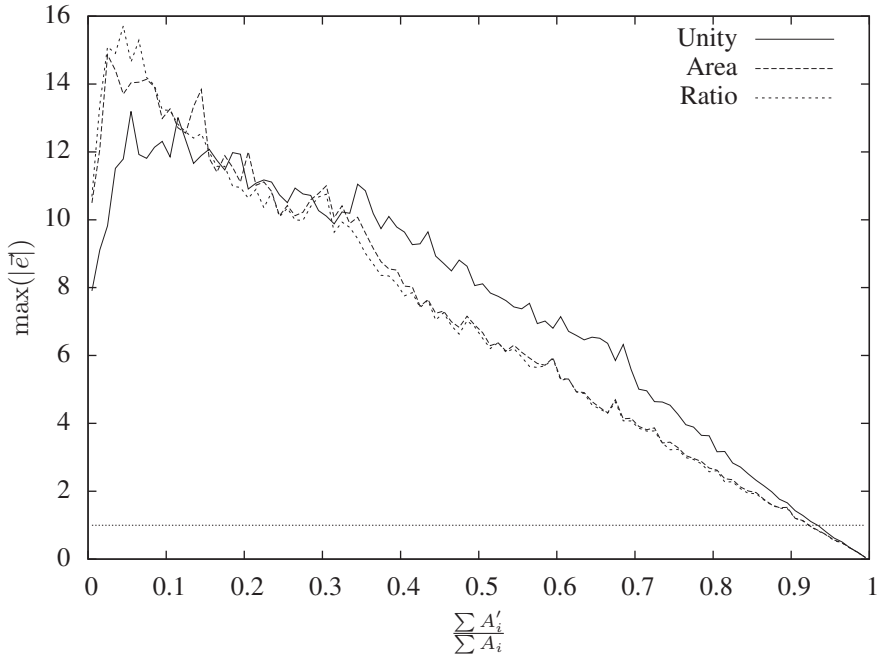
Area $w_i = A'_i$ – Measured or emulated area as weight of a marker's residual gap and

Ratio $w_i = \frac{A'_i}{A_i}$ – Ratio between measured or emulated area and undisturbed area as weight of a marker's residual gap.

Therefore, synthetic test data was used, which emulates consistently degenerated image processing results. Figure 13 shows the result of an experiment with the aid of uniform distributed pseudo random numbers. For this 10^8 data records have been generated with Algorithms 4 and 5. They have been aggregated into 1% quality classes with respect to $\sum A'_i / \sum A_i$. For $w_i = 1$ the plot shows a required mean imaging quality of ≈70% for the mean of the weighted Error (40) $|\vec{e}| = 1$. This value could be dropped by ≈10% through introduction of weights. The mean quality for the maximum weighted error dropped by ≈2%. Thus the system becomes slightly more error tolerant.



(a) Mean of the weighted error over mean-quality



(b) Maximum of the weighted error over mean-quality

Fig. 13. Experimental result: weighted error E over mean-quality $A'/A = \sum_i A'_i / \sum_i A_i$ for $w_i = 1$ (Unity), $w_i = A'_i$ (Area) and $w_i = \frac{A'_i}{A_i}$ (Ratio). By the introduction of weights, the required mean quality of the image processing result for average system failure case dropped by ca. 10%. The required mean quality for failure case dropped by ca. 2%. Note that the plots show mean respectively maximum of aggregated data over 1% quality classes. For this 10^8 data records have been generated with Algorithms 4 and 5.

5. CONCLUSION

In this contribution the fundamentals of an image processing system for robot localisation in the domain of RoboCup have been shown. This system is based on [Ler08] and strongly influenced by [BBV00]. The system utilises run length encoding for color image segmentation. In contrast to [BBV00] a precalculated discrete Voronoi classificator based on the Mahalanobis distance and a sample based minimum distance is applied. Due to vignetting effects, both distance measures have not been capable to discriminate color classes robustly. This drawback was successfully countered with localised image amplification. As the color classificator is generated by externally labeled samples, it may be considered as a form of supervised learning. Future work could abandon the dependence on labeled data and thus proceed to a form of unsupervised learning.

With a classificator resolution of 5bit per channel and a region of interest of about 1.5MP, the stated algorithms are capable of evaluating >40 frames per second on a 2.2GHz Core2Duo notebook. Thus requirements for processing speed are complied. Also the system has shown to be robust in practice. For evaluation of the image segments, the regression method Helmert transform has been applied to match the pattern model and the measurement data. The transform has been adapted to utilise additional data gathered in the process of image processing. Synthetic tests suggest that this adaption can reduce the mean localisation error or alternatively may slightly decline the demands of color classification quality.

6. REFERENCES

[BBD+09] P. Blank, M. Bleier, S. Drexler, J. Kallwies, P. Kugler, D. Lahmann, Ph. Nordhus, C. Riess, T. Swadzba, and J. Tully. ER-Force Team Description Paper for RoboCup 2009. In RoboCup Foundation, editor, *Proceedings-CD*, 2009.

[BBV00] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, pages 2061 – 2066, October 2000.

[Die06] R. Diestel. *Graphentheorie*. Springer Verlag, 2006.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[GGA+05] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau. Demosaicking: Color filter array interpolation. *IEEE Signal Process. Mag.*, 22(1):44–54, Jan. 2005.

[GJ07] F. J. Gruber and R. Jockel. *Formelsammlung für das Vermessungswesen*. Teubner-Verlag, 2007.

[ITM01] F. H. Imai, N. Tsumura, and Y. Miyake. Perceptual color difference metric for complex images based on mahalanobis distance. *J. Electronic Imaging*, 10:38539–3, 2001.

[KO05] D. Klimešová and E. Ocelíková. Bayes classifier in multidimensional data classification. In *15th International Conference on Process Control 2005 in Štrbské Pleso, Slovakia*, pages 188–1 – 188–5, June 2005.

[Koh90] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[Ler08] S. Lerm. Entwurf und Implementation einer neuen Version der Bildverarbeitung für die RoboCup Small Size League. Technische Universität Ilmenau, Fakultät für Maschinenbau, 2008. Diplomarbeit.

[LW95] C.-T. Li and R. Wilson. Image segmentation using multiresolution fourier transform. Technical report, 1995.

[M. 96] M. Ester and H.-P. Kriegel and J. Sander and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231. AAAI Press, 1996.

[Mah36] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936.

[MAN+11] M. Mousakhani, M. Ataei, A. K. Nasab, A. H. Mandegar, and A. Norouzi. Small Size Robot League. http://2011.iranopen.ir/Default.aspx?tabid=99&language=en-GB, visited July 2011.

[MDSG02] C.H. Messom, S. Demidenko, K. Subramaniam, and G.S. Gupta. Size/position identification in real-time image processing using run length encoding. In *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, volume 2, pages 1055 – 1059 vol.2, 2002.

[MTJAL05] E. Micheli-Tzanakou, Y. Jin, E. Angelini, and A. Laine. Wavelets in medical image processing: Denoising, segmentation, and registration. In J. S. Suri, D. L. Wilson, and S. Laxminarayan, editors, *Handbook of Biomedical Image Analysis*, Topics in Biomedical Engineering. International Book Series, pages 305–358. Springer US, 2005.

[PBC05] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: analysis and comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):148–154, 2005.

[Ren] R. J. Renka. Image segmentation with a sobolev gradient method.

[Sud06] A. Sud. Efficient computation of discrete voronoi diagram and homotopy-preserving simplified medial axis of a three-dimensional polyhedron. University of North Carolina at Chapel Hill Chapel Hill, NC, USA, 2006. Dissertation.

[Wan05] L. Wang. *Support Vector Machines: Theory and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[YGZ03] Q. Ye, W. Gao, and W. Zeng. Color image segmentation using density-based clustering. *Multimedia and Expo, IEEE International Conference on*, 2:401–404, 2003.

[YSL09] Y. Yan, Y. Shen, and S. Li. Unsupervised color-texture image segmentation based on a new clustering method. In *Proceedings of the 2009 International Conference on New Trends in Information and Service Science*, pages 784–787, Washington, DC, USA, 2009. IEEE Computer Society.

[ZH09] J. Zhang and J. Hu. Automatic segmentation technique for color images. *ICGST International Journal on Graphics, Vision and Image Processing, GVIP*, 09:41–49, 2009.