Image Processing for Localization of Mobile Robots

[1]*Ján JADLOVSKÝ,* [2]*Michal VARGA,* [3]*Michal KOPČÍK*

[1,2,3]Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics Technical University of Košice, Slovak Republic

[1]jan.jadlovsky@tuke.sk, [2]michal.varga.6@student.tuke.sk, [3]michal.kopcik@tuke.sk

***Abstract* – We present a real-time mobile robot tracking algorithm for robotic soccer and other applications which detects and tracks mobile robots with specific or arbitrary appearance. In robotic soccer, it uses a custom colour pattern for classification of our players. We briefly describe the implementation in context of the existing control application architecture.**

***Keywords* – computer vision, object tracking, robotic soccer, WCF**

I. Introduction

This algorithm was designed as a replacement for the existing vision system for the robotic soccer project. The existing solution had several shortcomings – it could not detect unknown robots (i.e. opponents) nor the ball. The new system must be able to do all that and should be usable in other applications as well.

A. Robotic soccer

The robotic soccer is a sport organised by FIRA – Federation of International Robot-soccer Association. Of the several leagues available, we chose to compete in MiroSot Middle League. Robotic soccer players are controlled remotely from a computer running the strategy control and vision system application. This computer is equipped with an overhead camera mounted above the playing field. The players continuously receive updates on their position and new waypoints from the control application.

B. Soccer player

For this competition we designed and constructed custom mobile robotic soccer players described in [1] (prototype on Fig. 1). Our players have a custom designed pattern (Fig. 2) placed on top to enable classification by the vision system. This pattern consists of colour (as required by the rules [2]) and bar code part. The bar code part, consisting of three stripes, encodes a three-digit binary identification number specifying the role of the player. The code is read left to right, starting with the most significant bit.

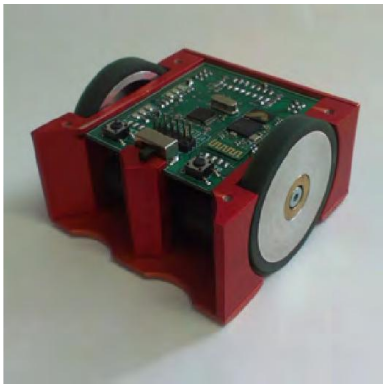

Fig. 1: Mobile robotic soccer player [1]

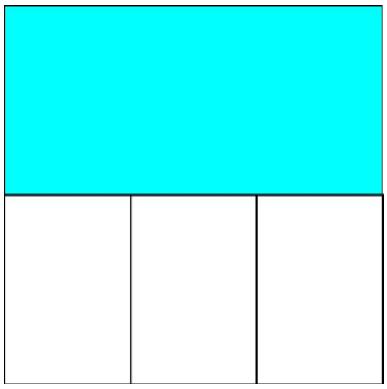

Fig. 2: Custom pattern used on our players [3]

II. Related Work

Object tracking is the domain of interest of many researchers around the world and has been undergoing rapid development in the past few years. Visual Object Tracking challenge in 2014 [4] evaluated 38 new and existing object tracking algorithms, out of which none came out consistently superior. We attempted to use one of the keypoint tracker algorithm – CMT [5] (Consensus-based Matching and Tracking of Keypoints for Object Tracking) – which eventually proved to be unsuitable for our task due to the lack of salient keypoints detected on some tested patterns.

III. Tracking Algorithm

As outlined in [6], the three steps of object tracking are detection, classification and tracking. We developed custom algorithms for the classification and tracking steps and used an Improved Adaptive Gaussian Mixture Model for Background Subtraction [7] in the first step.

In our algorithm, each object is re-classified in every frame to correct potential previous errors. Therefore, we perform tracking before classification so that we know the previous classification results of the tracked object assigned to the currently classified image region.

A. Detection

The only means of detecting mobile robots with no prior knowledge of their appearance or background is through motion detection. We used the aforementioned Gaussian mixture model-based algorithm to detect moving regions. This method is able to adapt itself to any static background so no prior information is required. Also, shadows are detected separately so the results are mostly invariant to lighting conditions.

When used for robotic soccer, we supply the expected robot area A_E (as defined by the rules) as a parameter. We filter out regions with area A_n lower than a specified value A_{min} and large regions are split using the k-means clustering algorithm. The number of clusters k_n for the n-th region is determined by rounding the relative region area A_{Rn} to the nearest integer ($nint()$ function) as shown by (1, 2). Very large regions (with k_n greater than k_{max}) are also discarded.

$$A_{Rn} = \frac{A_n}{A_E} \tag{1}$$

$$k = nint(A_{Rn}) \tag{2}$$

B. Tracking

Object tracking is accomplished by a point tracker using a single point (object centre) for each object. After the moving regions are extracted and processed, (O, R, d^2) triplets are generated for every object-region pair, where d^2 stands for the square distance between the expected object position and region centre. Position of the object is predicted using its last known location and velocity vector. The list is then ordered by ascending distance. By iterating over the list, regions are assigned to tracked objects. No object or region can be assigned more than once. If any region remains unassigned, a new object is initialized in its position. Objects that have not been seen for specified number of frames are discarded. After assignment, position and velocity is updated for each tracked object.

C. Classification

As stated in the rules [2], pattens on top of the player must contain at least 3.5cm $\times$ 3.5cm area (label) filled with their assigned team colour. We use this requirement to classify players into teams.

After converting the frame to HLS colour space, we segment the colour labels using the supplied threshold values for each channel. Resulting areas are fed through a median filter and then filtered by their area. For each resulting region, we calculate its centroid. When classifying a detected moving region, we test for presence of colour label centroid inside the region. If such centroid is found, we push a “yes” vote into object’s team voting queue, otherwise we push in a “no”. Then we determine its team by counting the votes in the queue. If there are more “yes” votes than “no” votes, we declare the player as ours, otherwise it is an opponent. This voting mechanism enhances the temporal stability of the classification and eliminates the effect of single-frame classification errors.

If the classified player is ours and its colour label has been detected in the current frame, we read the bar code to determine its ID (role). First, the orientation of the player is determined by performing PCA analysis of the colour label region. The second component is parallel to the

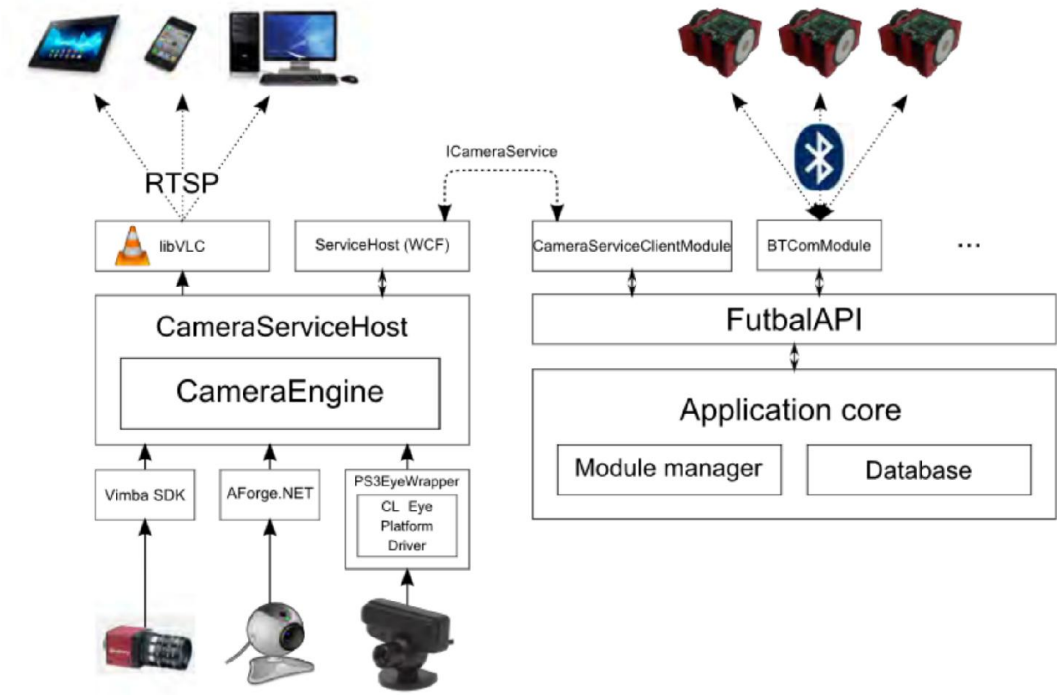

Fig. 3: Architecture of the proposed robotic soccer control system. Left half represents the new image-processing application, the right side shows integration with the existing control application.

forward-backward axis of the robot. We calculate the dot product of this vector and the vector from player centre to colour label centre. If the result is negative, we reverse the second component so it points forwards.

After the forward direction is known, we place the three sampling points on the bar code and transform these pixel values to three-digit binary code. The ID, represented by this code, is then pushed into another voting queue and the final ID is calculated in the same manner as when determining team of the player (now with 5 possible answers – one for each player role).

IV. Implementation

Considering the existing software solution, we decided to implement this algorithm in a stand-alone application. Therefore, it is not bound to the soccer control application and can be used freely in other projects. The application communicates with other processes using a WCF (Windows Communication Foundation) contract.

We built the application on .NET Framework using C# and C++ languages. OpenCV library was used for the image processing part implemented in a native C++ static library and linked in a managed .NET wrapper. This allows for the best possible performance optimization using the Microsoft Visual Studio compiler.

We extended the current solution's hardware support by PS3 Eye camera, capable of relatively high speed video capture (up to 185 FPS) at an affordable price point. Another addition is RTSP network video streaming. The final architecture of the soccer control system is depicted in Fig. 3.

A new WCF communication module has been created for the existing control application which receives data from the vision system and stores them in the internal database. Using the WCF contract, it is possible to control the object tracking process and modify all parameters remotely and in real time.

V. Results

We tested the algorithm with PS3 Eye camera, a red ball, and two robots – one wearing our pattern and other without cover to test opponent detection. The application successfully detected both players in all situations. When the two players met and created a single moving region, the algorithm correctly split it in half and continued tracking both players. The classification results stabilised after few frames and values stayed correct. Ball has been detected reliably in almost every frame, even during fast movement and while being pushed by a player.

We experienced classification problems with poor lighting conditions due to incorrect colour label segmentation. This caused significant noise in orientation and ID classification. We managed to correct this by introducing proper scene lighting.

The performance was satisfactory, reaching around 30 frames per second with the used camera and running on 3.2 GHz dual-core processor. We experienced no issues with WCF communication and RTSP video streaming was reliable, albeit lagging behind by approximately 3 seconds.

ACKNOWLEDGMENT

This work has been supported by the Research and Development Operational Program for project: University Science Park Technicom for innovative applications with knowledge technology support, ITMS code 26220220182, co-financed by the ERDF (80%) and by grant KEGA - 001TUKE-4/2015 (20%).

REFERENCES

[1] M. Kopčík, R. Bielek, and J. Jadlovský, "Construction and operating system of robosoccer agents," in *13th Scientific Conference of Young Researchers*, 2013.

[2] FIRA. (2006) Fira mirosot game rules. Federation of International Robot-soccer Association. [Online]. Available: http://www.fira.net/contents/data/MiroSot_Rules_Middle_League.doc

[3] M. Varga, "Rozpoznávanie obrazu a komunikácia v rámci riadenia robotického futbalu," Bachelor's thesis, Technical University of Košice, 2013.

[4] M. Kristan, R. Pflugfelder *et al.*, "The visual object tracking vot2014 challenge results," in *Computer Vision - ECCV 2014 Workshops*, ser. Lecture Notes in Computer Science, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Springer International Publishing, 2015, vol. 8926, pp. 191–217. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16181-5_14

[5] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Winter Conference on Applications of Computer Vision*. IEEE, Mar. 2014.

[6] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol*, vol. 2, 2014.

[7] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.