AKTET
Journal of Institute Alb-Shkenca
Revistë Shkencore e Institutit Alb-Shkenca

ISSN 2073-2244
www.alb-shkenca.org
Copyright © Institute Alb-Shkenca

# APPROACHES TO TRACKING IN COMPUTER VISION
# QASJET NË NDJEKJET NË VIZIONIN KOMPJUTERIK

SONILA DOBI[1], BESNIK DOBI[2]
[1]Technische Universitaet, Muenchen, Germany
[2]Military Academy Skanderbeg, Tirana, Albania
Email: sonila_dobi@mytum.de

**AKTET IV, 3: 468 - 473, 2011**

**PERMBLEDHJE**
Gjurmimi është një sfidë e madhe, por ende ka kërkesa të mëdha në fusha me interes, si robotika, bashkëveprimi makinë-njeri, aplikimet e realitetit të manipuluar, videosurvejimi, operacionet mjekësore të kompjuterizuara, sistemet e navigimit, përpunimi i sinjalit, aplikimet me imazhet etj. Ai kërkon integrim të vizionit kompjuterik, programeve të pajisjeve gjurmuese, algoritme bashkëkohore për problemet e kalibrimit të kamerave dhe të gjurmimit. Aplikime të ndryshme kërkojnë shënues të ndryshëm. Kërkesa kryesore është besueshmëria dhe shpejtësia e gjurmimit. Problemet që sistemi shfaq janë: ruajtja e identitetit të objektivave, zbulim objektesh, trajtim zhdukjesh të plota/pjesshme, rizbulim gjurmësh të humbura dhe rinisje sistemi, gjurmim, ruajtje e performancës në prani të zhurmave ose të pengesave dhe të ndryshimit të kushteve të mjedisit, të jetë në gjendje të gjurmojë në kohë reale. Hulumtimet vazhdojnë në këtë fushë, dhe jo të gjitha problemet janë zgjidhur. Kjo punë tregon si të trajtohen sfidat më delikate në gjurmimet e vizionit kompjuterik.
**Fjale kyçe:** gjurmim, kalibrimi-kamerave, kohë-reale, shënues, vision-kompjuterik.

**SUMMARY**
Tracking is a major challenge, but still shows great demand in many fields of interest including robotics, man-machine interfaces, augmented reality applications, video surveillance, computer-assisted surgery, navigation systems, signal processing, imaging, and many more areas. It requires integration of computer vision, drivers for various kinds of tracking hardware, state-of-the-art algorithms for solving common camera calibration and tracking tasks. Different applications require different types of markers. Mainly required is a reliable and fast tracking system. The problems the system faces include: maintaining the identity of targets, object detection, handling of full or partial occlusions, detecting lost tracks and re-initializing, tracking, maintaining performance in a noisy environment or with clutter and changing environment conditions, to be able to track in real-time. There is still research undergoing in this field, and not all of the problems are solved. This work shows how to tackle the most delicate challenges in tracking in computer vision.
**Key words:** tracking, camera-calibration, real-time, marker, computer-vision.

## I. INTRODUCTION

This paper shows various approaches in computer vision for tracking real world objects and achieving an application using most of these techniques together. Different approaches are analysed, tested and new technologies are combined together to have a robust and simple manipulation of the target. As an example, here is shown how to build a game using several approaches to track different kinds of objects and design a new plug-in to integrate a sophisticated tracking tool with another one specialized in game development. Often tests fail because the wrong approach is used for the specific task, therefore here is explained what should be used and when. The main focus of this work is to achieve a reliable, feasible, flexible, easy to use and less cost-effective tracking tool. Three different approaches are integrated together to create a new hybrid system which takes advantages of all of them individually. Some games which inspired us; Combat (Atari, 1977), Atari Battlezone (Atari, 1980), Rampart (Atari, 1990), Desktop Tower Defense (Paul Preece, 2007).

## II. MATERIALS AND METHODS

In this chapter we describe the most common tracking methods and the justification of the final chosen system. The mentioned tracking methods are; colour based, hexadecimal marker, 3D object detection with features (marker-less), edge based detection, infra-red, and template matching.

### II.1. Colour Based Tracking

This approach tracks the object according to its colour and intensity [1]. Although real world objects can be tracked, the drawback is the sensibility to light changes and noise, such as similar colour histograms between the background and the targets (Fig.1).

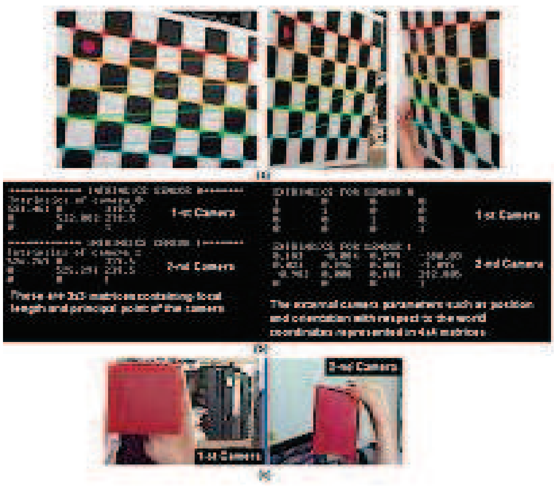
(a)
1-st Camera
2-nd Camera
1-st Camera
2-nd Camera
These are 3x3 matrices containing focal length and principal point of the camera
The extrinsic camera parameters such as position and orientation with respect to the world coordinates represented in 4x4 matrices
(b)
2-nd Camera
1-st Camera
(c)

**Figure 1.** Colour based tracking system. (a) Camera calibration with a chessboard pattern, (b) camera intrinsic and extrinsic matrices (c) tracked real world coloured block and its shape and position projected on both calibrated cameras.

### II.2. Hexadecimal Marker Tracking

In an environment where there is continuous change of the light source and there is noise from the background, these markers [2] (Fig.2) can be used with reliability and still having the drawback for the need to register the code in the system and not being natural in a game for children.

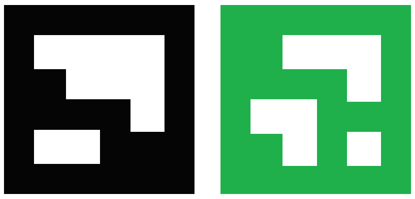

**Figure 2.** Hexadecimal markers with unique ID, two colours for two teams.

### II.3. 3D Object Detection with Features

Detecting the object using features [3] (Fig.3) requires the object to have distinct features like in the case of the robots, but not usable for the coloured blocks where there is no feature because there is only a uniform colour.

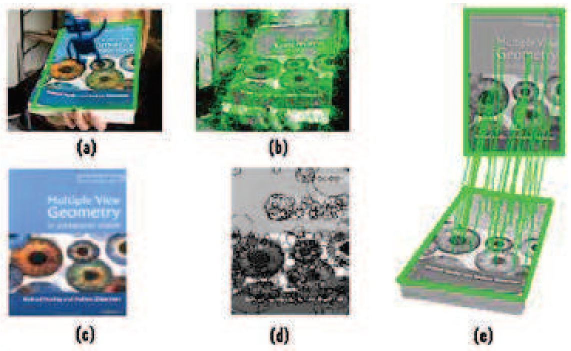

**Figure 3.** 3D Object detection with Features method. (a) camera image, (b) detected features on the image, (c) object model, (d) rendered image (at frontal view) and features sampling, (e) matching features, and pose estimation.

### II.4. Edge Based Detection/Tracking

This is another method based on the shape of the object which requires a uniform background with no noise such as objects with too many edges [4]. A floor of the same colour as a background is required in this case (Fig.4).

### II.5. Infrared Tracking System

Infrared markers and cameras are very reliable with the disadvantage that requires many tools, sometimes difficult calibration and very expensive equipments. It can be used though for other systems calibration.

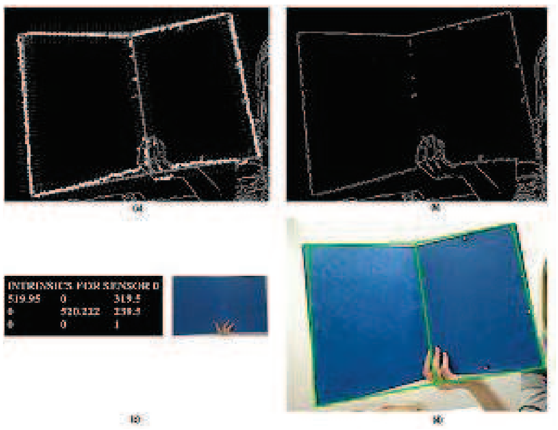

**Figure 4.** Tracking in 3D by means of local LSE optimization (contour-based modality). (a) Intensity edges, (b) Matched model-image edges, (c) Intrinsic parameters of the camera model, and 3D articulated object to be tracked, (d) Output estimate of the Kalman Filter.

**Figure 5.** Tracked objects in Virtools from the OpenTL application running at the same time.

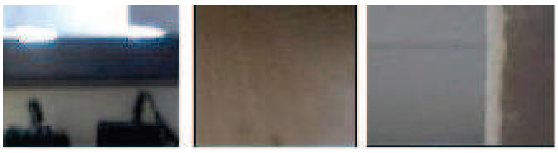

**Figure 6.** Selected textures from the camera to be tracked

### II.6. Template matching

We can track the robots using template matching [5], based on distinguishable features. This is less sensible to noise therefore more precise. Here (Fig.8 and Fig.9) is an example of template matching.

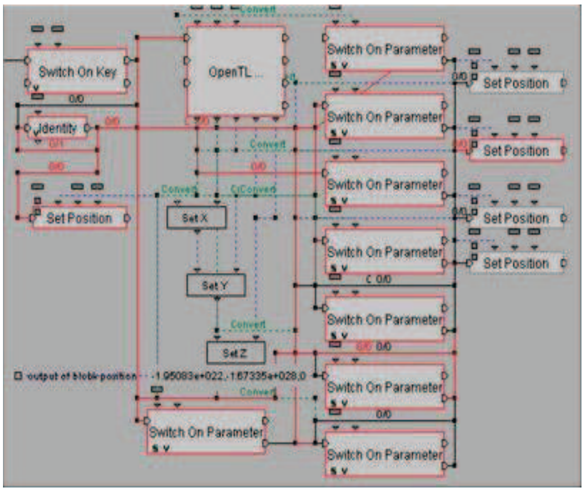

**Figure 7.** The Block diagram of the OpenTL Script

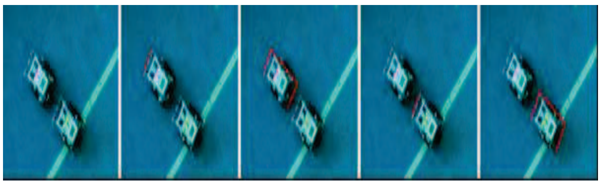

**Figure 8.** Selecting the robots.

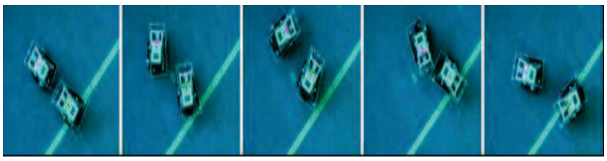

**Figure 9.** Tracking the robots.

## III. EXPERIMENT, CHOSEN METHOD, TOOLS AND ARCHITECTURE

We have now three available tracking systems; coded marker, infrared, and marker-less tracking. Initially we used the Augmented Reality Toolkit [6] and the Advanced Realtime Tracking (A.R.T.) [7] for infra marker tracing, Allied Vision Technologies (AVT) camera for machine vision with FireWire and Gigabit Ethernet (GigE) interface for coded markers. Ubitrack [6] was used to create the Spatial Relationship Graph (SRG) and Trackman [8] for calibration and tracking. Then we used OpenTL [9] for real world 2D/3D object (marker-less approach) tracking. We developed a plug-in to interface OpenTL to 3DVIA Virtools [10], where the game engine was designed. The plugin for virtools retrieves pose data from Ubitrack through a custom network port, while OpenTL runs inside virtools, and a NXT plugin communicates with the robots through the Bluetooth port. The components are connected in a multiplayer Client/Server architecture.

| ObjectNr. | BlobNr. | X poz. | Y poz. | Z poz. |
|---|---|---|---|---|
| 1 | 1 | 14 | 12 | 2 |
| 2 | 1 | 10 | 11 | 6 |
| 1 | 2 | 5 | 4 | 4 |
| 2 | 2 | 6 | 3 | 4 |
| 3 | 2 | 7 | 2 | 4 |

**Table 1.** Output of OpenTL code.

### III.1. OpenTL Plugin Development

We built a plug-in for OpentTL inside Virtools in a direct and straightforward approach. The OpenTL tracking code is called from the Virtools Plug-in functions. Therefore variables are easily accessible from both codes. As a result you can see the coloured circles on the right (Fig.5) that show the tracked objects which resemble the selected textures (Fig.6). The animated objects on the left (Fig.5) are given the 3D position in space of the circles on the right (Fig.5). The OpenTL building block (Fig.7) has five outputs; two object identification parameters, and three pose coordinates. The element “Switches on Parameters” triggers an output if the object identification parameters match. Once the output is triggered, the connected “Set-Position” element enables the X, Y, Z coordinates received form the OpenTL block to be sent to the connected 2D/3D object of the Virtools scene (Tab.1 and Tab.2). There is no external port connection, all the code runs within the same plug-in and there is no need for external tools.

### III.2. Detection Algorithm

1. Shape Appearance – Model Colour; a class which is used to store the shape-appearance of an object. It gives information about the material, size and form of the object. 2. Colour Segmentation method (Histogram); segments the image into different regions and then searches for the colour that most matches the region. The colour with the highest probability is the colour of the wanted region. 3. Blob (Connected Components) Detection – Multi Target; a blob is a connected component, which combines the detected pixels of the selected image together, and then gives the position of the centre of the image. 4. Discard Too Small detected blobs; if the blob is too small (not enough pixels) then it is discarded because probably it was not the one selected. 5. Get Blobs in a target which contains the pose translation parameters; all the blobs are added in a target vector to be referenced for later manipulation. 6. Possible step to be added – Tracking; so far all that was used is image processing but using OpenTL modules. OpenCV [11] could have been enough for the image processing level, but while using the 3D model tracking library, it is possible to track the object by identifying them with an ID.

| Plugin Input | Plugin Output | | | | | Moving the Objects | | |
|---|---|---|---|---|---|---|---|---|
| Connect | ObjectNr. | BlobNr. | X poz. | Y poz. | Z poz. | Switches Parameters Input Combination | | Update Position to Object Nr.: |
| 1 | 1 | 1 | 14 | 12 | 2 | 1 | 1 | 1 |
| 1 | 2 | 1 | 10 | 11 | 6 | 2 | 1 | 2 |
| 1 | 1 | 2 | 5 | 4 | 4 | 1 | 2 | 3 |
| 1 | 2 | 2 | 6 | 3 | 4 | 2 | 2 | 4 |
| 1 | 3 | 2 | 7 | 2 | 4 | 3 | 2 | 5 |

**Table 2.** Data flow for script in Fig.7. Coordinates X, Y, Z are in cm.

| | Minimum | Recomended |
|---|---|---|
| Processor | Intel 2 GHz Single Core | Intel > 2 GHz Core Duo |
| Memory | 1 GB | 2 GB |
| Graphics Card | NVIDIA GeForze 8600 | NVIDIA GeForce 9800 |
| Operating System | Windows Xp/Vista, Linux | |
| Free Disk Space | 300 MB | |
| Internet Connection | For installation of tools | |

**Table 3.** Minimal requirements for OpenTL.

## IV. RESULTS AND DISCUSSIONS

The AVT camera needs to be in a distance over 2m to have enough field of view to detect the markers. The used camera has no significant distortion error but small field of view, and the used markers need to be registered in the system as a Hexadecimal code. The A.R.T. system requires the use of Infra markers which are fragile and have a cost around 10 EUR each. For each track-able object, a minimum of three infra markers attached to the object is needed. Virtools receives the pose data from Ubitrack through network at a port specified from us. On the very first use, trackman should calibrate a minimum of three infrared cameras, that should be synchronized with ubitrack at the same network ports, and we should create the SRG scene in ubitrack to track the infra and hexadecimal markers. On our other proposed approach which uses only OpenTL for Virtools, there can be used coloured blocks instead of hexadecimal markers with no need to register so many different markers, and simpler cameras by setting the distortion parameters. The detection of the blocks is done via multi target blob detection allowing many different colours in the field and numbering them while identifying their category (for two teams, only two colours need to be registered). A.R.T. can be avoided together with its complicated calibration, many tools dependencies, expensive equipments, and network connection requirements, if OpenTL is used to track the robots. For this case the approach is to use template matching as a tracking method. The same camera which is used for the coloured blocks can be used and the minimal parameters are in Tab. 3. This method uses a Bayesian tracker with the Extended Information Filter (EIF) [12] and normalized cross-correlation (NCC) [13] for target loss detection.

## V. CONCLUSION

We have studied the common methods used in computer vision, augmented reality, tracking algorithms and have chosen the best approach for the given problem. In some cases we have combined two or more approaches together to have a robust and flexible system. Our main goal was to have a system that could be easily calibrated, reliable, less dependable on third party tools, user friendly, fast to manage and use, and achievable with a lower budget. We have used different methods for different kinds of targets while studying the objects characteristics such as colours, features, shape and motion. For each feature we have used a specific method that takes advantage the most of the certain characteristic. To boost the performance and minimize the error we have used specialized trackers and added extended filters. Our method is straight forward and fully integrates within the code of other specialized application tools. As a future work, we want to integrate our code in more application tools and boost reliability and robustness.

## VI. ACKNOWLEDGMENTS

This work had support from the Chair of “Robotics and Embedded Systems”, and “Computer Aided Medical Procedures & Augmented Reality” of the Technical University of Munich which provided us with all the necessary tools and literature. A special thank goes to PhD. Giorgio Panin with his help using OpenTL, Dipl.-Inf. Christian Waechter with his help for developing the plug-in, M.Sc. Conor Haines M.Sc. Nima Shahriary M.Sc. Babak Gholami for developing the game, and M.Sc. Artasches Mkhitaryan for his support in tracking.

## VII. BIBLIOGRAPHY

1. G. Panin, E. Roth, and A. Knoll (2008) Robust contour-based object tracking integrating color and edge likelihoods. International Workshop on Vision, Modeling and Visualization (VMV), Konstanz, Germany.
2. H. Kato and M. Billinghurst (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. IWAR'99, IEEE CS, San Francisco, CA, USA.
3. H. Bay, T. Tuytelaars, and L. Van Gool1 (2006) SURF: Speeded Up Robust Features. Computer Vision–ECCV Springer.
4. A. Yilmaz, X. Li, M. Shah (2004) Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. IEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 26, No. 11.
5. G. Panin, A. Knoll (2008) Mutual Information-Based 3D Object Tracking. International Journal of Computer Vision.
6. J. Newman, M. Wagner, M. Bauer, A. MacWilliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, G. Klinker (2004) Ubiquitous Tracking for Augmented Reality .International Symposium on Mixed and Augmented Reality (ISMAR). Arlington, VA, USA.
7. Advanced Realtime Tracking (A.R.T). July 20 2010. < http://www.est-kl.com/fileadmin/media/pdf/A.R.T/Flyer.pdf >
8. P. Keitler, D. Pustka, M. Huber, F. Echtler, G. Klinker (2010) Management of Tracking for Mixed and Augmented Reality Systems. The Engineering of Mixed Reality Systems, E. Dubois, P. Gray, L. Nigay (eds.), Human-Computer Interaction Series, Springer Verlag, 2010.
9. OpenTL. July 30 2010. <www.opentl.org>
10. 3DVIA Virtools. July 30 2010. www.virtools.com
11. OpenCV. July 30 2010 http://sourceforge.net/projects/opencv/ >
12. Giorgio Panin, Model-based Visual Tracking: The OpenTL Framework, New Jersey and Canada, John Wiley and Sons, 2011, p. 173.
13. J.P. Lewis, Fast normalized cross-correlation (1995) Vision Interface -Citeseer.