A Distributed Many-Camera System for Multi-Person Tracking

Claus Lenz[1], Thorsten Röder[1], Martin Eggers[2], Sikandar Amin[2], Thomas Kisler[2], Bernd Radig[2], Giorgio Panin[1], and Alois Knoll[1]

[1] Robotics and Embedded Systems Lab, TUM
{lenz,roeder,panin,knoll}@in.tum.de
[2] Intelligent Autonomous Systems Group, TUM
{eggers,amin,kisler,radig}@in.tum.de

Abstract. This article presents a modular, distributed and scalable many-camera system designed towards tracking multiple people simultaneously in a natural human-robot interaction scenario set in an apartment mock-up. The described system employs 40 high-resolution cameras networked to 15 computers, redundantly covering an area of approximately 100 square meters. The unique scale and set-up of the system require novel approaches for vision-based tracking, especially with respect to the transfer of targets between the different tracking processes while preserving the target identities. We propose an integrated approach to cope with these challenges, and focus on the system architecture, the target information management, the calibration of the cameras and the applied tracking methodologies themselves.

1 Introduction

1.1 Related Work

Intelligent camera surveillance is commonly employed for both security purposes as well as for smart rooms, which can autonomously react to perceived situations. Such systems can be found operating in real-time or focusing on the post-processing of previously acquired video data. Several surveys including Valera et al. [13] (with an emphasis on distributed systems) and Šegvić et al. [11] give a good overview of state-of-the-art techniques in that field.

A multi-agent-based approach is presented by Patricio et al. [7]. Smart rooms also frequently employ visual tracking, such as Lanz et al. [4]. Teixera et al. [12] present a camera sensor network for behavior recognition using address-event image sensors and sensory grammars in an assisted living environment. Other powerful approaches using smart-cameras with onboard processing that directly deliver data instead of images are presented by Rinner and Wolf [9] or in Hengstler et al. [3] with an eye on application oriented design of the sensor network. A related approach, also using color information and Monte-Carlo filtering using distributed cameras is described by Yamasaki et al. [14].

Precedents prove to be hard to find with respect to the large-scale and the real-time operation as presented in this article.

1.2 Overview of the System

Apartment Mock-Up On an experimental area of approximately 100 square meters, a kitchen and a living room have been arranged and furnished accordingly (see Figure 1), in which robots and humans can move freely. If we consider the exemplary scenario of robots serving beverages to humans, several challenges arise due to the fact that different people can order specific drinks. Therefore, a robust and fast way to estimate the position of humans is required, including a unique identification in order to be able to serve the desired beverage to the correct person even if the person moves along while fetching the drink itself. This information can also be useful to facilitate other human-robot joint activities including teaming, task-distribution among robots, or natural motion planning. For a comprehensive report on related research conducted on this mock-up, please refer to Brščić et al. [1].

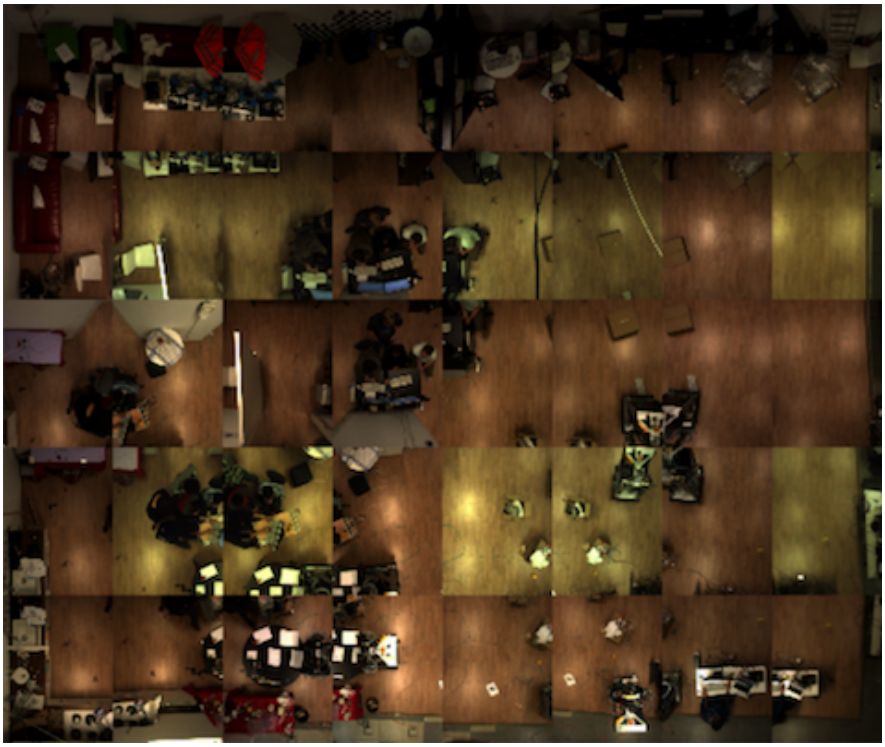

Fig. 1. View of the 40 partly overlapping camera views observing the apartment mock-up. Different illumination conditions are caused by object shadows, while varying chromatic appearances result from using cameras with different specifications and specular reflections are caused by light sources behind the ceiling-mounted cameras.

Hardware Installation The installed hardware consists of a total of 40 color cameras, each having a native resolution of 1024×768 pixels at a rate of 28 to 30 frames per second. All cameras are Ethernet-connected using the GigE-Vision communication standard as described in [2], and are installed on a metal scaffolding mounted at the ceiling. The camera fields of view (FOVs) cover the whole area facing top-down. This setup achieves a total FOV redundancy of approximately 75% at a height of 1.7 m, which according to Ogden et al. [5] is the approximate average height of an adult person. The cameras are grouped in threes and pairs respectively to form 14 camera groups, each of which is in turn linked via a Gigabit Ethernet (GigE) switch to a diskless processing node, where image capturing and processing itself takes place. A single server computer manages the diskless node network, and hosts the server applications described in Section 3.2.

For robustness and load-balancing reasons, adjacent cameras are assigned to different camera groups. This helps to compensate for the observable fact that human beings tend to flock together in social scenarios, rather than distribute evenly over the surveyed area. Besides the drawbacks of a relatively high amount of cameras being required to cover the area, and the requirement of managing frequent transfers of targets between camera FOVs, this specific camera setup offers the following advantages:

- The image resolution of each local FOV shows a high quality, compared to solutions using less cameras.
- All cameras are almost co-planar. This allows for application of the same tracking techniques and assumptions for the whole covered area, which reduces the system and algorithmic complexity.
- Under the assumption that people do not climb over each other, a full-body mutual occlusion is almost impossible to occur.
- Since the camera transformations w.r.t. a common world frame are known, the hand-over regions (shared FOV regions in which a target transfer may occur) can be defined and evaluated easily.
- Because of the local distribution of computational power, the number of simultaneously tracked people can be increased or the computational power can be shared in a scalable and distributed way with other computationally intensive approaches e.g. for gesture recognition or or activity analysis.

2 Single Camera Person Tracker

2.1 Person Detection

New persons are detected as they enter the FOV of a camera using a foreground-background segmentation (adaptive mixture of Gaussians) [15], followed by a blob clustering [10] and a data association method. No further a priori information about the person's appearance including color or texture clues is used up to now. Every blob that results from a foreground region is classified as a human by analyzing the area size and the aspect ratio of the outer dimensions.

The center of mass of a blob as x and y position on the sensor projection layer is subsequently upgraded to a 3D translatorial pose of the human target in the world frame using the extrinsic camera parameters. A virtual ray given by the focal point of the camera and the computed mass center of the blob is casted and then intersected with the ground floor, which has a z-coordinate of 0. Minor position errors due to perspective distortions and the fact that the height of the detected person can not be estimated using top-down mounted cameras are corrected in the first tracking step.

In addition, the person detection method is employed for validation and target association during the target transfer process described in Section 3.2.

2.2 Model Building

The 3D model approximating the human shape consists of a simple cylinder, roughly corresponding to human size in real-world coordinates. Once the overall scale has been computed from the detected blob, the relative proportions (location of the head with respect to the torso, relative size, etc.) are fixed according to an average model of the human body. The statistical color model is obtained by collecting the image pixels for the respective blob, in a 3D histogram in HSV color space. Different bin sizes are used, in order to give more importance to the color attributes rather than the intensity values, which are illumination-dependent. In this case, a robust combination of 8 bins for hue, 8 bins for saturation and 4 bins for value were used. In order to collect only pixels that do belong to the person, the background segmentation image is treated as a mask.

2.3 Multi-target MCMC filter

The color model of a person is now used in order to instantiate a new *target*, to which a unique ID number is assigned, and that will be tracked across the image sequence by this camera, until the person leaves the camera's FOV.

Tracking operates on a pre-defined set of degrees of freedom, which for our rigid 3D shape model is defined solely by the 2D translation on the floor (x, y coordinates). Therefore, the state vector of the $i-th$ target is given by $s^i = (t^i_x, t^i_y)$.

The tracking methodology basically consists in matching the reference color histogram to the current image, underlying the projected shape of the person. By using a calibrated camera model, we also take into account perspective effects while computing the *silhouette* in the camera image. These effects have a high impact for our setup, since the relative distance between the camera and the person is comparable with the depth extension of the target (i.e. the height of the person). Therefore, they cannot be neglected, especially for people in the peripheral view field.

In order to estimate the state of each person on the image sequence, a Bayesian Monte-Carlo tracking approach is used, described in more detail in [6]. This methodology consists of a particle filter, which maintains the global system

state, $s = (s^1, ..., s^m)$ of the m currently active targets in the scene, by means of a set of hypotheses s_h (or *particles*), that are updated from frame to frame by means of Markov-Chain Monte-Carlo sampling (MCMC). In particular, the Markov-Chain generation proceeds by iterating (for each particle $n = 1, ..., N$) two steps, that correspond to the *Metropolis-Hastings* algorithm.

The efficiency of the MCMC formulation is due to the fact that we update a *single target* i at a time (randomly chosen), which results in the computation of the proposal ratio only for this target $P(s_{i,t} | s_{i,t-1})$.

3 Upgrading to a Many-Camera Set-up

3.1 Calibration of Cameras

The calibration procedure is performed in two steps, with the intrinsic camera parameters being determined independently on all cameras in the first step. The second step then aims at determining the extrinsic parameters of all cameras w.r.t. the global world origin. To ensure optimum inter-camera consistency of calibration, which is an important issue regarding the transfer of tracked persons from one camera to another, we make additional use of an infrared tracking system consisting of six Visualeyez VZ 4000 [8] tracker bars, that measure accurately with a root mean square error of below 0.5 mm. A standard calibration pattern was enhanced with 4 infrared diodes, so that the infrared tracking system is able to determine the pose of the calibration pattern accurately at any position within the scene. The obtained pose information was transmitted to the diskless clients managing the cameras, with the cameras being used to record image sequences simultaneously, which can then be used to determine the extrinsic parameters via prevalent calibration methods.

3.2 Management of Tracking

The tracking algorithm described in Section 2 is working independently and asynchronously for each camera without knowledge about other cameras or synchronization mechanisms on the client sides. Using this design principle, a server application handles the centralized management of the tracking results and takes care of the transfer of tracked targets between camera FOVs (and consequently, processing nodes). The advantages of this approach lie with its full scalability and the lack of need for synchronization. To realize the approach, global modules for registration of the single trackers, a global display module, a module for the generation of unique target IDs and for the management of the transfer of targets between tracking clients were implemented. All of these modules are running on the server computer of our hardware installation, while the client applications, responsible for detection and tracking, run on the diskless processing nodes, as described in Section 1.2.

Global Registration The global registration instance of the system possesses a global representation of all connected cameras and processing nodes. This includes knowledge about 1) the unique IDs of the cameras and their connection to specific processing nodes, 2) the extrinsic and intrinsic parameters of each camera, 3) the possibility to enter the surveyed area through a specific camera FOV, 4) the expected sizes for camera images for the streaming process and 5) the expected size of a connected display.

The aforementioned knowledge about the setup is loaded to the system and can be exchanged, in order to adapt the system to other scenarios and setups. The client applications, which register their camera in the registration server, obtain the expected size of images to be sent to connected displays in return. Furthermore, the client applications request the global knowledge about the surveyed area and use the obtained information to instantiate their system with the correct intrinsic and extrinsic camera parameters. This leads to the advantage that no client application has to keep local information on the setup. Therefore, if the setup is changed e.g. by adding new cameras, moving cameras or recalibration, a simple restart of the registration and the client applications updates the whole system.

Global ID Generation In a distributed, asynchronous tracking system with completely independent tracking processes, it is an essential need that the tracked targets keep their identity after a target transfer from one tracking process to another, which occurs if a tracked target switches between camera FOVs. Once the detector module of a client application has found a new target, it can only be introduced and added in the server to the global system state using a uniquely generated ID. Therefore, the client *requests* a new ID from the global ID generator module. Using this ID, the client is able to add the target in the server with its current position in world space. Subsequently, the target's position will be broadcast to connected processes, e.g. those running on the robots in the scenario described above.

All clients have a direct connection to the server to add targets, update the target positions, or to initiate handover targets in case they are leaving the surveyed area. In order to maintain the scalability of the system, the server module does not possess a priori knowledge on the number of client applications. Therefore, a communication channel was created that broadcasts the control commands: a) transfer target: to give a client the responsibility to track a certain target and b) remove target: to release the responsibility of a client to track a certain target. Every client listens to this control channel and reacts only if the mentioned target is within his responsibility (remove target) or if he should take over the tracking (transfer target).

Management Server A challenging task is to decide which client should do the tracking, when to transfer targets, and where to transfer targets to. Performing the evaluation of 40 camera location at every position update (up to 15 Hz) requires very high computational power ($n-1$ comparisons). At this point,

(a) Frame 100 (b) Frame 150 (c) Frame 200 (d) Frame 250

Fig. 2. Example image sequence: joint view of four adjacent cameras tracking two different persons. a) and (b) show the tracking of a single person before and after a hand-over of the person target took place. In the same way (c) and (d) depict the tracking results of two persons, before and after the respective handover situations have occurred.

it is possible to utilize the extrinsic and intrinsic calibration parameters of the cameras, that are stored in the registration server. After fetching this information, a quad-tree is built up, dividing the scene in quarters: In the first step each center point and FOV of all cameras are projected on the floor. Now, the tree is iteratively refined dividing each node in 4 sub-nodes (areal quarters) until each node has only one camera left.

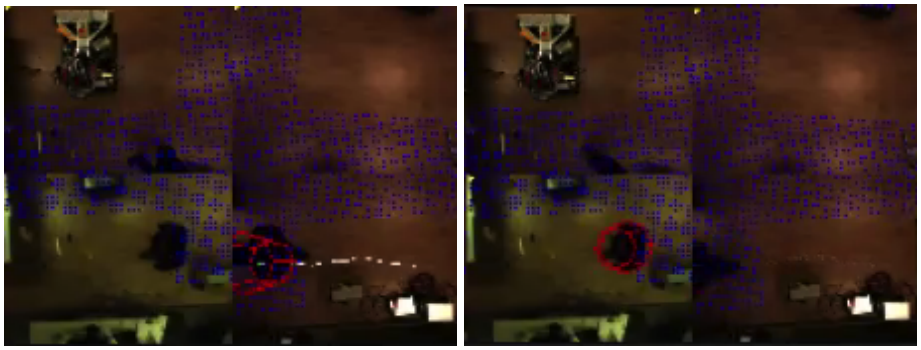

Fig. 3. Target transfer example sequence: Joint view of four adjacent cameras. Areas marked by blue dots represent valid target transfer regions between adjacent FOVs. The dots result from the application of space discrete (in world coordinates) tests, which consist of evaluation of the target transfer tree that was pre-computed from the respective camera projection matrix. The red cylinder represents the tracked person that walks through adjacent FOVs.

The tree nodes contain all the important information to select the optimal camera to transfer a target to. This includes the projection center of the camera on the ground floor, the FOV, the camera index, and the hostname of the client to which the camera is connected. After each update of the target's position, the pre-calculated tree can be traversed efficiently to find the next host and the

next camera index. Figure 3 depicts 4 adjacent cameras with blue dots which represent valid target transfer points between cameras. These points are the result of a space discrete (in world coordinates) test using the pre-calculated target transfer tree.

4 Results

The system was implemented using the proposed tracking and communication architecture. Subsequently, the system was evaluated by tracking persons exhibiting high variance w.r.t. their height, appearance and motion habits, as well as under different illumination conditions during the respective days. This was done in order to test the applicability of the rather coarse model assumptions, being the average height of people (1.7m), the applicability of the rigid non-articulated cylindrical shape representation and the evaluation of color statistics from the top view.

As depicted in Figure 1, the client-server architecture is able to stream miniaturized camera images to multiple connected display processes, which may be distributed over the internet. Figure 2 depicts the joint view of four adjacent cameras simultaneously tracking two individual persons. (a) and (b) shows the tracking of one person including a hand-over procedure. (c) and (d) depict the tracking of two persons. Figure 3 illustrates the pre-computed target transfer regions in a joint view of four adjacent cameras. The transfer regions can be estimated by evaluating the quad-tree which is computed using the extrinsic camera parameters.

Regarding the the coarse 3D cylindrical person model, several tracking approaches were tested. E.g. Figure 4 shows the comparison of two different evaluation strategies for the sampling of the color statistics, comparing the accuracy achieved and required computational time. The first strategy consists of the sampling of pixels within the underlying rotated bounding box of the cylindrical model. The second strategy projects the 3D cylindrical model into the sensor image and thus computes the full and exact shadow, which is used as a mask before pixel sampling takes place. Based on these results, we decided to adopt the full shadow approach, since the improved accuracy outweighs the drawback of a slightly higher computational cost in our setup.

5 Conclusion and Future Work

In this paper, a flexible, scalable and modular many-camera system for simultaneous tracking of multiple persons using natural features was presented. The approach was realized and evaluated using an apartment mock-up, sensorized by 40 GigE cameras which fully cover its approximately 100 square meters. Given this large amount of cameras, distribution of the computational processing among multiple computers is required, which is addressed using 14 diskless

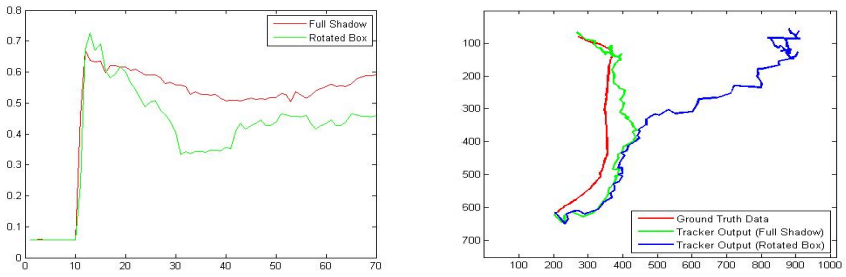

Fig. 4. Using the full shadow of the cylindric human shape approximation to compute the measurement results in a significant gain in accuracy while requiring only slightly more computation time. *left*: Time needed to evaluate measurement using rotated box or the full shadow. X axis denotes the number of frames, Y axis denotes the time in seconds, running three tracking modules simultaneously. *right*: Accuracy using rotated box or full shadow. X and Y axes denote pixel positions within the image.

client processing nodes operating up to three cameras each. A functional system for the management of target detection, target tracking and target transfer between processing nodes was presented.

Future work includes improving the detection step by application of a more robust classification of persons, rather than assuming every foreground pixel to be belonging to a human. This will help avoid erroneous detection alarms. Furthermore, the degrees of freedom tracked can be extended by including the estimation of persons' rotation angles. In our application context, this additional information would allow robots to intentionally approach persons from specific directions, as well as facilitating the evaluation of psychological experiments within the surveyed area.

6 Acknowledgements

This work was partly supported by the cluster of excellence *CoTeSys – Cognition for Technical System* (`http://www.cotesys.org`) within the projects *ITrackU – Image-based Tracking and Understanding* and *MuJoA – Multi Joint Action* funded by the DFG.

References

1. Brščić, D., Eggers, M., Rohrmüller, F., Kourakos, O., Sosnowski, S., Althoff, D., Lawitzky, M., Mörtl, A., Rambow, M., Koropouli, V., Hernández, J.M., Zang, X., Wang, W., Wollherr, D., Kühnlenz, K., Mayer, C., Kruse, T., Kirsch, A., Blume, J., Bannat, A., Rehrl, T., Wallhoff, F., Lorenz, T., Basili, P., Lenz, C., Röder, T., Panin, G., Maier, W., Hirche, S., Buss, M., Beetz, M., Radig, B., Schubö, A., Glasauer, S., Knoll, A., Steinbach, E.: Multi Joint Action in CoTeSys - setup and

challenges. Tech. Rep. CoTeSys-TR-10-01, CoTeSys Cluster of Excelence: Technische Universität München & Ludwig-Maximilians-Universität München, Munich, Germany (June 2010)
2. Chamberlain, G.: GigE Vision: standard route to video over IP. Industrial Ethernet Book (33), 35 (July 2006)
3. Hengstler, S., Aghajan, H., Goldsmith, A.: Application-Oriented Design of Smart Camera Networks. In: Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on. pp. 12–19 (2007)
4. Lanz, O., Chippendale, P., Brunelli, R.: An appearance-based particle filter for visual tracking in smart rooms (2007)
5. Ogden, C., Fryar, C., Carroll, M., Flegal, K.: Mean body weight, height, and body mass index, united states 1960–2002. Advance data from vital and health statistics (347) (2004), `http://www.cdc.gov/nchs/data/ad/ad347.pdf`
6. Panin, G., Röder, T., Knoll, A.: Integrating robust likelihoods with monte-carlo filters for multi-target tracking. In: International Workshop on Vision, Modeling and Visualization (VMV). Konstanz, Germany (Oct 2008)
7. Patricio, M.A., Carbo, J., Perez, O., Garcia, J., Molina, J.: Multi-agent framework in visual sensor networks. EURASIP Journal on Advances in Signal Processing(Print) 2007(7) (2007)
8. Visualeyez VZ 4000 (2009), `http://www.ptiphoenix.com/VZmodels.php`
9. Rinner, B., Wolf, W.: An introduction to distributed smart cameras. Proceedings of the IEEE 96(10), 1565–1575 (2008)
10. Rocha, L., Velho, L., Carvalho, P.C.P.: Motion reconstruction using moments analysis. In: SIBGRAPI '04: Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium. pp. 354–361. IEEE Computer Society, Washington, DC, USA (2004)
11. Segvic, S., Ribaric, S.: A software architecture for distributed visual tracking in a global vision localization system. p. 365 ff (2003)
12. Teixeira, T., Lymberopoulos, D., Culurciello, E., Aloimonos, Y., Savvides, A.: A lightweight camera sensor network operating on symbolic information. In: Proceedings of 1st Workshop on Distributed Smart Cameras, Held in Conjunction with ACM SenSys. Citeseer (2006)
13. Valera, M., Velastin, S.: Intelligent distributed surveillance systems: a review. IEEE Proceedings Vision, Image and Signal Processing 152(2), 192–204 (2005)
14. Yamasaki, T., Nishioka, Y., Aizawa, K.: Interactive retrieval for multi-camera surveillance systems featuring spatio-temporal summarization. In: MM '08: Proceeding of the 16th ACM international conference on Multimedia. pp. 797–800. ACM, New York, NY, USA (2008)
15. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition. vol. 2 (2004)