# REAL-TIME 3D MULTIPLE HUMAN TRACKING WITH ROBUSTNESS ENHANCEMENT THROUGH MACHINE LEARNING

Suraj Nair[1], Emmanuel Dean[1] and Alois Knoll[1]
[1]*Robotics and Embedded Systems, Technische Universität München, Germany*
{*nair,dean,knoll*}*@in.tum.de*

Keywords: Visual Tracking

Abstract: This paper presents a novel and robust vision-based real-time 3D multiple human tracking system. It is capable of automatically detecting and tracking multiple humans in real-time even when they occlude each other. Furthermore, it is robust towards drastically changing lighting conditions. The system consists of 2 parts, 1. a vision based human tracking system using multiple visual cues with a robust occlusion handling module, 2. a machine learning based module for intelligent multi-modal fusion and self adapting the system towards drastic light changes. The paper also proposes an approach to validate the system through zero-error ground truth data obtained by virtual environments. The system is also validated in real-world scenarios.

## 1 INTRODUCTION

This paper presents a novel real-time 3D multiple human tracking system with the primary focus on robustness enhancement through machine learning. It is a vision based system, capable of automatically detecting human targets. After detecting the targets, the trajectory of each detected target is tracked with a 3D pose in real-time. The system has an ability to resolve target occlusions in real-time and maintain individual trajectories provided the targets do not leave the designated tracking area. The occlusion handling system resolves mutual occlusion between the targets and serves as an important tool for robust tracking under circumstances of mutual occlusion in multiple camera views when the tracking scene consists of many targets.

A machine learning based approach is introduced to train and classify lighting conditions in the tracking environment. Lighting conditions being highly influential in robust tracking, its classification helps the tracker to take important decisions to maintain the robustness. On the basis of this classification, intelligent multi-modal fusion of two visual cues is performed. Depending on the current situation, the optimal weights in which the visual cues are fused in order to achieve the desired robustness is computed. This approach improves the robustness of the tracking system in terms of self adaptability to changing tracking conditions. Although the machine learning based lighting conditions classification is useful in multi-modal fusion, it finds an important use case in robust pre-processing of camera images such as background segmentation. Sudden changes in lighting conditions can be detected and the background model can be updated using this approach. The background model update is not trivial in presence of foreground targets. This paper introduces an approach to identify such situations and update the background model in presence of foreground targets under drastic changes in lighting conditions. It introduces a novel ground truth generation method through simulation due to un-avaliability of standard datasets like in pedestrian detection (give ref). Further, unlike existing systems such as (give refs) which propose robustness to illumination but do not provide experiments in that specific aspect, we conduct distinct experiments to prove that the system if robust to lighting conditions. A real-world application involving a Industrial robot and changing lighting conditions is also presented. Due to un-availibility of a common benchmarking platform for stereo multiple human trackers, extensive quantitative comparison with other systems is not presented as these systems do not benchmark their methodology in unified way.

## 2 PRIOR ART

Several systems have been developed to track humans using multiple cameras in both un-calibrated

and stereo-calibrated fashion. (Santos and Morimoto, 2011) provides a systematic mention of approaches (Eshel and Moses, 2008), (Fleuret et al., 2008), (Hu et al., 2006), (Kim and Davis, 2006), which use uncalibrated cameras and homography to perform people tracking. (Santos and Morimoto, 2011) use of a combination of the perspective geometry and the homography constraints from each camera view. This information is fused to check for the presence of people in each camera view.

(Soto et al., 2009) present another multi-target tracking system using multiple cameras. Their approach is focussed on a self-configuring camera network consisting of cameras with pan-tilt. The cameras keep track of the targets and adjust their parameters with respect to each other.

(Khan and Shah, 2008) present a slightly different approach of multi view tracking of people. They use information in combination from all views which is projected back to each camera view and a planar homographic occupancy constraints for likelihood computation. This is used to resolve occlusions and model scene clutter using the Schmieder and Weathersby clutter measure (Weathersby and Schmieder, 1984).

Multiple people trackers (Haritaoglu et al., 1998; Siebel and Maybank, 2002; Isard and MacCormick, 2001), have the common requirement of using a very little and generic off-line information concerning the shape and appearance of the person, while building and refining more precise models (colour, edges, background) during the on-line tracking task.

(Francois et al., 2006) combines target occupancy in the ground plane with colour and motion models to track people in continuous video sequences. This approach requires heuristics to rank the individual targets to avoid confusing them with another.

(Focken and Stiefelhagen, 2002) introduces a system for tracking people in a smart room. They use a calibrated camera system within a distributed framework. Each camera runs on a dedicated PC. The detected foreground regions are sent to a tracking agent which computes the locations of people from the detected regions.

Another work is presented by (Cai and Aggarwal, 1996). They use grey scale images from multiple fixed cameras to perform the tracking. They use multivariate Gaussian models to estimate closest matches of humans between consecutive image frames. The system proposed by (Dockstader and Tekalp, 2001) is aimed at tracking human motion with key focus on occlusions. Each camera view is independently processed on a individual computer. Within the Bayesian network, the observations from the different cameras are fused together in order to resolve the independent relations and confidence levels. An additional Kalman filter is used to update the 3D state estimates.

(Chang and Gong, 2001) present a multi camera people tracking system using Bayesian filtering based modality fusion. They employ a modality fusion technique based on the approach by (Toyama and Horvitz, 2000).

(Hayashi et al., 2004) present a stereo camera based people tracking system. They address the problem of tracking in rooms where the camera cannot be mounted high enough. They propose a method to project the 3D voxels on the tracking floor and thereby track their peaks for the purpose of ignoring view changes due to low camera mounting.

(Zhao et al., 2005) presents another stereo cameras based people tracking system. It is a real-time system to track humans over a wide area. A multi-camera fusion modules combines tracks of a single target in all view to a global track.

Considering the state of the art, the primary contribution of this work is made in the form of robustness enhancement through machine learning which makes the system robust to drastic changes in lighting conditions and improves the tracker robustness through intelligent multi-modal fusion of two visual modalities. Another important contribution is a robust occlusion handling system which can resolve multiple occlusion in real time.

# 3 SYSTEM ARCHITECTURE

The 3D multiple human tracking system uses visual information from multiple cameras in order to automatically detect and tracks humans in real time. The detection process operates independent of the tracking allowing detection of new targets when they enter the *tracking area* [1] while the tracker is tracking existing targets.

The target shape is modelled as a 3D rectangular box approximating to the dimensions of a human. The appearance model is generated in the form of a 2D joint probability histogram in all camera views. The target dynamics is modelled using the constant white noise acceleration $(CWNA)$ motion model. The tracker uses a bank of SIR particle filters (Isard and Blake, 1998), working on a 3D motion model, appearance model and optical flow. The particle filter provides the sequential prediction and update of the respective 3D $states = (x, y, z)$. For real-time performance, a global particle set is maintained and dis-

[1] The predefined camera workspace where the cameras can view the targets.

tributed evenly among the bank of particle filters in order to maintain real-time performance.

## 3.1 Tracking Pipeline

Fig. 1 describes the complete pipeline of the tracking system. Each module is discussed in detail in the subsections below.

### 3.1.1 Pre-processing of Sensor Images

Each sensor image undergoes a initial background segmentation step followed by RGB to HSV colour space conversion and optical flow segmentation.

### 3.1.2 On Line Target Detection

This module automatically detects targets when they enter the tracking area by performing a scan along the tracking floor area using the 3D box target model. A target is recorded if the target occupancy is beyond a certain threshold in 2 or more cameras. The target data consists of: 1. Unique *Target ID*, 2. Initial 3D pose, 3. Shape data, 4. Appearance data, 5. Occlusion test information, 6. Current 3D pose, 7. Velocity.

### 3.1.3 Occlusion Testing

It determines if a target is occluded by other targets in a particular camera view. This information is essential during target detection and tracking. These regions are obtained by warping the 3D pose of each target under consideration on to each camera image (id= 0,....,M). Each target is defined by a 3-dimensional container box comprised by 8 vertices

$$V_n(t) = \left\{ v_j \in \mathbb{R}^3 \mid j = 0,1,...,7 \right\} \tag{1}$$

where, $v_j$ is the $j^{th}$ vertex of target shape model defined in Cartesian space for the state $s(t)$. These vertices are projected on each camera as follows:

$$S_n(t) = \left\{ r_j \in \mathbb{R}^2 \mid r_j = K[R \mid T] v_j, \forall\, v_j \in V_n(t) \right\} \tag{2}$$

where, $S_n(t)$ is a set of the projected vertices of the target $n$. $K, R$, and $T$ describe the camera model. Then, we define $d_n(t)$ as the Oriented Bounding Box (OBB) of $S_n(t)$.

$$l_n(t) = \left\{ (x,y) \in \mathbb{R}^2 \mid (x,y) \in d_n(t) \right\} \tag{3}$$

The geometric meaning of $l_n(t)$ is all the pixels within the OBB $d_n(t)$. These pixels are used for the occlusion test.

Fig. 2 illustrates the occlusion test system. This system considers all the targets and computes their occupancies in each camera image and computes the euclidean distance from the camera to each target. The bounding box of target farthest from the camera is computed and rendered first. Once all targets are rendered an overlap test is conducted between the rendered regions to check which targets are occluded.

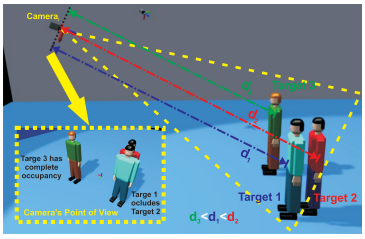


Figure 2: The figure illustrates the occlusion test system.

### 3.1.4 Tracker

Each target is equipped with its own particle filter. The visual modalities used are 2D colour histograms and optical flow. The tracking pipeline is as follows:

*Tracker prediction:* The particle filter generates several prior state hypotheses $s_t^i$ from the previous distribution $(s^i, w^i)_{t-1}$ through a prediction model. In this system the constant velocity model was used,

$$s_t^i = s_{t-1}^i + \dot{s}_{t-1}^i \tau_i + \frac{1}{2} v_t^i \tau_t^2 \tag{4}$$

where, $\dot{s}_{t-1}^i$ is the velocity and is constant, $v_t^i$ is a random acceleration. $\tau$ is the sampling interval.

*Likelihood:* The likelihood is computed on the projected hypothesis in each camera view. The colour matching is computed through a distance measure of the underlying and reference histograms through the Bhattacharyya coefficient (Bhattacharyya, 1943)

$$B_m(q_i(s), q_i^*) = \left[ 1 - \sum_N \sqrt{q_i^*(n)\, q_i(s,n)} \right]^{\frac{1}{2}} \tag{5}$$

The colour likelihood is then evaluated under a Gaussian model in the overall residual

$$P(z^{col} | s_t^i) \propto exp(-\prod_M (B_i^2 / \lambda)) \tag{6}$$

with given covariance $\lambda$.

Similarly, the optical flow distance measure if computed by comparing the projected motion vector of the hypothesis and the underlying motion vectors in the hypothesis region.

$$F_m(f_i(s), f_i^*) = \left[ 1 - \sum_N \sqrt{f_i^*(n)\, f_i(s,n)} \right]^{\frac{1}{2}} \tag{7}$$

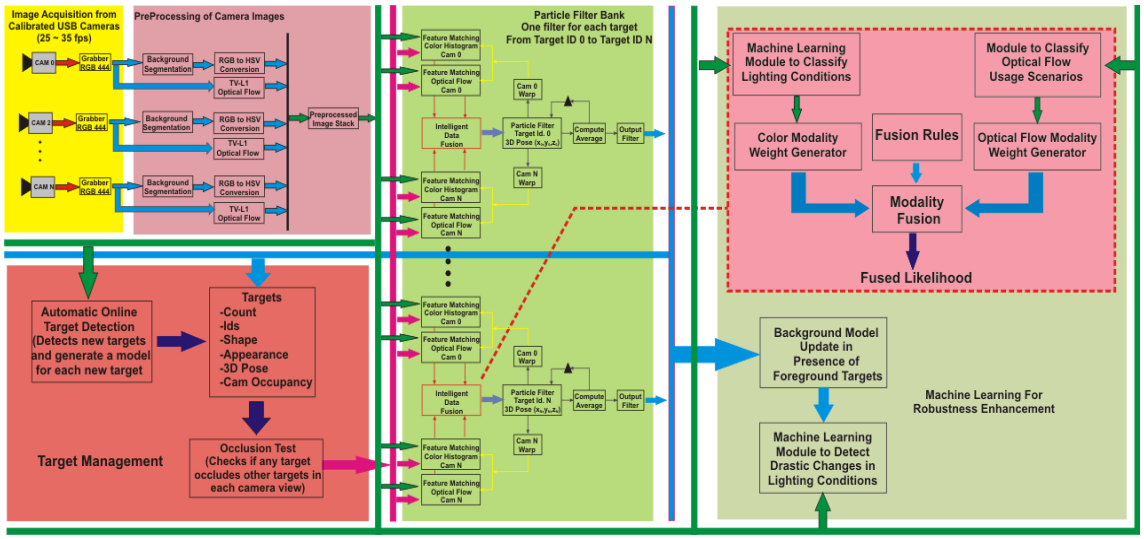


Figure 1: The figure illustrates the block diagram of the multiple human tracking system.

$F_m$ is the resulting distance measure from the optical flow modality for an individual hypothesis. The optical flow likelihood is computed as follows.

$$P(z^{flow}|s_t^i) \propto exp(-\prod_M (F_i^2/\lambda)) \quad (8)$$

*Multi-modal Fusion:* The intelligent multi-modal fusion module described in the next sections, generates the normalized weights $W_{col}$ and $W_{flow}$. The global likelihood for the hypothesis $s_t^i$ is then given by

$$P(z^{global}|s_t^i) = P(z^{col}|s_t^i)W_{col} + P(z^{flow}|s_t^i)W_{flow} \quad (9)$$

*Computing the estimated state:*
The average state $\bar{s}_t$,

$$\bar{s}_t = \frac{1}{N}\sum_i w_t^i s_t^i, \quad (10)$$

is computed and the three components $(\bar{x}, \bar{y}, \bar{z})$ are returned. In order to reduce the jitter in the output, the average pose can be smoothed using an exponential filter.

# 4 Machine Learning to Enhance Robustness

This section introduces an approach to improve the robustness of the system using machine learning techniques.

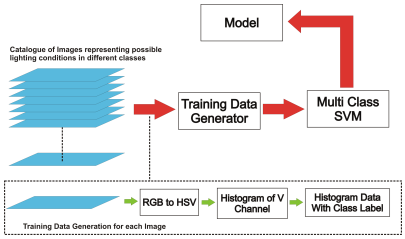


Figure 3: Building blocks of the SVM trainer for lighting conditions

## 4.1 Learning Lighting Conditions

Multi class support vector machines (Crammer and Singer, 2001) can be used as an important tool to learn the different possible lighting conditions that could occur during tracking. They are divided into classes where each class consists of a wide variety of possible lighting conditions. From the learning process, a model is generated which can be further used for online classification of the lighting conditions. Using this information the contribution of the individual modalities towards multi-modal fusion module can be computed.

Fig. 3 illustrates the building blocks of the support vector machine based training module for lighting conditions. It consists of a large set of training samples in the form of images. Each training sample is processed to obtain the training data. Once the training data is available, it is used by the *svm* training module to generate a model based on the classes in which the training data were grouped. The lighting types are *Insufficient lighting*, *Good lighting* and *Saturated lighting* representing classes *Bad* and

*Good*. The three stages in training are, see Fig. 4,

1. *RGB* to *HSV* Colour Space Conversion
2. Histogram Computation: $N$ bin normalized histogram of the $V$ channel is computed representing the intensity distribution.
3. Labelling: A class label is generated through automatic analysis or manual observation. The class label together with the histogram data forms one training data sample for the multi-class support vector machine.

Around 4000 images of each class were used to generate the training data. This makes the total training data set to consists of 12000 data samples. These samples were generated using camera images obtained from the real scene and from 3D simulations of the entire scene, where the lighting conditions could be controlled.

## 4.2 Classification of Lighting Conditions

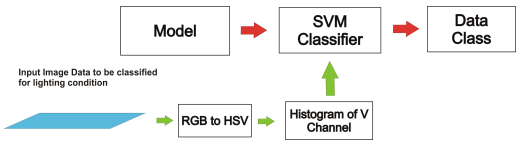


Figure 4: Building blocks of the SVM classifier for a lighting conditions.

Fig. 3 illustrates the building blocks of the support vector machine based lighting conditions classifier which uses the model generated by the *SVM* trainer.

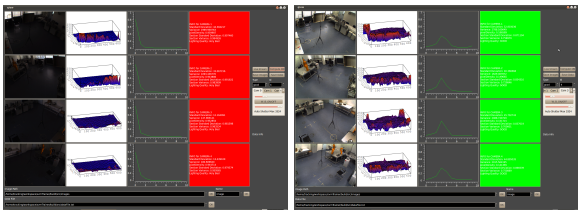

Figure 5: On-line classification of the lighting condition. Red: Bad, Green: Good

Fig. 5 illustrates the test conducted for the on-line classification of lighting conditions. The model is able to classify and associate the current lighting conditions in the camera views to their respective classes. In this experiment, the *SVM* model was trained for three classes of lighting conditions.

## 4.3 Background Model Update in the Presence of Foreground Targets

Updating the background model when light changes. The background model update is trivial in case of an empty scene, but becomes a complex task in the presence of foreground targets in the scene being tracked.

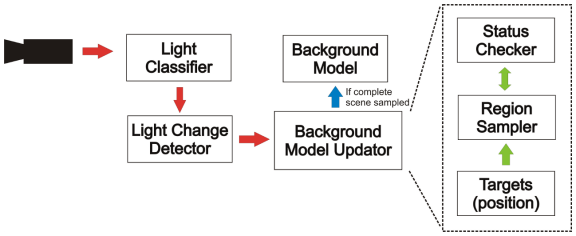


Figure 6: Background model update in presence of foreground targets.

This paper proposes an approach to update the background model under changing lighting condition while foreground targets are present in the scene. It exploits the fact that during the course of tracking the targets will move and expose the regions previously occluded by them. The occluded regions can be included into the background model once they are visible due to target motion. The assumption that the target will move is valid because, if the they do not move, then the tracker only needs to perform a extremely small local search to keep track of the target which does not require information from the background subtracter. Fig. 6 illustrate the Background Model Update Procedure (BMUP) under changing lighting conditions and in the presence of foreground targets. The BMUP comprises of three main parts:

- **Light Classifier:** determines which class the lighting conditions in the current camera image belongs to.
- **Light Change Detector:** continuously reads the classification result from the *Light Classifier* and compares it with the classification results of the previous instance and thereby detects drastic changes in lighting conditions.
- **Background Model Updater:** updates the background model when it is notified about a light change event by the *Light Change Detector*. It uses the target positions, region sampler and the status checker modules. If number of targets $N = 0$, the background model is updated with the image $I_{id}$. If $N > 0$, from each target position the occupancy region $L_{id}$ of each camera (id=0,...,M) is obtained. This is given by:

$$L(t) = \bigcup_{j=1}^{N} l_j(t) \tag{11}$$

where $l_j$ is given by Eq. 3.

This is the area that can not be included in the reference image for the new background model, and needs to be included when exposed. The current area for the reference image is initialised as:

$$D(t_0) = (A \cap L(t_0))^c \tag{12}$$

Then the background image is initialized,

$$I_{ref} = \{I(x,y) \mid x,y \in D(t_0)\} \tag{13}$$

where $A = \{(x,y) \mid x = 1,2,...,width,\ y = 1,2,...,heigh$

The unupdated regions are updated in time when the targets are in motion, thereby exposing the previously hidden regions, this is computed in the next form:

$$h_L(t) = (L(t-1) \setminus (L(t) \cap L(t-1))) \tag{14}$$

where $h_L$ is the new exposed pixels in the current frame. Then the background image is updated using these pixels as follows:

$$I_{ref} = \{I(x,y) \mid x,y \in h_L(t)\} \tag{15}$$

Finally, the current area at time $t$ is updated as below:

$$D(t) = D(t-1) \cup h_L(t) \tag{16}$$

$D(t)$ is updated until $|D(t)| = |A|$.

When the background update process is initiated the tracker suspends the new target detection process. Further, instead of generating the *HSV* image from the background segmented image, the tracker uses a mask to highlight only the local regions surrounding each target. Once the background model update is complete, the tracker activates the background subtraction module in the target detection and initial pre-processing phases. Figs. 7 illustrates the process. See video: `http://www.youtube.com/watch?v=LpnUkf2GEQ4`

## 4.4 Modality Weight Generation for Multi-modal Fusion

Fig. 8 describes the module performing the task of generating the weights for the individual visual modalities through scene analysis. This module consists of two scene analysis units, each analysing the usability of the individual modality in the current

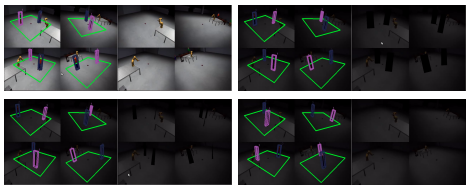

Figure 7: Background update in presence of foreground targets..

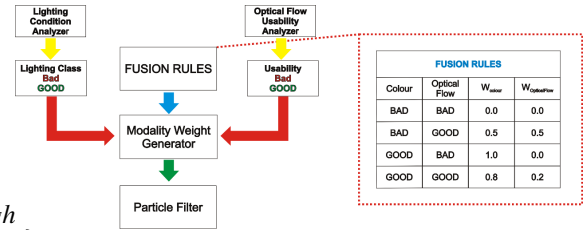


Figure 8: Intelligent fusion module to generate weights for the individual modalities through scene analysis.

scene. The usability of the modalities can be represented in the form of classes. The class categories can be divided into two simple types namely, *Bad* and *Good* or more if needed.

The optical flow usability class is considered to be Bad when,

- The target is stationary or moving with a velocity below a certain threshold $V_{st}$
- The target is moving closer than a defined threshold $d_{min}$ to another target and the absolute difference of the optical flow direction components is below a certain threshold $\theta_{min}$.

On the other hand, the optical flow usability class is considered to be Good when,

- The target is moving with a velocity higher than the defined threshold $V_{st}$ and at a distance greater than $d_{min}$ with respect to all other targets.
- It has a velocity component higher than $V_{st}$ and the absolute difference of its optical flow direction component with other targets is greater than $\theta_{min}$.

Once the usability classes of the respective visual modalities are known, this information is supplied to the modality weight generator. The rule based fusion technique is constructed through a fixed set of rules defined by the user. These rules specify the combination of normalized weights to be assigned to the two modalities for each possible combination of classes. Fig. 8 illustrates a simple fusion rule data-bank for the binary classes consisting of *Bad* and *Good* labels. As mentioned above, these classes can be extended to a wider range along with a more dense rule data-bank.

Once the individual weights for each modality is obtained, they are fused in order to obtain a global

likelihood. The fusion operation is performed for each hypothesis generated by the particle filter and for each camera view. When both modalities are unsuitable for tracking, the tracker declares a target loss and instantiates the target recovery mechanisms in the form of re-detection. The mathematical representation of the complete fusion procedure is formulated below:

$$U_i^{\text{col}our} = L_{svm} I_{id},\ U_i^{flow} = A_{flow} T_{tid} \quad (17)$$

$$\left(W_{\text{col}our}, W_{flow}\right) = R\left(U_i^{\text{col}our}, U_{i_{tid}}^{flow}\right) \quad (18)$$

where, $U^{colour_i}$ is the usability class for the colour modality in the $i^{th}$ hypothesis, $L_{svm}$ is the machine learning based lighting condition classifier and $I_{id}$ is the current image from the camera. $U_{i_{tid}}^{flow}$ is the usability class for the optical flow modality in the $i^{th}$ hypothesis for the target with id $tid$. $A_{flow}$ is the function which performs the optical flow usability check on the motion parameters of the current target given by $T_{tid}$. $(W_{colour}, W_{flow})$ are the unique weights for the two modalities using the fusion rule data-bank $R$. Finally, $L_{filter_h}$ is the global likelihood.

# 5 Experiments

In this section the experiments are discussed.

## 5.1 Ground Truth Generation

There is no unified benchmarking and quantitative analysis framework for stereo multiple human trackers. Different system test their method in different ways making quantitative comparisons difficult. Ground truth generation methods are either manual, semi-automatic or automatic (D'Orazio et al., 2009), (Dollár et al., 2009). They cannot guarantee accuracy since they themselves have a certain tolerance. In order to generate ground truth without inherent errors, our test environment was modelled in 3D in its completeness using Blender (Roosendaal and Selleri, 2004). The cameras were reproduced with exact intrinsic and extrinsic parameters. The light sources were modelled similar to the ones used in the lab environment. The humans were modelled using simple models. For each target, the motion trajectories can be planned and simulated. This implies that the human targets move with 2 *DOF* and an additional degree of freedom for rotation along the trunk axis. Once the animation is ready, it can be rendered using the perspective of the cameras. The simulated trajectory data of each target was extracted through a python script within Blender.

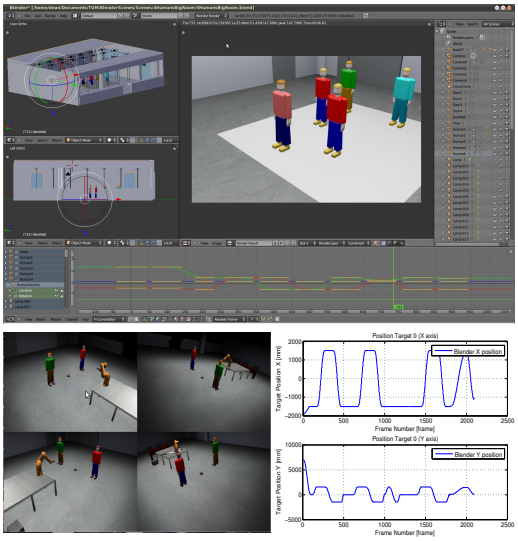


Figure 9: The lab environment modelled in 3D and extraction of simulated trajectory data.

## 5.2 Experimental Validations

The experiments were carried out in the virtual environment and real scenarios. The system was tested for various aspects. It was first tested with 5 targets in a scene, where two targets have similar appearance and move close to each other. This was followed by a similar test with 3 targets in the real environment. Another test was conducted with two targets with exact similar appearance moving very close to each other and in the same direction. Further, the system was tested both in the virtual and real environment under drastically changing lighting conditions. Finally the intelligent multi-modal fusion was tested. In all the tests the tracker never lost a target and always maintained its target Id.

Fig. 10 illustrates the first test with 5 targets. It shows the plots of the tracked trajectories along with the actual trajectories obtained from the ground truth generator. The right most column represents the error computed in the $X$ and $Y$ directions. It can be seen that the standard deviation of the error computer over the entire sequence is below 10 *cm* even under increased numbers of mutual occlusions simultaneously in multiple cameras. See video: `http://www.youtube.com/watch?v=-Y-sZ2g53fM`

For the real world scenario actual ground truth data is not available. Then, the system was tested in the actual laboratory. The experiment was performed

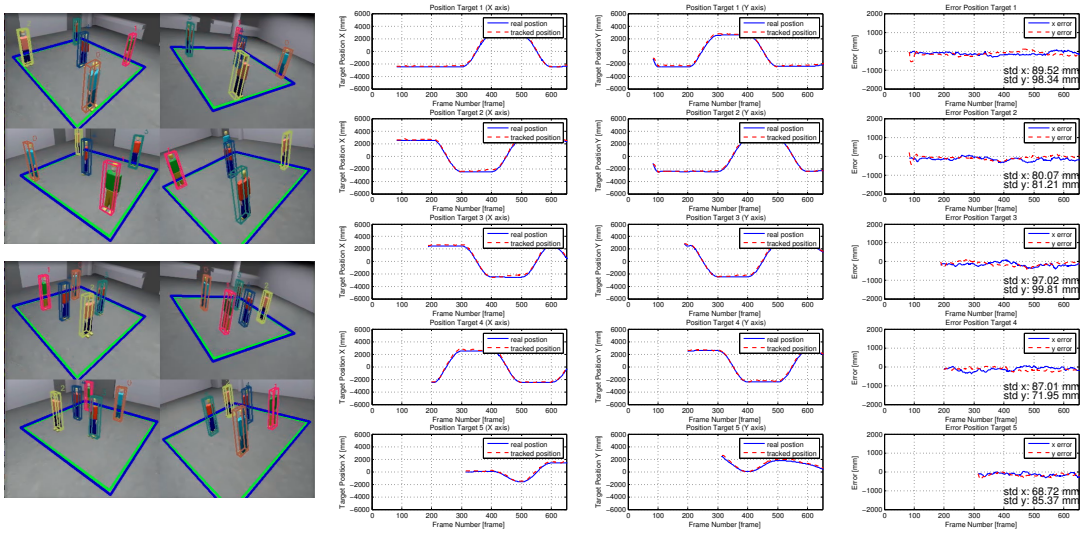


Figure 10: Illustration of the tracking system with 5 targets in the scene moving close to each other and two targets having similar appearance.

with 3 targets due to the limited area in our lab. In order to get an estimate of the trajectories, a fixed path was marked in the tracking area and the humans were asked to move along this path.

Fig. 11 illustrates the results obtained. In the first row, it can be observed that the targets are at their initial position and being tracked. Towards the right, a red box can be observed, which represents the path the targets are supposed to move on when observed from the top view. The second row shows the targets being tracked after they have moved. To its right, the generated trajectories have been plotted which approximate the desired shape. The trajectory of each target is plotted and is similar to the colour id set by the tracker. See video: `http://www.youtube.com/watch?v=wePVQ7cXB9c`

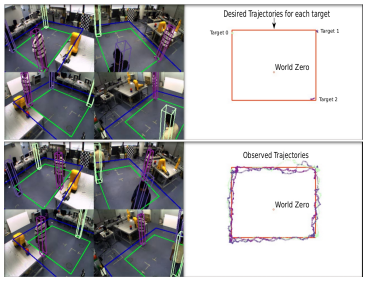


Figure 11: Experiment results in real world environment with 3 targets and the motion trajectories generated.

The next experiment consists of two targets with exactly the same appearance. The two targets enter the tracking area and move very close to each other at a distance less than 10 *cm*. To increase the tracking complexity, the motion in certain parts of the sequence are in the same direction.

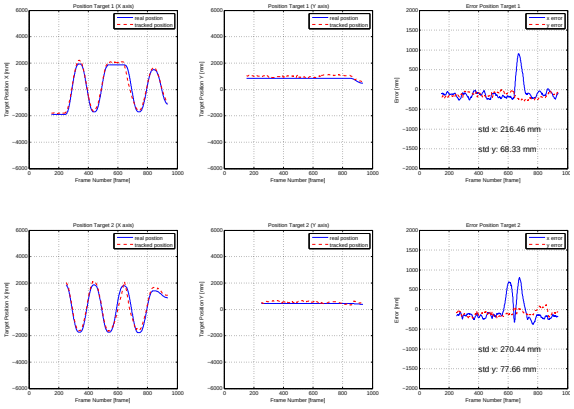


Figure 12: Experiments results between the tracked positions with respect to the ground truth of 2 targets with exactly similar appearance and very close motion in the same direction.

Fig. 12 and video `http://www.youtube.com/watch?v=u6OFTVWO-qg` presents an experiment in a scenario consisting of two targets with exactly the same appearance. It can be observed that there is a sharp overshoot in the error in the dominant direction of motion when the targets start moving in the same direction but recovers in a few frames through the occlusion handling module. The average error is appeared to be twice as much as normal due to the overshoot, but in other parts of the sequence the error is still lower than 10 *cm*.

Fig. 13 illustrates an experiment that validates the use of intelligent multi-modal fusion in order to main-

tain the robustness of the tracker. The experiment plots the feature matching distance from the colour and optical flow modalities.

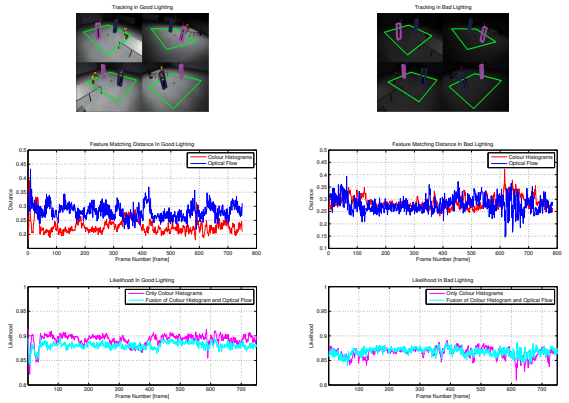


Figure 13: Experiments results for multi-modal fusion.

The quality of lighting in the second scene is very bad as compared to the first one. From the distance plots it can be observed that, the feature matching distance of colour histogram worsens in the bad lighting condition while the optical flow distance remains fairly constant. This shows the robustness of optical flow and the sensitivity of the colour distributions in changing lighting conditions and that optical flow information is a good supporting feature under such circumstances. In each of the two scenarios, two likelihood plots are generated. The first represents likelihood computed using only colour information followed by likelihood computed through multi-modal fusion of colour and optical flow information. The likelihood without fusion degrades in bad lighting conditions while the likelihood with multi-modal fusion remains fairly constant. This indicates that under bad lighting conditions the multi-modal fusion of colour and optical flow ensures robust and stable tracking results.

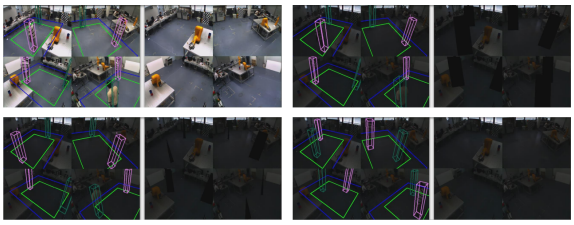

Figure 14: Experiment results in real world environment under drastically changing lighting conditions.

Fig. 14 illustrates the experiment conducted to validate the performance under drastic changes in lighting conditions in a real scenario. The system requires to detect such events and update the background model in presence of foreground targets. The top-left shows how the tracker successfully tracks the two targets. Top-right represents the background model. The bottom-left represents the frame in which the light changes drastically. Finally, the bottom-right shows the complete background model updated in each camera view and the tracker successfully tracks all the targets. See video: `http://www.youtube.com/watch?v=yZHCXgdDf14`

# 6 CONCLUSIONS AND FUTURE WORK

This paper introduced a vision based 3D multiple human tracker with primary focus on robustness enhancement. Novel techniques in the direction of robustness enhancement were introduced and validated. Multiple cameras and visual modalities were integrated in a single workstation. The system is highly robust and maintains real-time performance irrespective of number of targets. The primary contributions are: A vision based real-time 3D multiple human tracking system based on a modular building blocks approach. It is capable of detecting and tracking multiple humans in real-time within a desired area of interest. A module which detects and handles multiple occlusions between human targets while they are being tracked has been also introduced. This ensures robust tracking of targets, maintaining their Ids. A model trained to classify the current lighting conditions into one of its pre-defined classes. Depending on the classification results the quality of the lighting conditions is determined. The model is trained using a large dataset of lighting conditions representing the desired classes. A module that uses the machine learning based lighting conditions classifier in order to detect drastic changes in lighting conditions. It further performs the non trivial task of updating the background model in the presence of foreground targets being tracked. From the analysis of each visual modality the correct weights are generated in order to maintain the robustness of the tracker. Furthermore, a novel approach through which zero error ground truth data for evaluation and validation of the tracker was introduced. This is based on a 3D model of the complete workspace with great detail and simulation of human motion to extract the trajectories.

# REFERENCES

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their

probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109.

Cai, Q. and Aggarwal, J. (1996). Tracking human motion using multiple cameras. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 68–72. IEEE.

Chang, T. and Gong, S. (2001). Tracking multiple people with a multi-camera system. *womot*, page 0019.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multi-class svms. In *JMLR*.

Dockstader, S. and Tekalp, A. (2001). Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10):1441–1455.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *CVPR*.

D'Orazio, T., Leo, M., Mosca, N., Spagnolo, P., and Mazzeo, P. (2009). A semi-automatic system for ground truth generation of soccer video sequences. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 559–564. IEEE.

Eshel, R. and Moses, Y. (2008). Homography based multiple camera detection and tracking of people in a dense crowd.

Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multicamera people tracking with a probabilistic occupancy map. pattern analysis and machine intelligence. *IEEE Transactions on*, 30(2):267–282.

Focken, D. and Stiefelhagen, R. (2002). Towards vision-based 3-d people tracking in a smart room.

Francois, J. B., Berclaz, J., Fleuret, F., and Fua, P. (2006). Robust people tracking with global trajectory optimization. In *In Conference on Computer Vision and Pattern Recognition*, pages 744–750.

Haritaoglu, I., Harwood, D., and Davis, L. S. (1998). W4: A real time system for detecting and tracking people. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 962, Washington, DC, USA. IEEE Computer Society.

Hayashi, K., Hashimoto, M., Sumi, K., and Sasakawa, K. (2004). Multiple-person tracker with a fixed slanting stereo camera. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 681–686. IEEE.

Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., and Maybank, S. (2006). Principal axis-based correspondence between multiple cameras for people tracking. pattern analysis and machine intelligence. *IEEE Transactions on*, 28(4):663–671.

Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28.

Isard, M. and MacCormick, J. (2001). Bramble: A bayesian multiple-blob tracker. In *ICCV*, pages 34–41.

Khan, S. and Shah, M. (2008). Tracking multiple occluding people by localizing on multiple scene planes. *IEEE transactions on pattern analysis and machine intelligence*, pages 505–519.

Kim, K. and Davis, L. (2006). Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. *Computer Vision–ECCV 2006*, pages 98–109.

Roosendaal, T. and Selleri, S. (2004). *The Official Blender 2.3 Guide: Free 3D Creation Suite for Modeling, Animation, and Rendering*. No Starch Press.

Santos, T. T. and Morimoto, C. H. (2011). Multiple camera people detection and tracking using support integration. *Pattern Recognition Letters*, 32(1):47–55.

Siebel, N. T. and Maybank, S. J. (2002). Fusion of multiple tracking algorithms for robust people tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 373–387, London, UK. Springer-Verlag.

Soto, C., Song, B., and Roy-Chowdhury, A. (2009). Distributed multi-target tracking in a self-configuring camera network.

Toyama, K. and Horvitz, E. (2000). Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In *Proceedings of ACCV 2000, Fourth Asian Conference on Computer Vision*. Citeseer.

Weathersby, M. and Schmieder, D. (1984). An experiment quantifying the effect ofclutter on target detection. In *Proceedings of the International Society for Optical Engineering (SPIE)*, pages 26–33.

Zhao, T., Aggarwal, M., Kumar, R., and Sawhney, H. (2005). Real-time wide area multi-camera stereo tracking.