Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment

Keni Bernardin, Alexander Elbs, Rainer Stiefelhagen
Institut für Theoretische Informatik
Interactive Systems Lab
Universität Karlsruhe, 76131 Karlsruhe, Germany
keni@ira.uka.de, alex@segv.de, stiefel@ira.uka.de

Abstract

Simultaneous tracking of multiple persons in real world environments is an active research field and several approaches have been proposed, based on a variety of features and algorithms. Recently, there has been a growing interest in organizing systematic evaluations to compare the various techniques. Unfortunately, the lack of common metrics for measuring the performance of multiple object trackers still makes it hard to compare their results.
In this work, we introduce two intuitive and general metrics to allow for objective comparison of tracker characteristics, focusing on their precision in estimating object locations, their accuracy in recognizing object configurations and their ability to consistently label objects over time.
We also present a novel system for tracking multiple users in a smart room environment using several cameras, based on color histogram tracking of person regions and automatic initialization using special object detectors.
This system is used to demonstrate the expressiveness of the proposed metrics through a sample performance evaluation using real test video sequences of people interacting in the smart room.

1 Introduction and Related Work

The tracking of multiple persons in camera images is a very active research field with applications in many domains. These range from video surveillance, over automatic indexing, to intelligent interactive environments. Especially in the last case, a robust person tracking module can serve as a poweful building block to support other techniques, such as gesture recognizers, face identifiers, head pose estimators [10], scene analysis tools, etc. In the last few years, more and more approaches have been presented to tackle the problems posed by unconstrained, natural environments and bring person trackers out of the laboratory environment and into real world scenarios.

In recent times, there has also been a growing interest in performing systematic evaluations of such tracking tools with common databases and metrics. Examples are the CHIL project, funded by the EU [17], the VACE project in the U.S. [18], but also a growing number of workshops (PETS [19], EEMCV [20], etc). However, there is still no general agreement on a principled evaluation procedure using a common set of objective and intuitive metrics for measuring the performance of multiple object trackers. Because of this lack of metrics, some researchers present their tracking systems without any quantitative evaluation of their performance (e.g. [1, 7, 15]). On the other hand, a multitude of isolated measures were defined in individual contributions to validate trackers using various features and algorithms (see e.g. [2, 3, 5, 11, 13]), but no common agreement on a best set of measures exists.

To remedy this, this paper proposes a thorough procedure to detect all types of errors produced by a multiple object tracking system and introduces two novel metrics, the Multiple Object Tracking Precision ($MOTP$), and the Multiple Object Tracking Accuracy($MOTA$), that intuitively express a tracker's characteristics and could be used in general performance evaluations.

Perhaps the work that most closely relates to ours is that of Smith et al. in [4]. The authors also attempt to define an objective procedure and measures for multiple object tracker performance. However, key differences to our contribution exist: In [4], the autors introduce a large number of metrics: 5 for measuring object configuration errors, and 4 for measuring inconsistencies in object labeling over time. Some of the measures are defined in a dual way for trackers and for objects (e.g. MT/MO, FIT/FIO, TP/OP). This could make it difficult to gain a clear and direct understanding of the tracker's overall performance. Moreover, under certain conditions, some of these measures can behave in a non-intuitive fashion (such as the CD, as the authors state, or the $\overline{FP}$ and $\overline{FN}$, as we will demonstrate later). In comparison, we introduce just 2 overall performance measures that allow a clear and intuitive insight into the main tracker characteristics: its precision in estimating object positions, its ability to determine the number of objects and their configuration, and its skill at keeping consistent tracks over time. In addition, we offer an experimental validation of the presented theoretical framework by performing sample evaluation runs on 2 variants of a multiple person tracker, using real data recorded in a smart room environment. A demonstration run on simulated data is also performed to better illustrate the expressiveness of the proposed metrics.

The system used in our evaluations is a 3D multiple per-

son tracker developed for use in our smart room. It initializes automatically using special person detectors, performs color histogram tracking of body parts on several camera views and intelligently fuses the 2D information to produce a consistent set of 3D hypotheses. It is used as a proof of concept for the introduced $MOTP$ and $MOTA$ metrics, which are used to measure its accuracy on datasets of varying degrees of difficulty.

The remainder of the paper is organized as follows: Section 2 presents the new metrics, the $MOTP$ and the $MOTA$ and a detailed procedure for their computation. Section 3 briefly introduces the developed multiperson tracker which will be used in the evaluations. In Section 4, the sample performance measurements are shown and the usefulness of the metrics is discussed. Finally, Section 5 gives a summary and a conclusion.

2 Performance Metrics for Multiple Object Tracking

To help better understand the proposed evaluation metrics, we first explain what qualities we expect from an ideal multiple object tracker. It should at all points in time find the correct number of objects present and estimate the position of each object as precisely as possible (Note that properties such as the size, contour, orientation or speed of objects are not considered here). It should also keep consistent track of each object over time: Each object should be assigned a unique track ID which stays constant throughout the sequence (even after temporary occlusion, etc). This leads to the following design criteria for performance evaluation metrics:

- They should allow to judge the tracker's precision in determining exact object locations.
- They should reflect its ability to consistently track object configurations through time, i.e. to correctly trace object trajectories, producing exactly one trajectory per object.

Additionally, we expect useful metrics

- to have as few free parameters, adjustable thresholds, etc, as possible to help make evaluations straightforward and keep results comparable.
- to be clear, easily understandable and behave according to human intuition, especially in the occurence of multiple errors of different types or of uneven repartition of errors throughout the sequence.
- to be general enough to allow comparison of most types of trackers (2D, 3D trackers, object centroid trackers or object area trackers, etc).
- to be few in number and yet expressive, so they may be used e.g. in large evaluations where many systems are being compared.

Based on the above criteria, we propose a procedure for systematic and objective evaluation of a tracker's characteristics. Assuming that for every time frame t a multiple object tracker outputs a set of hypotheses $\{h_1 \dots h_m\}$ for a set of visible objects $\{o_1 \dots o_n\}$, the evaluation procedure comprises the following steps:
For each time frame t,

- Establish the best possible correspondence between hypotheses h_j and objects o_i
- For each found correspondence, compute the error in the object's position estimation.
- Accumulate all correspondence errors:
  - Count all objects for which no hypothesis was output as misses.
  - Count all tracker hypotheses for which no real object exists as false positives
  - Count all occurences where the tracking hypothesis for an object changed compared to previous frames as mismatch errors. This could happen, e.g., when two or more objects are swapped when they pass close to each other, or when an object track is reinitialized with a different ID, after it was previously lost because of occlusion.

Then, the tracking performance can be intuitively expressed in two numbers: The "tracking precision" which expresses how well exact positions of persons are estimated, and the "tracking accuracy" which shows how many mistakes the tracker made in terms of misses, false positives, mismatches, failures to recover tracks, etc. These measures will be explained in detail in the latter part of this section.

2.1 Establishing Correspondences Between Objects and Tracker Hypotheses

As explained above, the first step in evaluating the performance of a multiple object tracker is finding a continuous mapping between the sequence of object hypotheses $\{h_1 \dots h_m\}$ output by the tracker in each frame and the real objects $\{o_1 \dots o_n\}$. This is illustrated in Fig. 1. Naively, one would match the closest object-hypothesis pairs and treat all remaining objects as misses and all remaining hypotheses as false positives. A few important points need to be considered, though, which make the procedure less straightforward.

2.1.1 Valid Correspondences

First of all, the correspondence between an object o_i and a hypothesis h_j should not be made if their distance $dist_{i,j}$ exceeds a certain threshold T. There is a certain conceptual boundary beyond which we can no longer speak of an error in position estimation, but should rather argue that the tracker has missed the object and is tracking something else. This is illustrated in Fig. 2(a). For object area trackers (i.e.

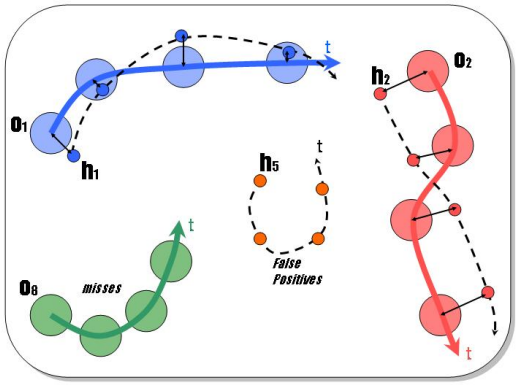


Figure 1: Mapping tracker hypotheses to objects. In the easiest case, matching the closest object-hypothesis pairs for each time frame t is sufficient

trackers that also estimate the size of objects or the area occupied by them), distance could be expressed in terms of the overlap between object and hypothesis, e.g. as in [2], and the threshold T could be set to zero overlap. For object centroid trackers, one could simply use the Euclidian distance, in 2D image coordinates or in real 3D world coordinates, between object centers and hypotheses, and the threshold could be, e.g., the average width of a person in pixels or cm. In the following, we refer to correspondences as *valid* if $dist_{i,j} < T$.

2.1.2 Consistent Tracking over Time

Second, to measure the tracker's ability to label objects consistently, one has to detect when conflicting correspondences have been made for an object over time. Fig. 2(b) illustrates the problem. Here, one track was mistakenly assigned to 3 different objects over the course of time. A mismatch can occur when objects come close to each other and the tracker wrongfully swaps their identities. It can also occur when a track was lost and reinitialized with a different identity. One way to measure such errors could be to decide on a "best" mapping (o_i, h_j) for every object o_i and hypothesis h_j, e.g. based on the initial correspondence made for o_i, or the most frequently made correspondence (o_i, h_j) in the whole sequence. One would then count all correspondences where this mapping is violated as errors. In some cases, this kind of measure can however become non-intuitive. As shown in Fig. 2(c), if, for example, the identity of object o_i is swapped just once in the middle of the tracking sequence, the time frame at which the swap occurs drastically influences the value output by the error measure.
This is why we follow a different approach: only count mismatch errors once at the time frame where a change in object-hypothesis mappings is made and consider the correspondences in intermediate segments as correct. Especially in cases where many objects are being tracked and mismatches are frequent, this gives us a more intuitive and expressive error measure.

To detect when a mismatch error occurs, a list of object-

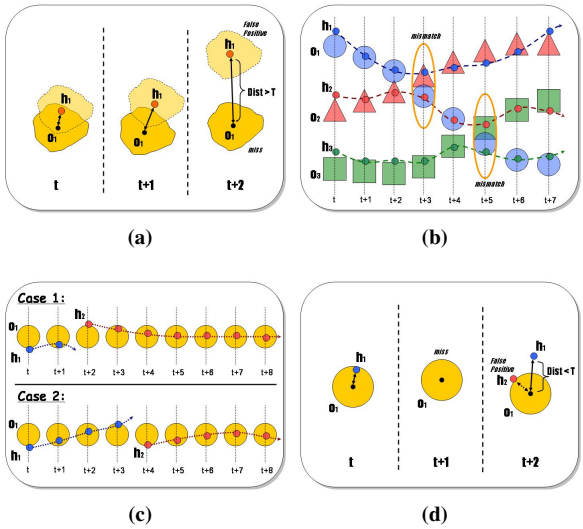


Figure 2: Optimal correspondences and error measures. Fig. 2(a): When the distance between o_1 and h_1 exceeds a certain threshold T, one can no longer make a correspondence. Instead, o_1 is considered missed and h_1 becomes a false positive. Fig. 2(b): Mismatched tracks. Here, h_2 is first mapped to o_2. After a few frames, though, o_1 and o_2 cross paths and h_2 follows the wrong object. Later, it wrongfully swaps again to o_3. Fig. 2(c): Problems when using a sequence-level "best" object-hypothesis mapping based on most frequently made correspondences. In the first case, o_1 is tracked just 2 frames by h_1, before the track is taken over by h_2. In the second case, h_1 tracks o_1 for almost half the sequence. In both cases, a "best" mapping would pair h_2 and o_1. This however leads to counting 2 mismatch errors for *case 1* and 4 errors for *case 2*, although in both cases only one error of the same kind was made. Fig. 2(d): Correct reinitialization of a track. At time t, o_1 is tracked by h_1. At $t+1$, the track is lost. At $t+2$, two valid hypotheses exist. The correspondence is made with h_1 although h_2 is closer to o_1, based on the knowledge of previous mappings up to time $t+1$

hypothesis mappings is constructed. Let $M_t = \{(o_i, h_j)\}$ be the set of mappings made up to time t and let $M_0 = \{\}$. Then, if a new correspondence is made at time $t+1$ between o_i and h_k which contradicts a mapping (o_i, h_j) in M_t, a mismatch error is counted and (o_i, h_j) is replaced by (o_i, h_k) in M_{t+1}.

The so constructed mapping list M_t can now help to establish optimal correspondences between objects and hypotheses at time $t+1$, when multiple valid choices exist. Fig. 2(d) shows such a case. When it is not clear, which hypothesis to match to an object o_i, priority is given to h_o with $(o_i, h_o) \in M_t$, as this is most likely the correct track. Other hypotheses are considered false positives, and could have occured because the tracker output several hypotheses for o_i, or because a hypothesis that previously tracked another object accidentally crossed over to o_i.

2.1.3 Mapping Procedure

Having clarified all the design choices behind our strategy for constructing object-hypothesis correspondences, we summarize the procedure:

Let $M_0 = \{\}$. For every time frame t,

1. For every mapping (o_i, h_j) in M_{t-1}, verify if it is still valid. If object o_i is still visible and tracker hypothesis h_j still exists at time t, and if their distance does not exceed the threshold T, make the correspondence between o_i and h_j for frame t.

2. For all objects for which no correspondence was made yet, try to find a matching hypothesis. Allow only one to one matches. Start by matching the pair with the minimal distance and then go on until the threshold T is exceeded or there are no more pairs to match. If a correspondence (o_i, h_k) is made that contradicts a mapping (o_i, h_j) in M_{t-1}, replace (o_i, h_j) with (o_i, h_k) in M_t. Count this as a mismatch error and let mme_t be the number of mismatch errors for frame t.

3. After the first two steps, a set of matching pairs for the current time frame is known. Let c_t be the number of matches found for time t. For each of theses matches, calculate the distance d_t^i between the object o_i and its corresponding hypothesis.

4. All remaining hypotheses are considered false positives. Similarly, all remaining objects are considered misses. Let fp_t and m_t be the number of false positives and misses respectively for frame t. Let also g_t be the number of objects present at time t.

5. Repeat the procedure from step 1 for the next time frame. Note that since for the initial frame, the set of mappings M_0 is empty, all correspondences made are initial and no mismatch errors occur.

In this way, a continuous mapping between objects and tracker hypotheses is defined and all tracking errors are accounted for.

2.2 Performance Metrics

Based on the matching strategy described above, two very intuitive metrics can be defined.

1. The *M*ultiple *O*bject *T*racking *P*recision ($MOTP$).

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}$$

It is the total position error for matched object-hypothesis pairs over all frames, averaged by the total number of matches made. It shows the ability of the tracker to estimate precise object positions, independent of its skill at recognizing object configurations, keeping consistent trajectories, etc.

2. The *M*ultiple *O*bject *T*racking *A*ccuracy ($MOTA$).

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

where m_t, fp_t and mme_t are the number of misses, of false positives and of mismatches respectively for time t. The $MOTA$ can be seen as composed of 3 error ratios:

$$\overline{m} = \frac{\sum_t m_t}{\sum_t g_t},$$

the ratio of misses in the sequence, computed over the total number of objects present in all frames,

$$\overline{fp} = \frac{\sum_t fp_t}{\sum_t g_t},$$

the ratio of false positives, and

$$\overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t},$$

the ratio of mismatches.

Summing up over the different error ratios gives us the total error rate E_{tot}, and $1 - E_{tot}$ is the resulting tracking accuracy. The $MOTA$ accounts for all object configuration errors made by the tracker, false positives, misses, mismatches, over all frames. It is similar to metrics widely used in other domains (such as the Word Error Rate (WER), commonly used in speech recognition) and gives a very intuitive measure of the tracker's performance at keeping accurate trajectories, independent of its precision in estimating object positions.

Remark on Computing Averages: Note that for both $MOTP$ and $MOTA$, it is important to first sum up all errors across frames before a final average or ratio can be computed. The reason is that computing ratios r_t for each frame t independently before calculating a global average $\frac{1}{n}\sum_t r_t$ for all n frames (such as, e.g., for the $\overline{FP}$ and $\overline{FN}$ measures in [4]), can lead to non-intuitive metric behavior. This is illustrated in Fig. 3. Although the tracker consistently missed most objects in the sequence, computing ratios independently per frame and then averaging would still yield only 50% miss rate. Summing up all misses first and computing a single global ratio, on the other hand, produces a more intuitive result of 80% miss rate.

3 A 3D Multiperson Tracker using Color and Object Detectors

In order to experimentally validate the new metrics introduced here, we performed an example evaluation of an indoor multiperson tracking system developed for our smartroom [16]. This system will now be briefly presented.

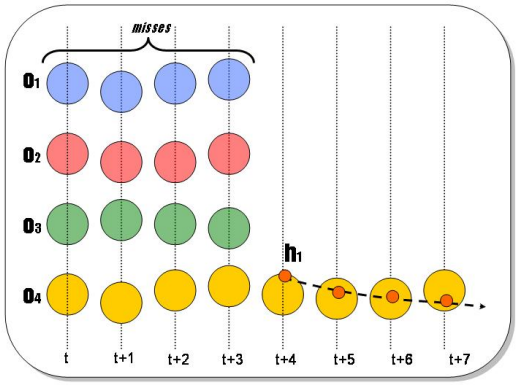


Figure 3: Computing error ratios. Assume a sequence length of 8 frames. For frames t_1 to t_4, 4 objects $o_1 \dots o_4$ are visible, but none is being tracked. For frames t_5 to t_8, only o_4 remains visible, and is being consistently tracked by h_1. In each frame $t_1 \dots t_4$, 4 objects are missed, resulting in 100% miss rate. In each frame $t_5 \dots t_8$, the miss rate is 0%. Averaging these frame level error rates yields a global result of $\frac{1}{8}(4{\cdot}100{+}4{\cdot}0) = 50\%$ miss rate. On the other hand, summing up all errors first, and computing a global ratio yields a far more intuitive result of $16misses/20objects = 80\%$

The developed system is a 3D tracker that uses several fixed cameras installed at the room corners. It is designed to function with a variable number of cameras, with precision increasing as the number of cameras grows. It performs tracking first separately on each camera image, using color histogram models. Color tracks are initialized automatically using a combination of foreground maps and special object detectors. The information from several cameras is then fused to produce 3D hypotheses of the persons' positions. A more detailed explanation of the system's different components is given in the following.

3.1 Classifier Cascades and Foreground Segmentation

A set of special object detectors is used to detect persons in the camera images. They are classifier cascades that build on haar-like features, as decribed in [8, 14]. For our implementation, the cascades were taken from the OpenCV [21] library. Two types of cascades are used: One trained to recognize whole silhouettes of standing persons (*full body*), and one to recognize the upper body region of standing or sitting persons (*upper body*). The image is scanned at different scales and bounding rectangles are obtained for regions likely to contain a person. By using these detectors, we avoid the drawbacks of creation/deletion zones and are able to initialize or recover a track at any place in the room.

Further, to reduce the amount of false detector hits, a preprocessing step is made on the image. It is first segmented into foreground regions by performing background subtraction using an adaptive background model. The foreground regions are then scanned using the classifier cascades. This combined approach offers two advantages: The cascades, on the one hand, increase robustness to segmentation errors, as foreground regions not belonging to persons, such as moved chairs, doors, shadows, etc, are ignored. The foreground segmentation, on the other hand, helps to decide which of the pixels inside a detection rectangle belong to a person, and which to the background. Knowing exactly which pixels belong to the detected person is useful to create accurate color histograms and improve color tracking performance.

3.2 Color Histogram Tracking and 2D Hypotheses

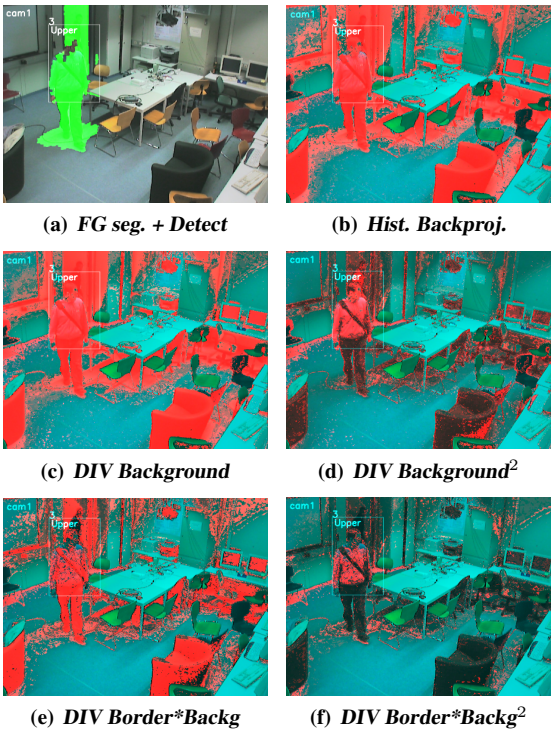
Whenever an object detector has found an upper or a full body in the image, a color histogram of the respective person region is constructed from the foreground pixels belonging to that region, and a track is initialized. The actual tracking is done based only on color features by using the meanshift algorithm [6] on histogram backprojection images. Care must be taken when creating the color histograms to reduce the negative effect of background colors that may have been mistakenly included in the person silhouette during the detection and segmentation phase. This is done by histogram division, as proposed in [9]. Several types of division are possible (division by a general background histogram, by the histogram of the background region immediately surrounding the person, etc, see Fig. 4). The choice of the best technique depends on the conditions at hand and is made automatically at each track initialization step, by making a quick prediction of the effect of each technique on the tracking behavior in the next frame.

To ensure continued tracking stability, the histogram model for a track is also adapted every time a classifier cascade produces a detection hit on that track. Tracks that are not confirmed by a detection hit for some time are deleted, as they are most likely erroneous.

The color based tracker, as described above, is used to produce a 2D hypothesis for the position of a person in the image. Based on the type of cascade that triggered initialization of the tracker, and the original size of the detected region, the body center of the person is estimated and output as hypothesis. When several types of trackers (upper body and full body) are available for the same person, a combined output is produced.

3.3 Fusion and Generation of 3D Hypotheses

The 2D hypotheses produced for every camera view are triangulated to produce 3D position estimates. For this, the cameras must be calibrated and their position relative to a general room coordinate system known. The lines of view (*LOV*) coming from the optical centers of the cameras and passing through the 2D hypothesis points in their respective image planes are intersected. When no exact intersection point exists, a residual distance between LOVs, the triangulation error, can be calculated. This error value is used by an intelligent 3D tracking algorithm to establish likely correspondences between 2D tracks (as in [12]). When



(a) ***FG seg. + Detect*** **(b)** ***Hist. Backproj.***

(c) ***DIV Background*** **(d)** ***DIV Background***^{2}

(e) ***DIV Border*Backg*** **(f)** ***DIV Border*Backg***^{2}

Figure 4: Improving color histograms. *a)* shows the results of foreground segmentation (in *green*) and object detection (*white rectangle*). The foreground pixels inside the rectangle are used to create the person's color histogram. *b)* shows the results of histogram backprojection. *Red* means high probability of belonging to the person, *blue* means low probability. *c), d), e) and f)* show the effect of different types of histogram division. *Background* stands for the general background histogram. *Border* stands for the histogram of the background region immediately surrounding the person

the triangulation error between a set of 2D hypotheses is small enough, they are associated to form a 3D track. Likewise, when it exceeds a certain threshold, the 2D hypothesis which contributes most to the error is dissociated again and the 3D track is maintained using the remaining hypotheses. The tracker requires a minimum of 2 cameras to produce 3D hypotheses, and becomes more robust as the number of cameras increases.

Once a 3D estimate for a person's position has been computed, it is further used to validate 2D tracks, to initiate color histogram tracking in camera views where the person has not yet been detected, to predict occlusions in a camera view and deactivate the involved 2D trackers, and to reinitialize tracking even in the absence of detector hits.

The developed multiperson tracker draws its strength from the intelligent fusion of several camera views. It initializes its tracks automatically, constantly adapts its color models and verifies the validity of its tracks through the use of special object detectors. It is capable of tracking several people, regardless if they are sitting, moving or standing still, in a cluttered environment with uneven lighting conditions.

4 Experimental Evaluation

To demonstrate the effectiveness of the $MOTP$ and $MOTA$ metrics, sample evaluation runs were made for the system presented in section 3, tracking real users in a realistic scenario, and for synthetic data simulating an imperfect tracker's output.

To this effect, a series of video recordings was made in the smart room involving several people. The sequences were captured by 4 fixed SONY DFW-V500 color firewire cameras placed in the room corners, delivering images with a resolution of 640x480 pixels at 15fps. The scenes show 3 to 4 people walking around in the room, conversing, standing still, sitting down in front of displays or at the table. The room is relatively cluttered, there is no clean background and uneven light is being cast by the ceiling lamps.

The recorded sequences were hand labeled, to provide the ground truth person positions the tracker hypotheses will be matched against. The centroids of the individuals' heads were marked in all camera views and triangulation of these points then provided the 3D reference positions. Manual labels were created only for every 15th frame of video, to ease the labeling task. While the tracker was run on all video frames, its output was evaluated only on labeled frames.

For this evaluation, the task was to estimate the positions of the persons on the ground, so the 3D ground truth points and the tracker hypotheses were first projected to the ground and error measures then computed in 2D. The distance measure used was the Euclidian distance in mm and the threshold T was set to 500mm.

Three different experiments were conducted using a real tracker and data:

Table 1: Results for real tracker data (*Run1* to *Run3*)

Run	$MOTP$	$\overline{m}$	$\overline{fp}$	$\overline{mme}$	$MOTA$
1:MPT_{SeqA}	168mm	5.6%	36.1%	2.4%	55.9%
2:MPT_{SeqB}	169mm	29.8%	28.9%	2.7%	38.6%
3:MPT^{Bst}_{SeqA}	168mm	5.7%	0.5%	5.4%	88.3%

Table 2: Results for synthetic tracker data (*Run4*)

σ	$MOTP$	$\overline{m}$	$\overline{fp}$	$\overline{mme}$	$MOTA$
200	232mm	5.2%	5.2%	0.0%	89.7%
400	322mm	43.5%	43.5%	0.0%	12.9%
1000	320mm	89.7%	89.7%	0.9%	-80.2%

- ***Run1:*** The previously described multiperson tracker (MPT) was evaluated on a first set of video sequences, Seq_A, comprised of three different recordings showing 3 to 4 persons in the room, with a total lenght of 5 minutes.

- ***Run2:*** The same system was used, but this time on a more challenging set of sequences, Seq_B, of same length as Seq_A, and also showing 4 interacting persons. In these sequences, persons were more frequently sitting at the table, with occlusions making tracking more difficult. Also, one of the users passed the scene very quickly and stayed in the room corner for most of the time, where he could only hardly be tracked.

- ***Run3:*** This time, the tracker was artificially modified to increase its performance and evaluated again on Seq_A. Instead of using real classifier cascades, the ground truth labels were used to generate "perfect" detection hits in the images for every 15th frame. This boosted the performance of the tracking system (MPT_{Boost}). This experiment was conducted to show how the new metrics can be used to compare tracking systems of different strengths when evaluated on the same data sets.

The results are shown in Table 1.

As can be seen, in all 3 cases, the tracker is very precise at estimating locations, once a person has been found (average error $< 17cm$). The first row shows that MPT performs fairly well on Seq_A, with almost no misses or mismatches. It does, however misjudge the amount of persons often, producing a false positive rate of $\overline{fp} = 36.1\%$, which roughly means that for every 3 real persons, the system mistakenly detected a fourth one. Tracking performance decreases somewhat for Seq_B. As persons were often sitting, they were more difficult to detect, which led to a definite increase in the miss ratio. This is rightfully reflected by a proportionate decrease of the $MOTA$. The numbers for MPT_{Boost} clearly show that it is a more performant tracker. It produces practically no misses, false positives or mismatches[1], as its simulated classifier cascades produce perfect detection hits to support its tracks. As a result, it achieves a very high tracking accuracy of 88%. These demonstration runs show how easily different tracker characteristics can be read and compared from the so presented tables.

To better show the behavior of the $MOTP$, an additional experiment (*Run4*) was conducted, this time using synthetic tracker data. The ground truth positions from Seq_A were taken as a starting point, and different levels of gaussian noise with fixed mean $\mu = 0$ and variable standard deviation σ were added independently to their x and y components. The resulting positions were used as tracker hypotheses and scored. Results are shown in Table 2 for $\sigma = 200$, 400 and 1000.

As the tracker hypotheses were generated from manual labels, the number of persons is always correctly guessed and all errors come from distance errors that exceed the allowed threshold T. For $\sigma = 200$, the (fake) tracker still performs well, making few errors. The average precision lies around 23cm, which is roughly the expectation of the position error d^i_t. As σ increases, the tracking accuracy deteriorates, and for $\sigma = 1000$, the worst result of -80% is obtained. As the threshold of 500mm forces us to consider all tracks with greater errors as misses (with resulting false positives), the $MOTP$ does on average not exceed 32cm, even as noise increases. Again, tracking performance is intuitively reflected in all the numbers in our table.

5 Summary and Conclusion

In order to systematically assess and compare the performance of different systems for multiple object tracking, metrics which reflect the quality and main characteristics of such systems are needed. Unfortunately, no agreement on a set of commonly applicable metrics has yet been reached.

In this paper, we have proposed two novel metrics for the evaluation of multiple object tracking systems. The proposed metrics – the Multiple Object Tracking Precision ($MOTP$) and the Multiple Object Tracking Accuracy ($MOTA$) – are general and intuitive, and allow for objective comparision of the main characteristics of tracking systems, such as their precision in localizing objects, their accuracy in recognizing object configurations and their ability to consistently track objects over time.

We have validated the correctness and expressiveness of the proposed metrics experimentally, using a system for tracking of multiple persons in a smart room and some simulations. The results show that the proposed metrics indeed reflect the tracking behaviour of the various used systems (real and simulated) in an intuitive and meaningful way.

The paper also briefly describes the 3D multiperson tracking system used in the experiments. This tracking

[1] Note that the numbers are not zero. This is because, although the simulated detector hits are perfect, errors can still be made because of wrong correspondences, erroneous color tracks, etc.

system combines color-histogram tracking with upper- and full-body detectors, and intelligently combines the 2D-trajectories from several views to form 3D person trajectories.

6 Acknowledgement

The work presented here was partly funded by the *European Union* (EU) under the integrated project CHIL, *Computers in the Human Interaction Loop* (Grant number IST-506909).

References

[1] Wei Niu, Long Jiao, Dan Han, and Yuan-Fang Wang, *"Real-Time Multi-Person Tracking in Video Surveillance"*, Proceedings of the Pacific Rim Multimedia Conference, Singapore, 2003.

[2] Rania Y. Khalaf and Stephen S. Intille, *"Improving Multiple People Tracking using Temporal Consistency"*, Massachusetts Institute of Technology, Cambridge, MA, MIT Dept. of Architecture House_n Project Technical Report, 2001.

[3] A. Mittal and L. S. Davis, *"M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo"*, Proc. of European Conf. on Computer Vision, LNCS 2350, pp. 18-33, 2002.

[4] Kevin Smith, Sileye Ba, Jean-Marc Odobez, Daniel Gatica-Perez, *"Evaluating Multi-Object Tracking"*, Workshop on Empirical Evaluation Methods in Computer Vision (EEMCV) 2005, San Diego, CA, June 20.

[5] Neal Checka, Kevin Wilson, Vibhav Rangarajan, Trevor Darrell, *"A Probabilistic Framework for Multi-modal Multi-Person Tracking"*, Proceedings of Workshop on Multi-Object Tracking, 2003.

[6] Dorin Comaniciu and Peter Meer, *"Mean Shift: A Robust Approach Toward Feature Space Analysis"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, May 2002.

[7] Ismail Haritaoglu, David Harwood and Larry S. Davis, *"W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People"*. Third Face and Gesture Recognition Conference, pp. 222–227, 1998.

[8] Rainer Lienhart and Jochen Maydt, *"An Extended Set of Haar-like Features for Rapid Object Detection"*. IEEE ICIP 2002, Vol. 1, pp. 900–903, Sep. 2002.

[9] Kai Nickel and Rainer Stiefelhagen, *"Pointing Gesture Recognition based on 3Dtracking of Face, Hands and Head Orientation"*, 5th International Conference on Multimodal Interfaces, Vancouver, Canada, Nov. 2003.

[10] Michael Voit, Kai Nickel, Rainer Stiefelhagen, *"Multi-view Head Pose Estimation using Neural Networks"*, 2nd Workshop on Face Processing in Video (FPiV'05), in ass. with IEEE CRV 2005, Victoria, Canada, May 2005.

[11] Kai Nickel, Tobias Gehrig, Rainer Stiefelhagen, John McDonough, *"A Joint Particle Filter for Audio-visual Speaker Tracking"*, International Conference on Multimodal Interfaces ICMI 05, Trento, Italy, October 2005.

[12] Dirk Focken, Rainer Stiefelhagen, *"Towards Vision-Based 3-D People Tracking in a Smart Room"*, IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, October 14-16, 2002, pp. 400-405.

[13] Hai Tao, Harpreet Sawhney and Rakesh Kumar, *"A Sampling Algorithm for Tracking Multiple Objects"*. Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, pp. 53–68, 1999.

[14] Paul Viola and Michael Jones, *"Rapid Object Detection using a Boosted Cascade of Simple Features"*. Conference On Computer Vision And Pattern Recognition, 2001.

[15] Christopher Wren, Ali Azarbayejani, Trevor Darrell, Alex Pentland, *"Pfinder: Real-Time Tracking of the Human Body"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 19, no 7, pp. 780–785, July 1997.

[16] Alexander Elbs, *"Mehrpersonentracking mittels Farbe und Detektorkaskaden"*. Diplomarbeit. Institut für Theoretische Informatik, Universität Karlsruhe, August 2005.

[17] CHIL - Computers In the Human Interaction Loop, http://chil.server.de

[18] VACE - Video Analysis and Content Extraction, http://www.ic-arda.org

[19] PETS - Performance Evaluation of Tracking and Surveillance, http://www.cbsr.ia.ac.cn/conferences/VS-PETS-2005/

[20] EEMCV - Empirical Evaluation Methods in Computer Vision, http://www.cs.colostate.edu/eemcv2005/

[21] OpenCV - Open Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/