Dynamical Information Fusion of Heterogeneous Sensors for 3D Tracking Using Particle Swarm Optimization

Ulrich Kirchmaier, Simon Hawe, Klaus Diepold
Lehrstuhl für Datenverarbeitung
Technische Universität München

This paper presents a new method for three dimensional object tracking by fusing information from stereo vision and stereo audio. From the audio data, directional information about an object is extracted by the Generalized Cross Correlation (GCC) and the object's position in the video data is detected using the Continuously Adaptive Mean shift (CAMshift) method. The obtained localization estimates combined with confidence measurements are then fused to track an object utilizing Particle Swarm Optimization (PSO). In our approach the particles move in the 3D space and iteratively evaluate their current position with regard to the localization estimates of the audio and video module and their confidences, which facilitates the direct determination of the object's three dimensional position. This technique has low computational complexity and its tracking performance is independent of any kind of model, statistics, or assumptions, contrary to classical methods. The introduction of confidence measurements further increases the robustness and reliability of the entire tracking system and allows an adaptive and dynamical information fusion of heterogenous sensor information.

1 Introduction

Today, object tracking is an increasing research topic, due to growing security requirements. Applications such as video-conferencing, surveillance, smart automobiles, and automatic scene analysis are few examples in the field of autonomous systems heavily relying on tracking using heterogeneous sensors. A variety of single-sensor techniques based solely on sound or vision already exists for that purpose. As single or homogeneous sensor techniques techniques have their specific weaknesses when deployed as stand-alone systems, it is advantageous to combine the information obtained by two or more heterogeneous sensors. In tasks like pedestrian tracking, a thermal camera is often fused with an optical sensor to enlarge the utilizable spectrum as well as the system's robustness [1]. Combinations of video and audio signals can amongst others enhance a speech event detection by incorporating the video data at hand. Audiovisual information fusion has been successfully applied to biometric person authentication [2], as well as to man-machine communication, like the smart kiosk, a terminal which enables automatic interaction between multiple speakers and a vending machine [3].

Different approaches for audiovisual fusion based object tracking have been established. Mostly recognized are techniques based on Kalman filters. In [4], a decentralized Kalman fusion technique is applied, which recursively combines audio and video state estimates into a more reliable global position estimate. While being one of the fastest methods due to its low complexity, the Kalman fusion approach is limited by its assumption of linear dynamics and an unimodal Gaussian posterior density. However, in real-world scenarios, especially in cluttered or noisy scenes, measurements tend to have non-Gaussian, multi-modal distributions. Furthermore,

its linear state transition system - hypothesizing a deterministic movement and speed of the object to be tracked - tends to fail with sudden and abrupt movements.

A more sophisticated extension of the Kalman filter is the particle filtering, utilized for audiovisual fusion in [5, 6, 7]. Contrary to Kalman's hypothesis of a Gaussian distribution, Particle filters model a stochastic process with an arbitrary probability density function by approximating it numerically with a cloud of particles. The obtained posterior density function estimate becomes an adequate approximation to the true posterior probability density function, when the number of particles is large. Consequently, a growing number of particles improves the tracking results, but also increases the computational costs and therefore degrades the system's speed.

Similar methods for audiovisual object tracking are probabilistic graphical models, which try to exploit the mutual relationship between two modalities, like audio and video data representing the same object, in order to achieve an optimal performance. The system presented in [8] uses probabilistic generative models to describe the observed data from multimedia sequences. The models are denoted as generative as they delineate the observed data in terms of the process that generated them, using additional variables that are not observable. The models are probabilistic due to the fact that they estimate probability distributions of signals rather than describing the signals themselves.

One of the most popular among graphical-model techniques using probabilistic models is the Bayesian network approach [6, 9], which is considered to be a powerful tool for information fusion. Bayesian networks are a way of modeling a joint probability distribution of multiple random variables. In [10], a Bayesian network was used to detect the time and position of speech events by analyzing audio and video data. The gained information was then utilized to robustly recognize and separate speech signals in noisy and reverberant environments. The system was successfully operating a conference room. A similar implementation can be found in [9]. Although the Bayesian approach is very simple and powerful in principle, its central drawback in practice is that it often requires intractably large amount of computations, mainly for the execution of integrations in a very high dimensional space of random variables.

A further related approach is the usage of Hidden Markov models (HMM) [11].

Besides computational complexity, the above mentioned fusion methods have the drawback of relying on model assumptions. With the methods becoming more complex, the limitations of a distribution model may be partly overcome and more generally applicable. However, this happens on the expense of exponentially increasing complexity and processing time, while being still dependent on different assumptions, models, training sets, statistics, and transition probabilities that have to be postulated.

Many object tracking implementations merely aim on detecting and tracking objects on the screen, without determining their positions in the real-world reference frame. The calculation of the three dimensional coordinates represents an additional task for conventional tracking i.e. disparity calculation or triangulation.

It is our aim in this study to establish a minimal-complexity 3D object tracking algorithm which keeps the hardware system implementation as simple as possible in order to achieve a real-time behavior while preserving a robust tracking performance.

The sound localization method adopted in this work deploys two microphones. Current sub-space methods, e.g. the MUltiple SIgnal Classification (MUSIC) algorithm [12], demand microphone arrays equipped with eight, 16 or more elements.

The visual detection algorithm used here deploys two cameras. In contrast to other approaches [13] in which laser sensors were additionally needed we only use cameras and microphones in order to achieve three dimensional tracking. The introduced tracking algorithm is solely based on color distributions to identify and track moving objects in a video sequence. It is a robust technique more flexible than the background subtraction method [10] and well-suited for abrupt changes in the camera position as well as for alterations in the environment [14].

The tracking information delivered by the audio module is fused with that of the video system in a novel manner using Particle Swarm Optimization algorithm (PSO). In our design, a particle is regarded as a possible object position in the solution space, which is simply the three dimensional space in which the object sojourns.

The implemented tracker is composed of an audio module, the stereo camera component, and the fusion module, shown in figure 1. The camera component acquires two video frames of the same scene at a time instant t. After a matching operation, it delivers a pair of correspondence points MP_l and MP_r, which describe the coordinates of the same part of the object projected onto the left and right frames, respectively. Furthermore,

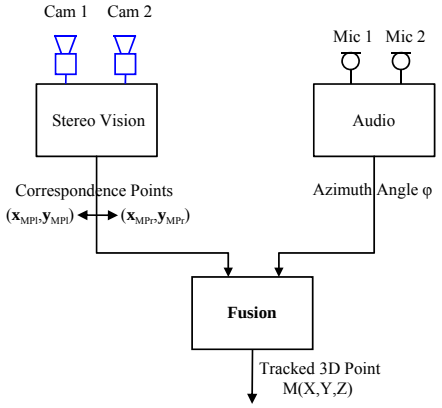

Figure 1: Components of the complete tracking system.

the visual system provides a confidence value $Conf_{vis}$, which evaluates the reliability of the visual system's result. Parallel to image processing, the audio block estimates the azimuth φ of the moving object at the same time instant t as well as a confidence measure $Conf_{aud}$, assessing the plausibility of the audio result. The information of these two blocks is then propagated to the PSO tracking module, which delivers a 3D position estimate $M(X,Y,Z)$ of the moving object at time instant t.

Parts of this work have been published in [15]. In this paper we further extend the basic approach by considering confidence measurements when evaluating the position estimates of audio and video modules. This allows an adaptive and dynamical information fusion of heterogenous sensor information, which further increases the robustness and the reliability of the entire tracking system. A detailed description of the tracking system will be presented in the following sections. To this end, this paper is organized as follows. First, the PSO concept is introduced in section 2. In section 3 and 4 we describe the audio and the video tracking systems independently. In section 5 we present a novel method for audiovisual fusion based on PSO. Experimental results and comparison with current tracking techniques are discussed in section 6. Section 7 concludes the current study and introduces venues for future work.

2 Particle Swarm Optimization

In recent years, researchers proved that the Particle Swarm Optimization is a highly efficient optimization method and a search algorithm with high performance capabilities and outstanding flexibility, making it suitable for a huge variety of signal processing tasks. The PSO algorithm, originally proposed in [16], is a simulation of a simplified social model. It draws its roots from artificial life, and was inspired by bird flocking, fish schooling, and swarming theory in particular.

The social metaphor that leads to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space", or search space, shared by neighboring individuals. Individuals may modify this "opinion state" based on three factors: (1) the knowledge of the environment (inertia part); (2) the individual's previous history of states (individual part); and (3) the previous history of states of the individual's neighborhood (social part).

An individual's neighborhood may be defined in several ways, configuring somehow the "social network" of this individual. Followingf rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over the time, a culture arises in which the individuals hold opinions that are closely related.

In the PSO algorithm each individual is called a "particle", and is subject to a movement in a multidimensional space that represents the belief space. Particles have memory, thus retaining part of their previous states. There is no restriction for particles to share the same point in belief space, but in any case their individuality is preserved. Each particle's movement is the composition of an initial random velocity and two randomly weighted

influences: individuality, the tendency to return to the particle's best previous position called $pbest$, and sociality, the tendency to move towards the neighborhood's best previous position $gbest$.

In order to obtain a measurement of the quality of a particle's current position $x(t)$ in the search space, a cost or fitness function $F(x)$ is defined to be minimized over $x(t)$.

The velocity $v_i(t)$ and position $x_i(t)$ of the particle i at any iteration is updated based on the following equations:

$$\begin{aligned} v_i(t+1) &= w \cdot v_i(t) + c_1 \cdot R_1 \cdot (pbest_i - x_i(t)) + \\ &+ c_2 \cdot R_2 \cdot (gbest - x_i(t)), \quad (1) \\ x_i(t+1) &= x_i(t) + v_i(t+1), \quad (2) \end{aligned}$$

where R_1 and $R_2 \sim [0,1]$ are uniformly distributed random variables„ w is a decay constant controlling the swarms convergence behavior, and c_1 and c_2 represent the "cognitive" respectively the "social" component (the memory and the communication) of the swarm. The velocity $v_i(t+1)$ describes the positional displacement between two time steps.

According to equation (1), each particle adjusts its velocity by combing three forces: keeping the velocity of the last moment, moving to the best position from its own memory, moving to the best position found by its neighbors. Different parameters in (1) provide varied balance among those three factors. A particle moves in the search space according to the combined velocity calculated by (1) to achieve a new position, which can present a new value of object features. This procedure can efficiently prevent the local optimum effect.

In [17], a method for recognition and localization of pedestrians in 3D space using multiple view geometry was described. This method combined feature-based 2D object classification with efficient search mechanisms based on PSO. Each particle represented a self-contained classifier, "flying" through the solution space, seeking the most "object-like" region in space. Thus, it effectively expanded the scope of an image based 2D classifier to 3D space.

An optimization-based framework for computer vision was developed in [18]. It combined ideas from PSO and statistical pattern recognition to rapidly and accurately detect and classify objects in visual imagery. Swarm intelligence was used to locate objects, like people, ground vehicles and boats by optimizing the classification confidence level. Using a cognitive swarm of cooperating classifier particles, the PSO search time was reduced by a factor of 208 compared with brute force exhaustive search.

The PSO algorithm was applied to visual object tracking in [19]. Haar-like features, extracted from a scene were transformed into a high dimensional solution space. A framework was developed combining a PSO-based searching algorithm and a Bayes-based probability algorithm. The PSO-based searching algorithm identified changes in the scene, and the probability-based algorithm estimated the best candidate of the object with the highest possibility.

In our paper we introduce a new method for object tracking in three dimensional space. The audio and vision information is fused dynamically using PSO. This novel tracking technique is faster than existing approaches while being independent of any kind of model, statistics or assumptions.

3 Sound Source Localization

Our sound localization system setup consists of two microphones placed horizontally 47 cm away from each other, and uses the Time Delay Of Arrival (TDOA) method for localization. This system delivers an azimuth angle φ which represents the relative angle between the origin of the system and the object being tracked. The audio system furthermore delivers the value $Conf_{aud}$, which describes the confidence of the obtained audio localization, i.e. the confidence of the audio system.

3.1 TDOA-Based Method

In the TDOA-based method of sound source localization, two (or more) sensors are used to estimate the traveling time of a plane wave propagating across an array of sensors, resulting in a time delay.

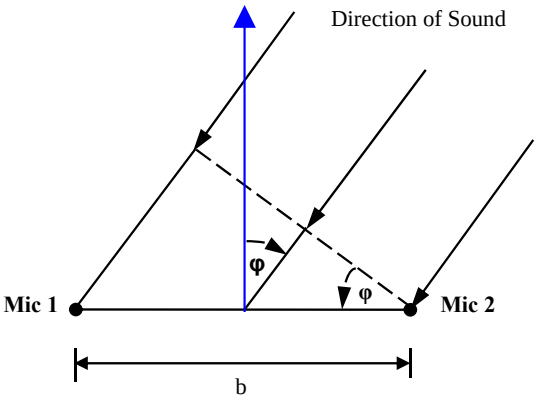

Figure 2: Geometry of the audio system, plane wave scenario. φ denotes the angle of the sound source with respect to the audio system, which consists of two microphones $Mic1$ and $Mic2$, separated by the baseline b.

Assuming a single sound source with free-field planar wave propagation in low uncorrelated noise and low reverberant conditions, the delay τ can be estimated by calculating the cross correlation of the two microphones' signals [20]. The estimated delay $\hat{\tau}_{TDOA}$ resulting from the signal source is obtained by a maximum search on the correlation result, which under consideration of the system's geography leads to the desired angle φ using

$$\varphi = \arcsin\left(\frac{\hat{\tau}_{TDOA} \cdot c}{b}\right), \tag{3}$$

where c denotes the velocity of sound and b represents the distance between the two microphones, as shown in figure 2.

Using digital signals, the accuracy of TDOA based sound localization is limited by the sampling frequency f_s, as the reciprocal value of f_s represents the smallest possible delay difference. The microphone distance b determines the maximum quantity of different delay steps. This leads to the complete room facing the microphones being divided into

$$2 \cdot \left\lfloor \frac{b f_s}{c} \right\rfloor + 1 \tag{4}$$

sections, which denotes the spatial resolution.

3.2 GCC PHAT

Since the cross correlation in time domain yields poor delay estimation with broadband signals already under low noise and reverberation conditions, the Generalized Cross Correlation (GCC) was introduced as a better solution [21]. The GCC-function $R_{x_1x_2}(\tau)$ is given by the inverse Fourier transform of the signals' cross power spectral density, that was weighted with a filter function $A(\omega)$

$$R_{x_1x_2}(\tau) = \frac{1}{2}\pi \int_{-\infty}^{\infty} A(\omega) X_1(\omega) X_2^*(\omega) e^{-i\omega t} dt, \tag{5}$$

assuming an infinite observation time, where τ again denotes the time delay between the frequency domain signals $X_1(\omega)$ and $X_2(\omega)$.

A drawback of the GCC method which uses a filter $A(\omega) = 1$, is that a maximum in the GCC function $R_{x_1x_2}(\tau)$ becomes hard to detect with narrow band signals, as the relative height of the maximum peak compared to neighboring peaks is proportional to the signals bandwidth. As a solution for this problem, a filter function which also tends to make the GCC more robust against reverberation is the PHase-Transform (PHAT) [21]. The PHAT

weighting function is defined as

$$A(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|}, \tag{6}$$

which whitens the GCC-function by weighting all components of the spectrum equally. This pre-whitening improves the detection of narrow band signals. Assuming the signal $s(t)$ uncorrelated to $n_1(t)$ and $n_2(t)$, it can be shown that the resulting PHAT-weighted GCC yields

$$R_{x_1x_2}(\tau) = \delta(\tau + \tau_{TDOA}), \tag{7}$$

resulting in a single peak at the TDOA [20].

3.3 Robustness and Confidence

In order to further increase the robustness of the GCC-PHAT sound source localization, a frame of samples is taken out of the current audio stream of each microphone and divided into N overlapping and equally sized windows, with the window size being the FFT length. For each microphone, every window i, with $i = 1...N$, is multiplied with the Hamming window and then fed to the GCC algorithm, resulting in an estimation of $\hat{\varphi}_i$. A maximum likelihood estimation of $\hat{\varphi}$ can be obtained by using the following equation

$$\hat{\varphi} = \frac{1}{N}\sum_{i=1}^{N}\hat{\varphi}_i. \tag{8}$$

This equation postulates that the sound source moves considerably slow, i.e. it does not move during one frame. Equation (8) results from the minimization of the error between the estimated angles $\hat{\varphi}_i$ and the true localization angle, assuming that the variance of φ does not change for all windows because the signal and the sensors remain unaltered during one time frame. For more information about maximum likelihood, the reader is advised to refer to [22] and [23]. To achieve a proper confidence value from the audio system, the variance is calculated additionally to the maximum likelihood angle $\hat{\varphi}$. The variance of a frame serves as a confidence value. As a high variance value means a weak sound detection, the confidence value must be low. With the values being normalized, the audio confidence can be defined as

$$Conf_{aud} = 1 - \frac{1}{N}(\hat{\varphi} - \hat{\varphi}_i)^2. \tag{9}$$

4 Visual Object Localization

Our vision system consists of two cameras, which deliver two correspondence or matching points, $MP_l = (x_{MPl}, y_{MPl})$ from the left frame and $MP_r = (x_{MPr}, y_{MPr})$ from the right frame. These points represent the 2D-projection of the object to be tracked. Additionally, the vision system delivers the value $Conf_{vis}$, which describes the confidence of the obtained matching point pair, i.e. the confidence of the vision system.

4.1 CAMshift Algorithm

Introduced in [14], the Continuously Adaptive Mean shift (CAMshift) is based on the mean shift algorithm [24]. The mean shift is a non-parametric approach to detect the mode of a probability distribution using a recursive procedure that converges to the closest stationarity point. The main steps of the mean shift procedure are:

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location in the search window.
4. Center the search window at the mean location computed in Step 3.

5. Repeat Steps 3 and 4 until convergence (or until the mean location moves less than a preset threshold).

In order to apply the mean shift method, which was implemented for probability distributions, to color tracking, a probability distribution image of the desired color, like the skin color for face tracking, is created. Using the HSV color space, 1D histograms in the Hue layer or 2D Histograms using the Hue and Saturation layer, which decrease the false detection [25], are generated to represent the distribution.

In the next step, the histogram back-projection [26] of an input image is executed based on the calculated histogram, which results in a color probability image P_{color}.

In a discrete value image, the mean location of the search window (x_c, y_c) is computed by

$$x_c = \frac{M_{10}}{M_{00}}, \tag{10}$$

$$y_c = \frac{M_{01}}{M_{00}}, \tag{11}$$

where M_{00}, M_{10} and M_{01} are the zeroth and first moments, defined as

$$\begin{aligned} M_{00} &= \textstyle\sum_x \sum_y I(x,y), \\ M_{10} &= \textstyle\sum_x \sum_y xI(x,y), \\ M_{01} &= \textstyle\sum_x \sum_y yI(x,y), \end{aligned} \tag{12}$$

with $I(x,y)$ being the intensity value of a pixel at position (x,y) in the probability image P_{color}, and x, y taking coordinate inside the search window.

The CAMshift algorithm expands the mean shift method by adjusting the size of the search window. By this extension, the method can be applied to image sequences which contain a changing shape of the tracked color distribution. The main steps of the CAMshift algorithm include

1. Choose the initial location of the search window.
2. Mean Shift as above (one or many iterations); store the zeroth moment.
3. Set the search window size equal to a function of the zeroth moment found in Step 2.
4. Repeat Steps 2 and 3 until convergence (mean location moves less than a preset threshold).

For detailed information on the CAMshift algorithm, the reader is advised to refer to [14].

4.2 Block Matching and Confidence

The CAMshift algorithm is applied to both left and right frames of the stereo vision system, yielding the two center points (x_{cl}, y_{cl}) and (x_{cr}, y_{cr}), and the track-boxes of the CAMshift. The track-box size matches the size of the CAMshift window when the algorithm has converged. In the next step a search window $W_{w\times h}$ is centered at (x_{cr}, y_{cr}) in the right frame. Inside this window a two dimensional block matching search of the left track box is initiated. This search uses the two dimensional normalized cross correlation $R(x,y)$.

The position with the maximum value of $R(x,y)$ in the right frame represents the corresponding point $MP_r = (x_{MPr}, y_{MPr})$ to the CAMshift center point (x_{cl}, y_{cl}) in the left frame, which represents the left correspondence point, i.e. $MP_l = (x_{MPl}, y_{MPl}) = (x_{cl}, y_{cl})$. The detected correspondence points are forwarded to the PSO fusion block introduced in the following section. In addition, the value of the detected maximum of $R(x,y)$ is sent to the fusion module, serving as confidence value,

$$Conf_{vis} = \max(R(x,y)), \tag{13}$$

with $Conf_{vis} = 1$ representing a perfect match.

5 Audiovisual Information Fusion and Tracking

The task of the fusion module is to combine the information conveyed by the audio and vision algorithm in order to deliver an estimate of the current 3D position corresponding to the tracked object, relative to the system origin. In this section we will briefly explain a Kalman-based fusion technique which will serve as a reference system to compare the accuracy, performance, and execution time of our fusion algorithm. Afterwards we present our PSO fusion technique.

5.1 Kalman-Based Fusion

As a reference tracking system, we implemented a three dimensional tracker using triangulation [27] and a basic linear Kalman filter [28]. The triangulation uses the two matching points MP_l and MP_r delivered by the visual localization system explained in section 4 to calculate a 3D position $(X_{vis}, Y_{vis}, Z_{vis})$ of a moving object $Alpha$, by intersecting two rays R_l and R_r defined as follows

$$R_l = \quad O_l \cdot MP_l, \tag{14}$$

$$R_r = \quad O_r \cdot MP_r, \tag{15}$$

with O_l and O_r being the origin of the left and right frames, respectively. The triangulation is calculated with P_l and P_r using the least square method.

Due to a finite resolution as well as calibration and feature localization errors, the two rays will rarely intersect in praxis. Therefore a line segment perpendicular to R_l and R_r that intersects both rays is constructed and the mid-point $Alpha(X_{vis}, Y_{vis}, Z_{vis})$, being the closest point to both R_l and R_r, is regarded as the 3D position of the object to be tracked. This is illustrated in figure 3.

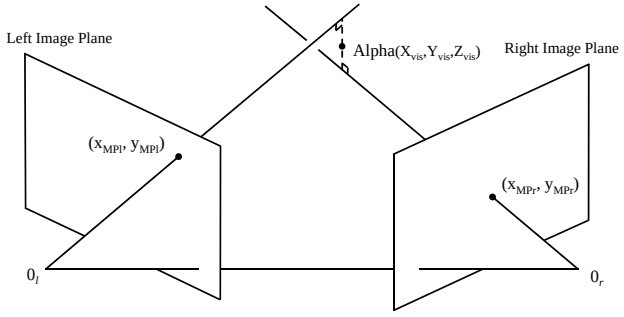

Figure 3: Geometry of the triangulation

Since our system deploys only one microphone pair and thereby detects only the azimuthal location, we use an approach different from the decentralized system presented in [4]. In our approach, a single Kalman filter, is applied to the tracking system. It uses a linear state transition matrix and adds the audio information by calculating an X_{audio} additionally to the X_{vis}, using the relation $X_{audio} = Z_{vis} \times \tan(\varphi)$ analog to the method explained in section 5.2.1, see figure 5. The X_{audio} is then added to the system by expanding the measurement vector z by one entry.

5.2 PSO-Based Fusion

The basic concept of our PSO-based audiovisual object tracker is illustrated in figure 4. Every particle M represents a position in the three dimensional space, i.e. $M \in \mathbb{R}^3$, in a coordinate system relative to the audiovisual system origin. The valid solution space is thereby the area restricted by the field of view of the stereo camera pair where the object $Alpha$ moves.

With the position information φ, obtained from the audio system, and from the vision system MP_l and MP_r, the particles moving in the three dimensional space can test a fitness function at their current position by calculating the radian and Euclidean distances to the positions favored by the audio and video system, respectively.

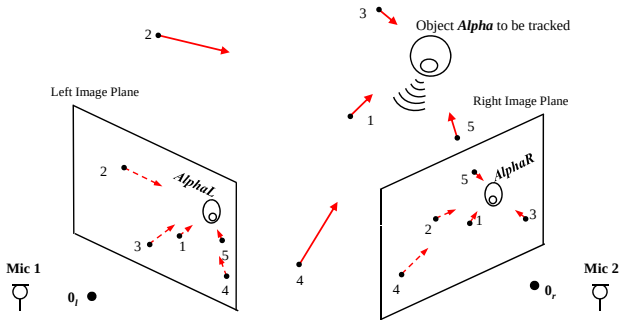

Figure 4: Model of the PSO tracker with five particles "flying" in 3D space towards the sound emitting object to be tracked, called $Alpha$. $AlphaL$ and $AlphaR$ denote the projection of $Alpha$ on the left and right frame.

5.2.1 Audio

In order to evaluate the current position of a particle with reference to the azimuth angle obtained by the audio system as illustrated in section 3, an audio distance D_{audio} is introduced. This variable represents the angular distance in radians between the audio azimuth angle φ and the angle α which lies between the particle's current position and the audio system origin. This is illustrated in figure 5. The distance D_{audio} is normalized by π, which is the greatest possible angular difference between α and φ.

$$D_{audio} = \frac{|\varphi - \alpha|}{\pi}, \tag{16}$$

with

$$\alpha = \arctan\left(\frac{X_M}{Z_M}\right), \tag{17}$$

where Z_M and X_M are the Z and X coordinates of the particle's position, respectively. Since the audio angle φ

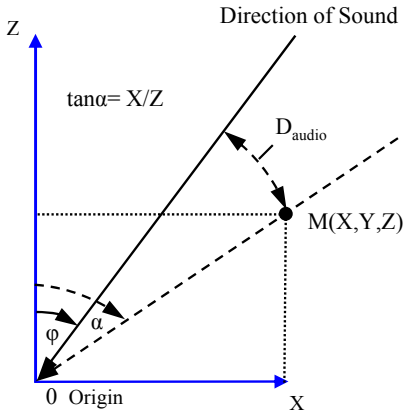

Figure 5: Relation between X and Z coordinate of the particle M, its resulting azimuth angle α relative to the audio system and the audio angle φ detected by the sound source localization.

represents an azimuth angle, $\frac{X_M}{Z_M}$ is equivalent to the tangent of the azimuth angle α.

5.2.2 Vision

In order to evaluate the particle's current position with respect to the stereo vision system, the particle is projected on the left and right frame. This process leads to the left and right image points $m_l = (x_{m_l}, y_{m_l})$ and

$m_r = (x_{m_l}, y_{m_r})$, respectively. Using a calibrated stereo camera system, the projection is obtained by

$$m_l = P_l \cdot M = K_l \cdot [I|0] \cdot M, \tag{18}$$

$$m_r = P_r \cdot M = K_r \cdot [R|t] \cdot M, \tag{19}$$

where P_l and P_r are the projection matrices for the left and right frame, under the assumption that the origin of the left image plane O_l is regarded as the system's origin. The matrices $[I|0]$ and $[R|t]$ describe the homography between the left and right frame in a homogeneous coordinate system. R and t denote the rotation matrix and the translation vector, respectively. The part I represents the identity matrix. The terms K_l and K_r delineate the camera matrices, which describe the mapping of 3D points onto a 2D camera plane via the internal camera parameters, i.e. focal length, principal point, skew coefficient and the radial and tangential lens distortions. For further information about these parameters, the reader is advised to e.g. [27].

The normalized values D_{left} and D_{right} represent the Euclidean distances between the projection of the current position of a particle M on the left and right image frame, i.e. m_l and m_r, and the corresponding localization points from the vision system, MP_l and MP_r. This relation is defined as

$$D_{left} = \sqrt{\frac{(x_{MPl} - x_{m_l})^2 + (y_{MPl} - y_{m_l})^2}{width^2 + height^2}}, \tag{20}$$

$$D_{right} = \sqrt{\frac{(x_{MPr} - x_{m_r})^2 + (y_{MPr} - y_{m_r})^2}{width^2 + height^2}}, \tag{21}$$

where $width$ and $height$ are the width and height of the left and right image frames, as illustrated in figure 6.

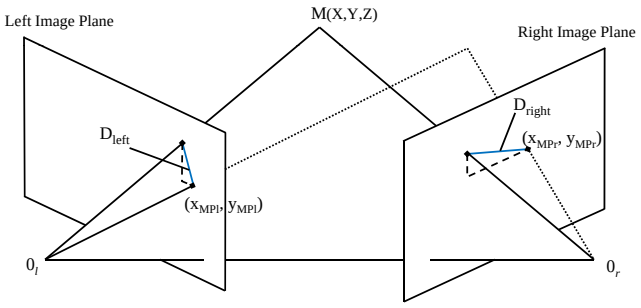

Figure 6: Model of a convergent stereo camera system showing both Euclidean distances D_{left}, D_{right} along with the position of a particle M and its corresponding projections.

5.2.3 Fitness Function

As explained in section 2, each particle tests the quality of its current position in every iteration by calculating its fitness function F. This function must be minimized, and therefore decrease when the particle's position is close to the object to be tracked in the solution space. The distances D_{left} and D_{right} measure the position quality with respect to the video-based localization module. Likewise, D_{audio} directly measures the quality with respect to the audio-based module. The fitness function F uses these three distance values together with additional weighting factors in the following way:

$$\begin{aligned} F &= w_{audio} \cdot D_{audio} + w_{vision} \cdot D_{vision} + \\ &+ w_{visbalance} \cdot D_{balance}, \end{aligned} \tag{22}$$

where w_{audio}, w_{vision}, and $w_{visbalance}$ denote weighting factors of each component. D_{vision} is defined as the sum

$$D_{vision} = D_{left} + D_{right}, \tag{23}$$

of the left and right distance value, giving the overall visual error. $D_{visbalance}$ is the absolute difference between the left and right distance,

$$D_{visbalance} = |D_{left} - D_{right}|. \quad (24)$$

This term forces the algorithm to prefer the particle with the minimum distance to both the left and right matching point, instead of only one.

Since the three distance values D_{left}, D_{right}, and D_{audio} are normalized between 0 and 1, the weighting factors can be used to adapt the fusion to the confidence level of the results provided by the audio and the vision system. This process is created dynamically, with w_{audio}, w_{vision} and $w_{visbalance}$ changing with each frame. For this purpose, an adequate variable that evaluates the confidence of the localization results is additionally provided by the audio and vision system, as explained in sections 3 and 4. Following the definitions of the normalized confidence values $Conf_{aud}$ and $Conf_{vis}$, the weighting factors can be defined as

$$w_{audio} = Conf_{aud} \quad (25)$$

$$w_{vision} = w_{visbalance} = Conf_{vis}. \quad (26)$$

The PSO fusion algorithm delivers a 3D location estimate of the tracked object and stops iterating when either a predefined maximum number of iterations has been executed, or a minimum value F_{min} for the fitness function F is reached, i.e. the convergence criterion is fulfilled. The 3D position $gbestPos(X,Y,Z)$ of the global best solution $gbest$ that has been encountered represents the current position of the tracked object.

The tracker has to cope with two opposed tasks: on the one hand, the particles must be prevented from getting stuck in a once obtained solution where they stop moving and tracking. On the other hand, successful tracking should include a prediction step instead of a new independent detection in each frame. The position of an object does not change dramatically within two time steps, which should be considered for fast convergence. In order to achieve these two characteristics, the particles are randomly re-initialized at new positions in each new time instant, except for the global best particle, which allows a global search in each instant. Furthermore, a subset of the swarm is positioned randomly in a limited space near the global best of the last frame, which fulfills the requirement of a position prediction independent of any model.

Thereby, the PSO tracking system has the advantage that a movement prediction can be considered, but the tracking will not fail with abrupt movements, as the particles not belonging to this subset will examine the other possible locations.

As its global searching behavior makes the PSO algorithm robust against local minima, deviations between the detected and the object's real position mainly result from localization errors of the audio and vision block caused by weak calibration or false detection of the preceding modules.

6 Results and Comparison

To test and evaluate our tracker, audio and video data of a person moving and talking in an area facing the stereo camera and stereo microphone system were taken. The hardware deployed consisted of two firewire cameras and two AKG omni-directional microphones. In a first implementation, the videos were taken with 15 frames per second with a resolution of 640×480 pixels. The audio material was recorded using a sampling frequency of 44100 Hz. For a single audio calculation step, we captured and processed four to eight windows for each microphone, with a FFT window length of 1024 samples and 50% overlap of the audio stream. This results in time frames of maximum 104.48 milliseconds. Furthermore we implemented an optimized version of the PSO and Kalman system using the MutanT development framework [29], which enables multithreaded execution of the different modules, permitting an efficient usage of the processor's cores. This version allowed online testing at 30 frames per second.

The tracker was implemented in C++ using $OpenCV$ library [30], and the $FFTW$ library was used for computing the fast Fourier transforms. The parameters of the PSO algorithm were set to $w = 0.7$ and $c_1 = c_2 = 2$.

6.1 Speed

Our algorithm was tested on three different computers, a notebook using a 1.6 GHz AMD processor with 512 MB RAM, a workstation using an Intel QuadCore with 3 GB RAM and a 2,5 GHz QuadCore notebook with 3 GB RAM.

Table 6.1 shows the mean speed performance of the PSO tracking on the different computers.

workstation	PSO (ms)	Kalman (ms)
AMD 1,6 GHz 512MB RAM	100,4	106,1
Intel Quadcore 2,6 GHz 3GB RAM	50,6	52,0
Intel Quadcore 2,5 GHz 3GB RAM using MutanT [29]	24,8	24,8

Table 1: Mean computation time of Kalman and PSO based tracking for different workstations and implementations in milliseconds.

Regarding the computation times, it should be mentioned that the vision system running two CAMshift trackers requires more than 60% of the execution time. Without the optimization, the audio system requires up to 30% for delivering the azimuth angle. These are preprocessing times which are added to the Kalman and PSO fusion techniques. Both Kalman and PSO based tracking modules use less than 10% of the execution time. In the MutanT-based implementation, both the Kalman and PSO Module needed a mean computation time of 2 milliseconds and the entire tracking system ran at the camera framerate.

Appropriate combinations of particles, iterations, and convergence criteria $Fmin$ requires equal or slightly shorter computational time than the Kalman reference system. It should be noted that the linear triangulation method combined with a linear Kalman system are one of the fastest state-of-the-art tracking systems today. More complex systems achieve similar tracking performance, under the same experimental conditions, yet demanding higher computational costs.

6.2 Accuracy

The MutanT version of the competing algorithms were tested online. To obtain ground truth data and evidence on the tracker's accuracy in the x and z dimension, we additionally recorded the position of the tracked object using a SICK LMS 210 LIDAR. The LIDAR delivers a position estimate every 100 ms with an accuracy of 0,5 degrees regarding the angular resolution and 0,015 m regarding the z direction. The x and z coordinate of the current position estimated by the PSO and the Kalman tracker were compared to the position obtained by the LIDAR. The two plots in figure 7 show the x and z positions (left) and the z positions over time (right) of an exemplary test run, with a moving object. The positions are recorded by the LIDAR, the Kalman, and the PSO tracking in each LIDAR measurement step. We performed several test runs with differently moving objects, all yielding similar results.

The mean and maximum errors in x and z direction, and the mean Euclidean error in meter obtained by the PSO tracker and the reference Kalman system for this tracking are shown in table 6.2.

	mean (m)		max (m)		Euclidean
	x	z	x	z	mean (m)
PSO	0,0577	0,0677	0,2130	0,2242	0.0997
Kalman	0,1085	0,1305	0,4034	0,3994	0.1867

Table 2: Mean and maximum error in x and z direction, and Euclidean mean error (in meter) of the PSO and Kalman tracker compared to LIDAR data.

These results show that while the Kalman system delivers a more smoothed track than the PSO in slow movements, the PSO tracking can adapt faster to position changes. This results in a slightly smaller overall error for the PSO tracker when an object moves with changing velocity and direction.

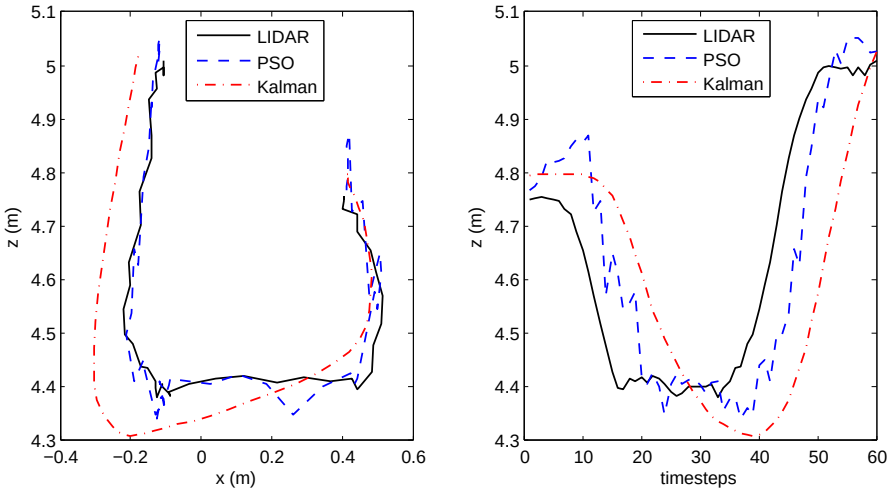

Figure 7: Tracking results as well as the LIDAR positions of the x and z coordinate (left) as well as the z coordinate over time (right).

6.3 Comparison

Comparable works on object tracking, using audio and video data and a variety of fusion techniques can be found in the literature, most of them using one to four cameras and microphone arrays with 8 up to 80 microphones. An object tracker similar to our reference system, using a decentralized Kalman filter structure was introduced in [31], yielding one 3D Position estimate per second. In [6], a Bayesian network was applied to a speaker detection problem, whose posterior probability distribution, was once computed exactly and then approximated using particle filtering. While speeding up the tracking procedure about ten times, the particle filter approximation still needed up to three seconds per frame, making the tracking an offline process. The graphical model based object tracker presented in [8] requires a computation time of three minutes to process one minute of video. In [7], a joint audio visual tracker based on a particle filter framework was implemented and used in video conferencing for speaker detection and tracking. Parallel processing was applied to achieve a performance of approximately eight frames per second, i.e. a computation time of about 125 ms per frame.

Another approach using a Bayesian network for fusion and human tracking was proposed in [9]. The tracking performance was tested using five different test sequences, and led to a computation times ranging from 2,0 to 2,3 seconds per frame.

7 Conclusion and Future Work

We presented a novel 3D object tracking system based on dynamically fusing audiovisual information with regard to the reliability of the single modules. Our PSO based fusion approach does not need any models, statistics or learning phase. It overcomes the problems of classic audiovisual fusion methods, that are based on assumptions regarding the distributions of variables, or that tend to become complex by reducing these limitations. Speed performance was shown to be slightly faster than the Kalman tracking, which can be regarded as the simplest of the existing systems. It outperforms more complex methods like particle filtering or Bayesian inference, which tend to become computationally expensive. A further advantage of our algorithm is that it avoids local minima and is insusceptible to false localization. While the addition of the sound source localization could not increase the accuracy of the 3D position detection, it increased the entire system's robustness regarding object occlusion.

Smoothness constraints can be adapted to the PSO system, allowing a straightened tracking while keeping the PSO ability to adapt to sudden changes. Furthermore, the PSO tracker can be extended and used for multi object tracking. Another possible enhancement is to include the information of additional sensors, for

example range sensors like LIDAR. For this purpose, the PSO systems fitness function can be easily adopted by adding the additional information to the fitness function and adapting the convergence criterion. Furthermore, the presented three dimensional particle search may be a replacement of existing triangulation functions and moreover a new and fast ability to create complex disparity maps from point correspondences.

References

[1] A. Leykin, R. Hammoud, Pedestrian tracking by fusion of thermal-visible surveillance videos, MVA Machine Visions and Applications Journal (2008) 1–9.

[2] N. Poh, J. Korczak, Hybrid biometric person authentication using face and voice features, Lecture Notes in Computer Science 2091 (2001) 348–353.

[3] A. Christian, B. Avery, Digital smart kiosk project, in: SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co, Los Angeles, California, United States, 1998, pp. 155–162.

[4] M. Brandstein, D.Ward, Microphone Arrays, Signal Processing Techniques and Applications, Springer-Verlag, 2001, Ch. 2/10, pp. 203–222.

[5] N. Checka, K. Wilson, Person tracking using audio-video sensor fusion, Sow procedings, Massachusetts Institute of Technologie Oxygen Research Group (2002).

[6] H. Asoh, et al, An application of a particle filter to bayesian multiple sound source tracking with audio and video information fusion, in: Proceedings of the Seventh International Conference on Information Fusion, International Society of Information Fusion, Mountain View, CA, 2004, pp. 805–812.

[7] D. N. Zotkin, R. Duraiswami, L. S. Davis, Joint audio-visual tracking using particle filters, EURASIP Journal on Applied Signal Processing 2002 (2002) 1154–1164.

[8] M. J. Beal, N. Jojic, H. Attias, A graphical model for audiovisual object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 828–836.

[9] X. Zou, B. Bhanu, Tracking humans using multi-modal fusion, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05 - Workshops, IEEE Computer Society, Washington, DC, USA, 2005, p. 4.

[10] F. Asano, et al, Detection and separation of speech event using audio and video information fusion and its application to robust speech interface, EURASIP Journal on Applied Signal Processing 2004 (11) (2004) 1727–1738.

[11] A. Noulas, B. Kröse, Probabilistic audio visual sensor fusion for speaker detection, Tech. rep., University of Amsterdam Intelligent Autonomous Systems (July 2006).

[12] R. O. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Transactions on Antennas and Propagation AP-34 (3) (1986) 276–280.

[13] J. Fritsch, M. Kleinehagenbrock, S. Lang, G. Fink, G. Sagerer, Audiovisual person tracking with a mobile robot, in: Proceedings of the International Conference on Intelligent Autonomous Systems, IOS Press, 2004, pp. 898–906.

[14] G. R. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technology Journal Q2 (1998) 15.

[15] F. Keyrouz, U. Kirchmaier, K. Diepold, Three dimensional object tracking based on audiovisual fusion using patricle swarm optimization, in: VDE 11th International Conference on Information Fusion, 2008, pp. 611–615.

[16] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.

[17] P. Saisan, S. Medasani, Y. Owechko, Multi-view classifier swarms for pedestrian detection and tracking, Computer Vision and Pattern Recognition Workshop 0 (2005) 18. doi:http://doi.ieeecomputersociety.org/10.1109/CVPR.2005.499.

[18] Y. Owechko, S. Medasani, Cognitive swarms for rapid detection of objects and associations in visual imagery, in: Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS, 2005, pp. 420–423.

[19] Y. Zheng, Y. Meng, Adaptive object tracking using particle swarm optimization, in: Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Jacksonville, FL, USA, 2007, pp. 43–48.

[20] H. Teutsch, Wavefield decomposition using microphone arrays and its application acoustic scene analysis, Ph.D. thesis, TU Erlangen (2005).

[21] C. H. Knapp, G. C. Carter, The generalized cross correlation method for estimation of time delay, IEEE Transactions on Acoustics, Speech and Signal Processing 24 (4) (1976) 320–327.

[22] V. C. Raykar, Csmc 660 project solutions optimization methods for sound source localization using microphone arrays, Tech. rep., University of Maryland Institute For Advanced Computer Studies (December 2002).

[23] R. Isermann, Identifikation dynamischer Systeme 2, 2nd Edition, Springer-Verlag, 1992.

[24] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Transactions on Information Theory IT 21 (1) (1975) 32–40.

[25] R. Belaroussi, M. Milgram, Face detection and skin color based tracking : a comparative study, in: Proceedings of the 2007 International Conference on Image Processing, Computer Vision and Pattern Recognition, IPCV, CSREA Press, 2007, pp. 506–512.

[26] M. J. Swain, D. H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 1727–1738.

[27] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[28] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME - Journal of Basic Engineering 82 (1960) 35–45.

[29] S. Hawe, U. Kirchmaier, K. Diepold, Mutant: A modular and generic tool for multi-sensor data processing, in: 12th International Conference on Information Fusion, Seattle, 2009.

[30] Intel, Opencv computer vision library (2006).
URL intel.com/technology/computing/opencv/index.htm

[31] F. Talantzis, A. Pnevmatikakis, L. C. Polymenakos, Real time audio-visual person tracking, in: IEEE 8th Workshop on Multimedia Signal Processing, 2006, 2006, pp. 243–247.