A hybrid algorithm for tracking and following people using a robotic dog

Liem, M.C.; Visser, A.; Groen, F.C.A.

Publication date
2008

Published in
HRI 2008: Proceedings of the Third ACM/IEEE Conference on Human-Robot Interaction

A Hybrid Algorithm for Tracking and Following People using a Robotic Dog

Martijn Liem
Instituut voor Informatica,
Universiteit van Amsterdam
Kruislaan 403 1098SJ
Amsterdam, The Netherlands
mliem@science.uva.nl

Arnoud Visser
Instituut voor Informatica,
Universiteit van Amsterdam
Kruislaan 403 1098SJ
Amsterdam, The Netherlands
arnoud@science.uva.nl

Frans Groen*
Instituut voor Informatica,
Universiteit van Amsterdam
Kruislaan 403 1098SJ
Amsterdam, The Netherlands
groen@science.uva.nl

ABSTRACT

The capability to follow a person in a domestic environment is an important prerequisite for a robot companion. In this paper, a tracking algorithm is presented that makes it possible to follow a person using a small robot. This algorithm can track a person while moving around, regardless of the sometimes erratic movements of the legged robot. Robust performance is obtained by fusion of two algorithms, one based on salient features and one on color histograms. Re-initializing object histograms enables the system to track a person even when the illumination in the environment changes. By being able to re-initialize the system on run time using background subtraction, the system gains an extra level of robustness.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics — *Commercial robots and applications*; K.4.2 [**Computers and Society**]: Social Issues—*Assistive technologies for persons with disabilities*

General Terms

Algorithms, Design, Measurements, Experimentation

Keywords

Robot companion, awareness and monitoring of humans

1. INTRODUCTION

A very active area of research within image processing is the tracking or following of people. Many algorithms for finding and keeping track of people walking around have been proposed [4, 6, 9, 10, 21], but are mostly based on static (surveillance) cameras. The next logical step would be to combine a tracking algorithm with a mobile robot and to make the robot follow people around an area. There are many possible applications for autonomous person following by robots that can be thought of. Not only could it be useful in a domestic environment, but it could also be used in offices, museums and hospitals.

Figure 1: Sony AIBO robot dog in a domestic environment.

While in domestic environments a following robot could be used to assist people while they move around the house, for example carrying serving trays, more socially relevant applications could be found in elderly care. Since the number of elderly people is growing rapidly, the need for nursing personnel is bound breaking. It could be a great improvement when people needing intensive care could be monitored without a human constantly checking the current status of the person. The same could be done using a wearable alarm button or camera surveillance, but as people forget to wear the buttons and dislike the idea of being watched all the time, these methods are not optimal. It would be much better when the monitoring could be done using unobtrusive methods.

The Sony AIBO robot dog offers such an unobtrusive method. It can be seen as a multi-sensor platform embedding, among other sensors, a camera, stereo microphones and infra-red distance sensors in a dog-like four legged robot. Because of the appearance of the robot, people tend to regard it much more like a toy dog puppy than a camera surveillance system (see figure 1)[1]. Furthermore, as long

*The authors were supported by EU Integrated Project COGNIRON ("The Cognitive Companion") FP6-002020.

HRI'08, March 12–15, 2008, Amsterdam, The Netherlands.

[1]Picture published by the courtesy of the AIBO Research Team, http://aibo.telin.nl/.

as the robot can work autonomously and no actual person is needed to watch the video streams all day, using this kind of monitoring device will bring about fewer privacy issues.

In this research, the Sony AIBO robot dog is used for tracking and following a person around an area. The robot should be able to automatically locate the person and initiate tracking. The robot is able to perceive its environment using a small camera mounted in its nose. To be able to do this, multiple algorithms will be combined into a fusion algorithm in which the strong points of the algorithms compensate for the weaker points of the others. Following behavior will be exhibited due to a feedback loop between the vision based tracking method and the movements of the robot.

In the next few sections, a concise survey on object tracking and person following will be given. Next, a combination of two state-of-the-art algorithms will be described that allows for robust tracking and following of a person using a Sony AIBO dog. Following the description of the algorithm, the results of our experimental tests will be given. The implications of this research will be outlined in the conclusion.

2. RELATED WORK

In recent years, much research has been done related to computer vision-based person tracking. Some of the methods researched are directly related to the specific object that needs to be tracked. Examples of these methods are template matching [8, 14], shape fitting [1] and human modelling [25]. These methods rely on the recognition of a human shape in an image to detect and possibly track a person through a scene. All these methods need a general understanding of the shape or composition of a human body.

Many object tracking methods rely on the segmentation of the object or person from the image. Segmentation methods like simple background subtraction [21, 23, 27] or more advanced C-means clustering [3, 16] should be used to extract the objects before classification can be performed. Because people are neither uniformly colored nor textured, C-means clustering will presumably result in multiple clusters representing one person. In this case, it will be very cumbersome to find out which parts together make up a person, which makes it impractical to utilize when a complete person should be tracked.

One frequently used method for locating people in a scene and segmenting them from that scene is background estimation and subtraction [21, 23, 27]. Like many methods for object segmentation and tracking, this type of segmentation method relies on a static camera, like those used in most surveillance situations. The method is based on estimation of the distribution (Gaussian Mixture Model) of the RGB values of each pixel and classification into background and foreground. While the robot, and therefore the camera, is moving around, the assumption that the background is static does not hold. However, when the robot stands still, background subtraction could be a useful method. It could be used to initialize other tracking algorithms by using background estimation to segment objects from the scene and to provide data on these objects to other methods. Taking into account a short period in which the robot needs to stand still will not necessarily pose a problem.

Object displacement can also be detected using salient points [19, 24, 12]. Salient points are features in an image that can easily be tracked due to their structure. Selected points could be corners, edges or more complex textures like triangles or salt-and-pepper figures. Several of those feature points are located in the image and tracked to the next frame. Based on those tracks, displacements in that part of the image can be estimated. Optical flow methods [11, 15] can be used to combine those displacements. This kind of method can also be used to describe a dense motion field in which the displacement of each separate pixel is estimated on a frame to frame basis. In this case, all pixels can be seen as features of which the motion is estimated. Optical flow methods work well in combination with a moving camera. They can be used to estimate the movement model of the background of a scene. Regions in the image where the movement deviates from that model can be indicated as objects and segmented from the background. A method like this could be used to find and track persons using a mobile camera. It could also be used to get the features for initializing another tracking method, as suggested in [24] or estimate the new position of a known object.

An application using feature points in object tracking can be found in [13]. This paper describes a way to use "flocks of features" for tracking a hand in front of a mobile camera, making use of the Kanade-Lucas-Tomasi (KLT) feature point tracker. The tracker is initialised using skin color segmentation. Tracking is done by making use of the spatial relationship between the features found. A disadvantage of the method is that it makes use of pre-initialised thresholds determining the size of the hand. This is not a problem as long as the maximum distance between the object to be tracked and the camera is kept small, which is achieved by mounting the camera on the person. In our case, the distance between camera and person is more flexible which will result in many more fluctuations in relative object size.

An efficient method for tracking complete objects is by tracking the color histogram of the object. A well known example of such a method is the mean-shift algorithm described in [4]. An extension of this method described in [26] introduces an adaptive version which enables adjusting the search window. The algorithm allows to grow or shrink a Gaussian kernel around the object to be tracked. Furthermore, the orientation of the object can be estimated, which allows this method to detect, for instance, whether a person being tracked is standing up or lying down. Another advantage is that the method is robust to camera movement, including rotations.

In [23], another method for color histogram tracking is presented. This method uses a blob representation of the person instead of a kernel estimation. It is also robust to deformable object shapes due to the distinction between the object center and a deformable boundary around this center. In our case the usage of an object blob segmented from the background is a disadvantage. As discussed before, in the case of a moving camera it is difficult to segment a complete person. While it is possible to track multiple segments of a person separately, clustering could result in many different objects (shoes, pants, shirt, arms, hands, head, hair) and tracking all these object would be very resource intensive.

Several projects have also aimed to find ways to follow people using a mobile robot. In [18], multiple people are tracked using a laser-range scanner mounted on a robot. Data about people's positions gathered this way can easily be used for person following or avoidance. In [20], [17] and [7], mobile robots are used to locate and follow people using various vision- as well as audio-based algorithms.

All of these projects use advanced robots equipped with high quality cameras or laser-range scanners. Furthermore, wheeled, child-high robots are used for all experiments. At those points, the experiments largely differ from the ones presented here; since the AIBO can be regarded as a much more low-end, low-profile robot which furthermore is propelled using four legs instead of wheels. This results in much less stable sensor data as soon as the robot starts moving. These factors request a robust method which uses as much information from the provided data as possible.

3. HYBRID ALGORITHM

Inspired by the methods described in section 2, a number of algorithms is combined into a hybrid tracking algorithm. The most important requirements for the tracking algorithm are that it should be robust to unpredictable and irregular camera movement, not too sensitive to changes in illumination common in a domestic environment and not reliant on explicit body markers to be worn by the person to be tracked. Furthermore it was decided to use only camera images for tracking, since the infra-red distance sensors provide very sparse and noisy data while audio is not very intuitive to use for tracking.

The algorithms were selected based on the way they perform tracking. For initialization purposes, the Gaussian Mixture Model (GMM) Background Subtraction method described in [27] was used. This method makes a fast and accurate segmentation of foreground and background objects in a given scene. Therefore, this method is very useful for making an initial segmentation of the person to be tracked, after which the other algorithms can be instantiated using the information from the segmentation. At a later moment, the algorithm can be used to re-initialize the other trackers or to support them during tracking.

For the purpose of tracking a person while the camera is moving, the Expectation Maximization (EM)-based color histogram tracker from [26] is used. This method needs a reference color histogram from the person to be tracked, and uses the EM algorithm [5] to track a Gaussian kernel from the previous location of the person to the estimated new location of the person. Tracking is done by comparing the reference histogram with the histogram at the estimated new location. In this hybrid algorithm, $8 \times 8 \times 8$ RGB color histograms are used.

Additionally, the Kanade-Lucas-Tomasi (KLT) algorithm described by [15, 19, 22] is used. This method uses its own kind of salient features to track from one frame to another. Therefore, this method is completely independent from image color tracking or global image differences. By tracking about 150 features from frame to frame, the algorithm can give a accurate estimation of the new position of an image feature in a new frame.

A combination of these methods will be necessary because by themselves they are unable to perform stable tracking. The background estimation algorithm works well as long as there is a static background in the image. As soon as the camera starts moving however, the background will become unstable and the algorithm will no longer be able to make an accurate estimation of foreground and background objects. This means it can be used well as an initialization method as long as the robot is standing still, but will be of no further use as soon as the robot starts moving to follow the person.

At this point the EM-shift method can be used to track the person while he or she is moving, followed by the robot. The problem at this point is the dependency of the method on the perceived colors. When the person walks past a background object having a color similar to a color from the person's histogram, the algorithm will get confused and has a chance to snap onto the background object. Another cause for the algorithm to lose track is changes in illumination. Those changes will cause colors to be perceived differently from the ones stored in the reference histogram, which will also cause the algorithm to lose track. As a last point, the EM-shift algorithm needs to have part of the person being tracked inside the tracking kernel when it is initialized on a new movie frame. When this is not the case, the algorithm will be unable to estimate in which direction it should move to refit on the person.

To compensate for the problems considering the color histogram tracker, the KLT tracker is used as extra support. While this method is capable of tracking a large number of features from one frame to another, it does not segment real-world objects using these features. The points tracked tend to be an abstract notion of an image feature, without any direct relation to what they represent in the image. This makes it impossible to decide which features belong to the same object in an image. Feature clustering could be done using their relative optical flow, but this method is not guaranteed to give reliable results. Another method should be used to detect to which object certain features belong. A last problem with the KLT tracker is that for every new frame, feature points will be filtered out. To continue tracking over longer periods, new feature points should be generated for every frame. The EM-shift method can be used to select new feature points on the person to be tracked.

The EM-shift algorithm and the KLT tracker can be put together in a feedback loop which makes them support each other. After initializing the system using background subtraction BS, a segmentation mask with a location θ_r and a shape V_r is available which represents the moving person O_r. The KLT algorithm starts by looking for salient features located all over the image. In the next step, the segmentation mask resulting from background subtraction can be used to select only those feature points P_r located on the person. Furthermore, the mask is used to get the reference color histogram H_r from the person.

At this point, the feature points selected are tracked to the next frame. For each feature point the reliability of the match is estimated. Some points will be ignored, because the match is not good enough (the feature is no longer clearly visible). Other points can be matched well enough, but with an unrealistic displacement vector (the feature is visible in the image, but not on the tracked person). These points will be dropped after the EM-shift algorithm has been applied. Typically, for each frame 50% of the selected feature points can be tracked reliably to a new location. The position and shape of the person can be estimated by taking the mean and the covariance of the set of feature points. The average position can be used as a reasonable estimate for the person's position, although there is no guarantee that the position is exactly centered. The covariance is not a good estimate for the shape of the person, because this region is likely to be smaller than the actual person, due to the many feature points that have been lost. If this process continues for a few frames, the KLT tracker would slowly lose all feature points to track, and should be re-initialized with the background

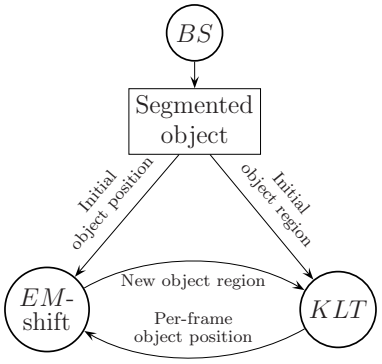

Figure 2: Interaction between the Background Subtraction (BS), EM-shift and KLT algorithm.

subtraction. Fortunately, the EM-shift algorithm can be used to estimate the area covered by the person, and suggest new feature points to be tracked.

To start the EM-shift algorithm, an estimation of the initial object position and shape is needed. Normally, the previous known position is used, but this makes the method sensitive to large object motion. Instead, the initial estimate of the person is based on the mean and covariance of the features tracked by the KLT tracker. Using this initialization, the EM-shift algorithm will start to find an optimal fit regarding the similarity between its reference color histogram and the color histogram of the area below the current kernel. The EM-shift kernel will grow and shrink until if finds a good enough match and returns the newly estimated mean and covariance. This estimation can now be used to locate new feature points in, which can then be tracked to the new frame.

Figure 2 shows a diagram of the hybrid algorithm. In this diagram it is made clear how the three algorithms work together and what kind of information is exchanged between the algorithms. In principal the KLT algorithm supports the EM-shift algorithm with a good starting position to search for the shape with a color histogram matching the reference H_r. On the other side the EM-shift algorithm supports the KLT algorithm each frame with an independent estimate of the area where the person can be found, which allows to compensate for lost feature points.

The detailed steps of the algorithm are illustrated in figure 3. In this figure, the segmented object of the BS algorithm, the location and shape of the EM-shift algorithm and the feature points of the KLT algorithm are shown on top of the images that produced those features. Figure 3.1 displays the original image I_r after the background model is learned (which typically takes a few frames). Figure 3.2 displays the pixels where image I_r is different from the background model in different gray-values, representing a moving person. With standard image processing techniques like erosion and dilation, those images can be segmented giving a binary mask. This binary mask can be used to get a reference color histogram H_r, as illustrated in figure 3.3. In this case the color histogram would contain mainly light blue of the trousers and dark blue of the sweater. Notice that the chairs in the background are dark blue as well. In figure 3.4 the ellipse indicates the shape of the Gaussian kernel which is found by the EM-shift algorithm for H_r. The binary mask generated from the segmented object can also be used to generate a set of feature points P_r. In figure 3.5 thirteen white feature points are drawn. Eight of those feature points could be tracked to the next frame, as illustrated in figure 3.6. The average position of those points is a little bit higher and to the left of the previous estimate of the moving person O_r. This position is used to initialize the search of the EM-shift algorithm, as indicated with the ellipse in figure 3.7. The shape of the Gaussian kernel is adjusted so that its color histogram contains nearly the same distribution of colors as H_r. The result of this adjustment is illustrated in figure 3.8. The result of the EM-shift algorithm is used to select a number of new feature points P_j on the moving person. The new feature points P_j are indicated with white stars in figure 3.9. Those eighteen feature points are tracked to the next frame, as illustrated in figure 3.10.

At some point, the tracker will probably no longer be able to keep track of the person, due to lost features or color conflicts. When it is likely that the object being tracked is not the person, the robot should no longer try to follow the tracker. Instead, it should try to relocate the person and re-initialize the tracker. Detection of a lost track can be done by using the similarity measure from the EM-shift algorithm. When the tracker moves away from the object, it is likely that the maximum similarity that can be found between the reference histogram and the current histogram is not as high as when the correct object is tracked. By putting a threshold on the minimal similarity needed to be certain that the correct object is tracked, the system can be signalled when re-initialization is needed. It is assumed that erroneous tracking behavior is detected soon enough to assume the person is still visible in the robot's field of view. Therefore, when a bad track has been detected, the robot's movement will immediately be halted after which background estimation is executed. A moving person can now be distinguished from the background. This gives a new segmentation mask which is used to update the EM-shift reference histogram and find new feature points. The reference histogram is updated in such a way that a small amount of the previous histogram is still left in, which makes the tracker more robust to changes in object color.

Sometimes, background estimation can also be used during tracking. At moments when the robot is standing still and is not moving its head, background subtraction is done to provide an extra support for the tracker. After a few frames of background estimation, the largest segmented object is selected and compared to the current tracker location. When at least 25% of the pixels in the current kernel overlap with the background subtracted object and the current kernel contains more than 25 pixels, the background subtraction information is used to adjust the current tracker position.

4. MOVING THE ROBOT

Besides tracking the person through the image sequence, the complete algorithm (see algorithm 1) should also include the controls that direct the robot to actively follow the human. This is done in two stages. The first stage is to control the head of the robot to try to keep the center of the person in the center of the camera. The second stage is to control the body of the robot. The first stage is performed by controlling the pan/tilt head movement of the AIBO based on the position of the person in the image. When the x and

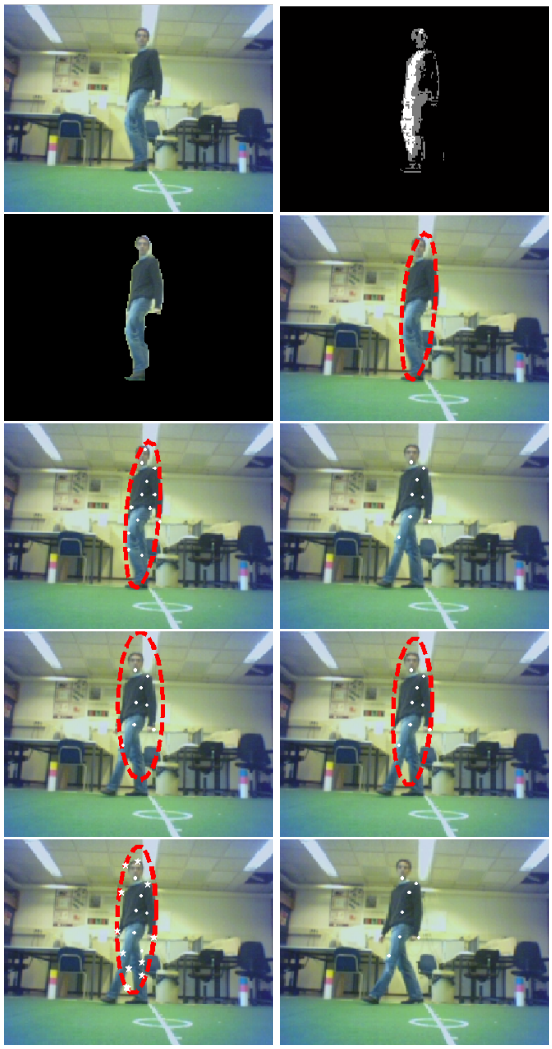

Figure 3: Complete tracking procedure, left to right, top to bottom: 1) the original image, 2) background subtraction, 3) object segmentation used to get reference color histogram, 4) object region estimated by EM-shift, 5) feature points inside the initial object region, 6) track feature points to the next frame, 7) initialize EM-shift kernel on mean and covariance of tracked feature points, 8) execute EM-shift to find new person position, 9) search new features on object, 10) track features to next frame.

y coordinates of the position are normalized by setting the width and the height of the image to 1, the horizontal and vertical viewing angle of the camera can then be used to calculate the necessary head movement. The head-pan should be adjusted using half the horizontal viewing angle, multiplied with the normalized, relative x position of the object, while the head-tilt should be modified with half the vertical viewing angle multiplied with the object's normalized, relative y position.

To be able to follow the person being tracked, the dimensions of the AIBO should be taken into account. The AIBO is a small robot. When the robot is standing straight up, the camera mounted in his nose is about 22 cm above the floor. To get a complete view of a person with a height of 180 cm at a distance of 200 cm, the head of the AIBO should be tilted by 16°. Because the tracker will work best at the closest distance at which the complete person can be seen, the tilt of the head can be used to maintain an optimal distance between the robot and the person. When the head is tilted less than 16°, the distance between the robot and the person is probably larger than 2 meters and the robot should walk forward until the tilt is within range again. As soon as the head-tilt is larger, the distance is too small and the robot should walk backwards.

This method works fine because the robot only has to follow a human. The precise distance is not really an issue, as long as it is not too close to be uncomfortable. Because of the pre-set tilt threshold, smaller people will automatically be followed at a slightly closer distance than taller people. The robot will also adjust its distance as soon as the person being followed bends through its knees, for instance.

To follow a human, the robot should also be able to turn its body. The robot should turn its body to prevent that the person moves out of the field of view of the robot. An AIBO can move its head 186° horizontally, so a person will be lost when the head should be panned more than 93° to the left or the right. Therefore, when the pan of the head becomes larger than 35°, the robot is signaled to turn itself in the direction of the person. This takes about two seconds, so starting to turn should be done well before the head-pan reaches its maximum. Because one standard turn of the AIBO in the Universal Real-time Behavior Interface (URBI) is about 35°, this is the most logical head-pan threshold.

When the robot moves, it constantly keeps tracking the person and updating its head position to keep the person centered. Because the movement of the AIBO is very shaky, the head-tilt threshold will often be exceeded without the robot needing to change its distance. To prevent unnecessary movement to keep its distance, the head-tilt will be averaged over the last five frames, so extreme head movement because of the shaky walk will not influence the robot's movement. Furthermore, during its following behavior it is likely the panning threshold will be reached as well. When this happens, the robot is first stopped to get a better estimate of the person's current position, after which the turning sequence is started. When the head-pan is below the threshold again, the robot is allowed to continue move forwards or backwards.

During the walking sequence, the shaky head movement makes it hard to make an accurate estimation of the distance between the person and the robot. Therefore, the robot is briefly halted every fifteen steps to be able to re stabilize its view and get a better estimate of the distance between the

Algorithm 1: The hybrid fusion algorithm

Input: Image sequence containing a person walking through a room
Output: Robot following the person

while *true* **do**
  Analyze the first images $I_1 \dots I_r$ with background subtraction BS;
  Learn to separate a large moving object O_r from the background;
  Estimate from the location θ_r and shape V_r of object O_r;
  Initialize EM-shift by memorizing reference color histogram H_r of object O_r;
  Initialize KLT by selecting feature points P_i on the object O_r;
  while *current color histogram* H_i *matches* H_r **do**
    Track feature points P_i to next frame $j = i + 1$ using KLT;
    Estimate location θ_j with KLT;
    Estimate shape V_j with EM-shift starting at location θ_j;
    Select extra feature points P_j in the shape V_j;
    Adapt head position of AIBO to center object position in image;
    if *AIBO head tilt exceed threshold* **then**
      Begin moving AIBO;
    else if *AIBO head pan exceed threshold* **then**
      Begin turning AIBO;
    else if *Head position within limits* **then**
      Stop AIBO movement;
  Stop all AIBO movement;

person and the robot. This also helps the tracker not to get lost due to the erratic movement of the camera.

5. RESULTS

Experiments using various movie sequences as well as real-time image streams indicate the soundness of our approach. The movie sequences are collected from both a static robot and a moving robot. Tests using a moving camera were done in two ways, using direct active vision on the AIBO while it walks around on its legs as well as using an AIBO mounted on top of a wheeled robot. This last combination results in a driving system where the AIBO uses its head to track the person while the wheeled robot executes the movement commands normally performed by the legs. This test setup was used to determine the influence of the AIBO's erratic way of moving on the results obtained. Tracking quality is determined by computing the overlap between a hand generated ground truth and the tracking kernel in each frame. The overlap between the tracking kernel and the ground truth, relative to the size of the tracking kernel, determines the quality.

Camera images from the AIBO robot are taken with a small camera in the nose of the robot, at a resolution of 208×160 pixels and a frame rate of about 15 fps. Real-time experiments were done by streaming the images taken by the AIBO to a PC using a WiFi connection. Image processing was done on an AMD 3500+ processor with 2 Gb of RAM. The algorithms consist of a combination of C++ code interfaced with Matlab 7.1. For the basic algorithms, implementations from [27], [26] and [2] were used and adapted to work together. After processing the images, robot control commands are sent back to the AIBO using WiFi. The Universal Real-time Behavior Interface (URBI) was used as the robot control interface.

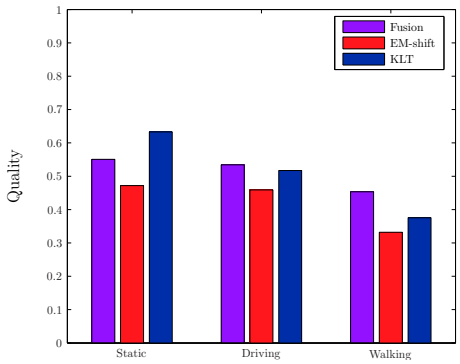

Figure 4: Average quality of the three algorithms, tested on static movies, movies taken using the driving robot and movies taken on the walking AIBO.

The static movie sequences are recorded to be able to compare our results with previous results. As can be expected, both basic algorithms show a good performance for the static case, as indicated in figure 4. It can also be seen that in all cases, the EM-shift tracker is outperformed by the KLT tracker. As long as the color distribution of the person is unique within the direct surrounding area, the algorithm has little trouble tracking the person. This is reflected in the EM-shift result for static movies shown in figure 4. This bar shows that the average match between the ground truth and the EM-shift kernel is over 50%. Tracking becomes harder as soon as matching colors are found in the scene, as illustrated in figure 5 (top row). This figure shows how the dark blue chairs in the background are slowly included in the kernel of the EM-shift algorithm, until at the 4th frame a combination of the light blue trouser and the chairs is tracked. Similar colors in foreground and background make it hard for the algorithm to stay focused. The tracker either gets stuck on the wrong object or the kernel starts growing in such a way that much more than the person is covered by the kernel. In the latter case, while the person is still within the track, the kernel does no longer provide a useful estimate of the person's position. Because the color distribution of the person can not be guaranteed to be unique in a natural setting, and the tests with the moving robot are performed in the same environment, equivalent problems are encountered for movies taken with the moving robot.

When considering the static movies, the basic KLT algorithm gives the best performance. Because the main sensitivity of this algorithm lies within erratic or swift object movement, which is very limited for the static movies, the stand alone KLT tracker is able to outperform both other trackers, which both suffer from color similarities. The more movement is introduced, the less reliable KLT can track by itself. While KLT quality is almost 0.65 for static movies, it ends up almost 40% lower at just below 0.4 when the robot moves around on its legs.

The real strength of the fusion algorithm becomes clear when movement is introduced. The tracker is able to stay focused on the person for a much longer period of time, can handle erratic and swift object movements and prevents unlimited kernel growth by combining the strength of both

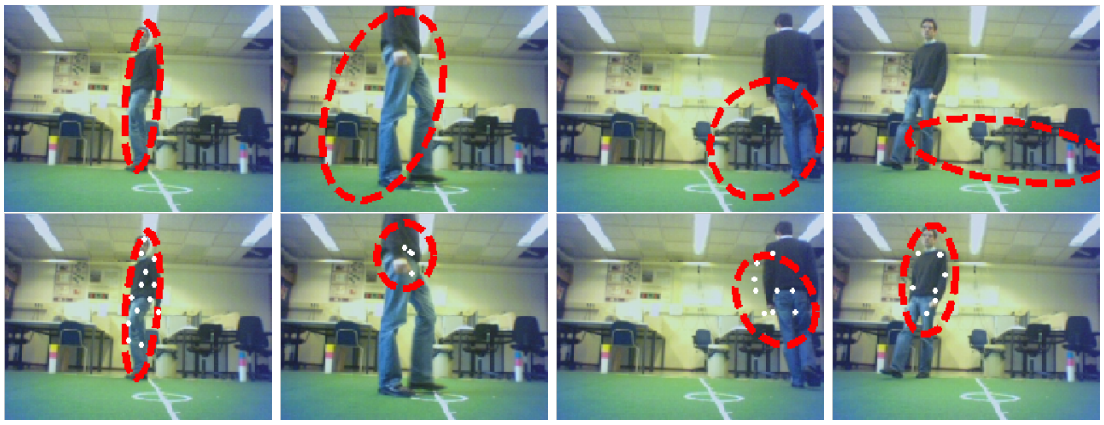

Figure 5: Comparison between EM-shift only tracking (top) and combined KLT and EM-shift tracking (bottom). The fusion algorithm clearly shows its stability.

	Walking	Driving	Static
Avg. # of frames	317	313	160
Avg. interval length	12.96%	16.59%	26.74%
Fr. fuse $\geq$ EM-shift	80.23%	79.11%	66.24%
Fusion/EM-shift	1.4300	1.2155	1.3112
Fr. fuse $\geq$ KLT	77.06%	67.89%	42.89%
Fusion/KLT	1.1175	1.0733	0.9671

Table 1: Statistics on quality measurements. Average number of frames for each movie type, average length of each interval between two reinitialisation moments, amount of frames for which the fusion algorithm has equal or higher quality then the EM-shift algorithm, average quality of the fusion algorithm compared to EM-shift, amount of frames for which the fusion algorithm has equal or higher quality then the KLT algorithm and average quality of the fusion algorithm compared to KLT.

basic algorithms. The region found using color information helps to keep the feature points distributed over the person, while the center location of the person found using the feature points keeps the kernel focused. An illustration of the stability can be seen in figure 5 (bottom row). As is show in figure 4 the quality of the fusion algorithm only degrades slightly when the robot starts moving. This means that performance of the fusion algorithm relative to the basic algorithms increases largely, as indicated in table 1. The results in this table show that the fusion algorithm performs better than the KLT algorithm for about 77% of the frames when the robot is walking, while the fusion algorithm outperforms the EM-shift algorithm for even 80% of the frames, with a total improvement of a factor 1.43.

When comparing the results of the walking robot and the driving robot in table 1, the influence of the AIBO's erratic movement pattern on the tracking quality becomes clear. Movement also has benefits. Because the robot uses active vision, it can make sure that the object is always in the center of the image, which makes tracking easier. On the more stable moving platform (driving robot), the basic algorithms can also benefit from the active vision. In this case the difference in quality between the basic algorithms and the fusion algorithm is less evident (respectively a factor 1.07 and 1.21). These results show that the strong points of the fusion algorithm can specifically be found when tracking is done on less stable moving platforms.

The remaining difficulty is related to tracking fast moving objects. Mainly due to the load of the multiple trackers on the testing system, the maximum speed at which new frames can be processed is about 2 fps. This means that there is a timespan of about 0.5 seconds between two frames. This timespan allows a fast person such a large displacement that search windows are initiated far off the actual location. Therefore, the system is sensitive to fast movement. Using a faster (multi-core) system for testing purposes will probably improve the results, since this will allow higher frame rates and thus smaller object movement in-between frames. Part of this can of course also be achieved by improving the efficiency of the algorithms used.

As long as the lighting conditions of the environment in which the person moves around are constant, EM-shift and KLT, supported by the movement of the robot, are very well able to keep a good track. At the moment a more severe change in illumination occurs, the difference between the EM-shift reference histogram and the current histogram gets too large and the algorithm starts re-initializing. In most cases, the track of the person will still be quite good, so a temporary stand-still of the robot is unlikely to allow the person to move outside the field-of-view of the robot. In this case, re-initialization can be seen as a mere update of the reference color histogram. This will allow the tracker to maintain a stable track for a longer period of time later on.

A collection of movies recorded during the experiments described above can be found online[2]. The experiments were performed in our laboratory with no special care for the illumination, reflecting the changing circumstances in a domestic environment.

6. CONCLUSION

In this paper, a tracking algorithm is presented that is able to follow a person using a small robot. This algorithm can track a person while moving around, regardless of the sometimes erratic movements of the robot. By being able to re-initialize the system on run time using background

[2]http://www.science.uva.nl/~arnoud/education/mliem/

subtraction, the system gains an extra level of robustness. Re-initializing object histograms enables the system to track a person while the illumination in the environment changes.

This fusion algorithm is combined with the robot control system in a feedback loop, which enables the robot to react to things perceived. The tracker is supported by this feedback loop because the object being tracked is kept in the center of the robot's view. This compensates for displacements between frames which would otherwise be too large to track well. By using head motion of the robot controlled by the position of the tracker in the image, the robot is able to estimate its distance with respect to the person being tracked and adapts its distance accordingly.

Experiments show that the algorithm works well and the robot is able to follow a person through a room. Because of the sensitivity of the camera used, the system performs less in low-light regions. This is a problem of the AIBO and not of the algorithm used however. While lighting conditions are sufficient for the camera to get a good look of the scene, tracking will work fine and the following behavior exhibited by the AIBO is excellent.

7. REFERENCES

[1] A. M. Baumberg and D. C. Hogg. Learning Flexible Models from Image Sequences. In *Third European Conference on Computer Vision*, volume 1, pages 299–308, 1994.

[2] S. Birchfield. KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker, 1997.

[3] J. Chen, T. Pappas, A. Mojsilovic, and B. Rogowitz. Adaptive Image Segmentation Based on Color and Texture. *ICIP*, 2002.

[4] D. Comaniciu and P. Meer. Real-Time Tracking of Non-Rigid Objects using Mean Shift. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:142–149, 2000.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 1(39):1–38, 1977.

[6] D. A. Forsyth and M. M. Fleck. Body Plans. *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 678–683, June 1997.

[7] J. Fritsch, M. Kleinehagenbrock, S. Lang, G. A. Fink, and G. Sagerer. Audiovisual Person Tracking with a Mobile Robot. In *Proc. Int. Conf. on Intelligent Autonomous Systems*, pages 898–906, March 2004.

[8] D. M. Gavrila. Pedestrian Detection from a Moving Vehicle. *Proc. of European Conference on Computer Vision*, pages 37–49, 2000.

[9] D. M. Gavrila and V. Philomin. Real-time Object Detection for "Smart" Vehicles. In *Proc. of IEEE International Conference on Computer Vision*, volume 1, pages 87–93, 1999.

[10] I. Haritaoglu, D. Harwood, and L. S. Davis. W^4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, Augustus 2000.

[11] B. K. P. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[12] T. Kadir and M. Brady. Saliency, Scale and Image Description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[13] M. Kölsch and M. Turk. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In *2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, volume 10, page 158, 2004.

[14] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving Target Classification and Tracking from Real-time Video. In *4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 8–14, 1998.

[15] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[16] D. Pham and J. Prince. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. *Proc. SPIE Medical Imaging 1998: Image Processing*, 3338(2):555–563, 1998.

[17] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Wörz. Vision Based Person Tracking with a Mobile Robot. In *Ninth British Machine Vision Conference*, pages 418–427, 1998.

[18] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters. *Int. Journal of Robotics Research*, 22(2), Februari 2003.

[19] J. Shi and C. Tomasi. Good Features to Track. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[20] H. Sidenbladh, D. Kragić, and H. I. Christensen. A Person Following Behaviour for a Mobile Robot. In *IEEE International Conference on Robotics and Automation*, pages 670–675, 1999.

[21] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision Pattern Recognition*, pages 246–252, 1999.

[22] C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[23] P. J. Withagen. *Object detection and segmentation for visual surveillance*. PhD thesis, Universiteit van Amsterdam, 2005.

[24] W. Zajdel, Z. Zivkovic, and B. Kröse. Keeping Track of Humans: Have I Seen This Person Before? *ICRA 2005*, pages 2081–2086, April 2005.

[25] L. Zhao. *Dressed Human Modeling, Detection, and Parts Localization*. PhD thesis, The Robotics Institute, Carnegie Mellon University, July 2001.

[26] Z. Zivkovic and B. Kröse. An EM-like algorithm for color-histogram-based object tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:798–803, 2004.

[27] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, May 2006.