EXTRACTION OF MOVING OBJECTS WITH A MOVING MOBILE ROBOT

Marc Ebner *

* *Wilhelm-Schickard-Institut für Informatik*
Eberhard-Karls-Universität Tübingen
Arbeitsbereich Rechnerarchitektur
Köstlinstraße 6, 72074 Tübingen, Germany
ebner@informatik.uni-tuebingen.de

Abstract: An algorithm for the extraction of multiple moving objects from an RGB video sequence has been developed. The video sequence is taken with a camera which is attached to a mobile robot. Since both the camera and the robot may be moving, the stationary background is moving relative to the camera. Compensation for the known camera motion is done in order to achieve fast extraction of moving objects. Since the camera and the robot are computer controlled the necessary transformation matrixes of the corresponding camera motion can be readily calculated.

Keywords: Moving Objects, Image motion compensation, Image analysis, Mobile Robots, Robot vision

1. INTRODUCTION

1.1 *Motivation*

Mobile robots are supposed to work in an environment which is also occupied by people and other mobile robots. Therefore it is important that the robot is able to derive which objects in its vicinity are itself moving and which belong to the stationary background.

Video sequences supply a large amount of information about the environment (Nagel, 1986). Moving objects can be determined using visual information alone. However, on an autonomous mobile robot there also exists an algorithm which controls the robot. This information can be used to speed up the extraction of moving objects.

1.2 *Background*

Several different approaches have been developed to extract moving objects from a video sequence. Most of them calculate the optical flow (Horn and Schunck, 1981) for the images and extract the moving objects along the discontinuity boundaries of the optical flow. Smith and Brady (1994, 1995) have applied this approach successfully. Nelson (1991) has developed two ways to extract moving objects with an arbitrary moving camera. He first extracts regions showing motion which is inconsistent with the camera motion. However, information about camera movements is derived directly from the images. The second approach determines regions which show a highly accelerated movement. Denzler and Niemann (1996) have recently introduced active rays which allow fast real time extraction of moving contours. This approach is especially suited for homogeneous contours. Denzler and Niemann proposed to extend their approach to region based features. For translatory movement of the camera the images can be transformed into polar space (Jain, 1984) where moving objects can easily be separated from the stationary background. This approach has also been followed by Frazier and Nevatia (1990). Mur-

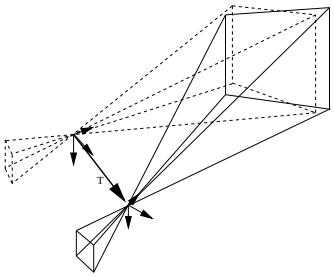

Fig. 1. Camera movement.

ray and Basu (1994) used the information about rotatory camera movement to compensate for the camera movement and to extract moving edges using a pan/tilt camera.

In this paper the approach taken by Murray and Basu is followed. However, on a mobile robot one has to compensate for arbitrary camera movement. For the calculations in this paper the notation of Craig (1989) is used. Let ${}^{C(t_1)}\mathbf{P} = [X(t_1), Y(t_1), Z(t_1)]$ be an object point in the camera frame $\{C(t_1)\}$ at time t_1. The same point has coordinates ${}^{C(t_2)}\mathbf{P} = [X(t_2), Y(t_2), Z(t_2)]$ in the camera frame $\{C(t_2)\}$ at time t_2. If the camera motion (figure 1) from $\{C(t_1)\}$ to $\{C(t_2)\}$ is ${}^{C(t_2)}_{C(t_1)}\mathbf{T}$ one has

$$ {}^{C(t_2)}\mathbf{P} = {}^{C(t_2)}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} \tag{1} $$

$$ \begin{bmatrix} X(t_2) \\ Y(t_2) \\ Z(t_2) \\ 1 \end{bmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} X(t_1) \\ Y(t_1) \\ Z(t_1) \\ 1 \end{bmatrix} \tag{2} $$

Using perspective projection with focal length f the new coordinates at time t_2 can be calculated (Murray and Basu, 1994). Large letters are used to denote three dimensional coordinates and small letters to denote image plane coordinates.

$$ x(t_2) = f\frac{X(t_2)}{Z(t_2)} \tag{3} $$

$$ = f\frac{r_{11}X(t_1) + r_{12}Y(t_1) + r_{13}Z(t_1) + t_x}{r_{31}X(t_1) + r_{32}Y(t_1) + r_{33}Z(t_1) + t_z} \tag{4} $$

$$ = f\frac{r_{11}x(t_1) + r_{12}y(t_1) + fr_{13} + f\frac{t_x}{Z(t_1)}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33} + f\frac{t_z}{Z(t_1)}} \tag{5} $$

$$ y(t_2) = f\frac{Y(t_2)}{Z(t_2)} \tag{6} $$

$$ = f\frac{r_{21}X(t_1) + r_{22}Y(t_1) + r_{23}Z(t_1) + t_y}{r_{31}X(t_1) + r_{32}Y(t_1) + r_{33}Z(t_1) + t_z} \tag{7} $$

$$ = f\frac{r_{21}x(t_1) + r_{22}y(t_1) + fr_{23} + f\frac{t_y}{Z(t_1)}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33} + f\frac{t_z}{Z(t_1)}} \tag{8} $$

Since Murray and Basu only need to compensate for rotatory camera motions they assume that the distance of the objects in view is large. This gives the following expression for the new coordinates.

$$ \lim_{Z(t_1)\to\infty} x(t_2) = f\frac{r_{11}x(t_1) + r_{12}y(t_1) + fr_{13}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33}} \tag{9} $$

$$ \lim_{Z(t_1)\to\infty} y(t_2) = f\frac{r_{21}x(t_1) + r_{22}y(t_1) + fr_{23}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33}} \tag{10} $$

For rotatory motions one can take the limit $Z \to \infty$, however, for translatory camera motions the distance of every point projected onto the camera image has to be known. Thus one would have to solve the structure from motion problem first.

In this paper equation 5 and equation 8 are used to compensate for arbitrary camera motions to predict the new image. It is assumed that the depth differences of the scene in view of the camera are small compared to the distance from the camera to the objects in view. Thus all distances $Z(t_1)$ are approximated by using an average value $\bar{Z}$ as the distance of the points. Equation 11 and equation 12 calculate the motion field (Horn, 1986) of a plane located at a distance $\bar{Z}$ from the camera. The average distance of the points to the camera is continuously calculated and updated as described in the algorithm below.

2. SYSTEM ARCHITECTURE

The algorithm described here has been developed by the author in (Ebner, 1996). It consists of two main parts. The first part detects changes which have occurred from one image to the next by compensating for the known ego-motion. The second part extracts the moving objects using the detected regions as input. Moving objects are extracted by creating a hypothesis about the moving objects that might be in the sequence. If a hypothesis can be validated in the following images, the object will be turned into a moving object. An overview of the individual steps of the algorithm can be seen in figure 2. In the following two sections each of the steps is described in detail.

3. CHANGE DETECTION

The algorithm for change detection previously only operated on gray scale video sequences and has now been extended to full color video sequences. The first part of the algorithm now operates on all three bands (red, green and blue).

It has also been tried to convert the RGB images into HSI (hue, saturation and intensity) space and extract the moving objects there. However the hue is subject to changes whenever the saturation is low. Also there are only few sufficiently saturated pixels in the environment where the experiments were performed.

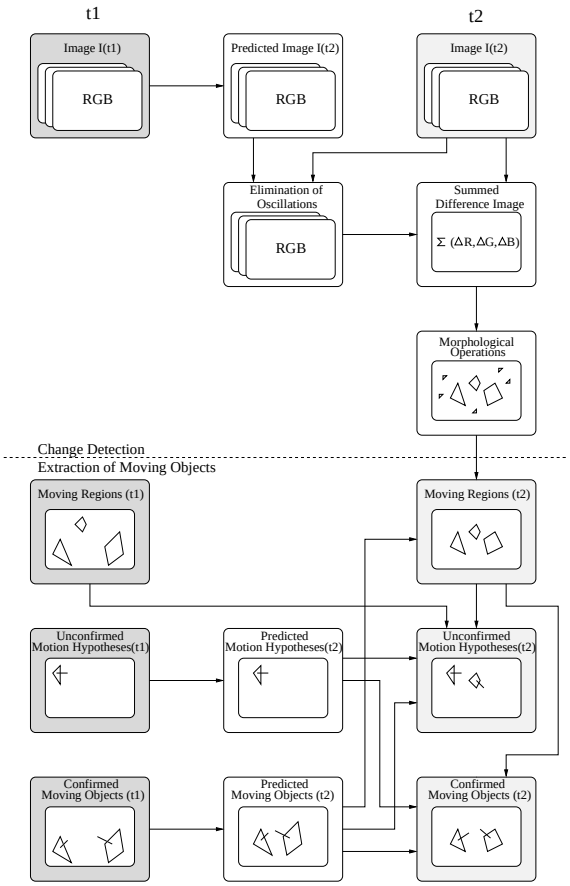

Fig. 2. Algorithm to extract moving objects. Input to the algorithm is the image $I(t_1)$, moving regions $\mathrm{MR}(t_1)$, motion hypotheses $\mathrm{MH}(t_1)$ and the moving objects $\mathrm{MO}(t_1)$ that have previously been extracted. The algorithm then calculates these sets for time t_2. See text for a full explanation of the individual steps.

3.1 *Compensation for known camera movement*

The algorithm operates on two images $I(t_1)$ and $I(t_2)$ taken at times t_1 and t_2 respectively. The algorithm first compensates for the known camera motion ${}^{C(t_2)}_{C(t_1)}\mathbf{T}$. The predicted image $\tilde{I}(t_2)$ at time t_2 is calculated from the image $I(t_1)$ at time t_1. Every point of the image is transformed as described above.

$$x(t_2) = f\frac{r_{11}x(t_1) + r_{12}y(t_1) + fr_{13} + f\frac{t_x}{Z}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33} + f\frac{t_z}{Z}} \quad (11)$$

$$y(t_2) = f\frac{r_{21}x(t_1) + r_{22}y(t_1) + fr_{23} + f\frac{t_y}{Z}}{r_{31}x(t_1) + r_{32}y(t_1) + fr_{33} + f\frac{t_z}{Z}} \quad (12)$$

To estimate the average distance $\bar{Z}$ from the camera to the objects in view, equation 11 and equation 12 are used to derive the distances of the interesting points which are extracted directly from the images. The correspondence between interesting points is established as described in the next subsection. The distance of the interesting points can only be calculated reliably if the camera has made a large translatory movement from one image to the next. If no moving objects have been extracted yet incorrect distances might accidentally be calculated due to the movement of points which belong to a moving object. These distances are eliminated using an median filter which assumes that more interesting points are located on the background rather than on the objects.

3.2 *Elimination of oscillations*

Since the camera is attached to a mobile robot,the camera is subject to oscillations. For the experiments shown here a pan/tilt-camera has been used. Thus one has to eliminate horizontal and vertical oscillations of the camera. The oscillations cannot be calculated from the information supplied by the controllers. Therefore the necessary shift in the horizontal and vertical direction are derived directly from the images.

To calculate the required horizontal and vertical shift the set of interesting points from image $I(t_2)$ is extracted. The set of interesting points from image $I(t_1)$ has already been extracted in the previous iteration of the algorithm. The coordinates can be transformed to their corresponding points in the predicted image $\tilde{I}(t_2)$. Point correspondences can then be found using a standard correlation technique (Jain *et al.*, 1995).

Let $\tilde{F}(t_2)$ be the set of interesting points of image $\tilde{I}(t_2)$ and $F(t_2)$ be the set of interesting points of image $I(t_2)$. The set of interesting points can be determined with any standard feature detection operator. Here the Moravec interest operator (Moravec, 1977) which extracts points with high variance of the surrounding pixel values has been used. The following measure is calculated for every point $(x_1, y_1) \in \tilde{F}(t_2)$ and $(x_2, y_2) \in F(t_2)$ which describes the goodness of the match.

$$c = w \sum_{b\in\{R,G,B\}} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{h}{2}}^{\frac{h}{2}} \left(\tilde{I}_b(x_1+i, y_1+j) - I_b(x_2+i, y_2+j)\right)^2 \quad (13)$$

where $w = \left(1 + \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}\right)$. The factor w is one if both points have the same coordinates. It increases with the distance of the two points. Thus preference is given to a close match rather than a match of two distant points. Two equally good matches could otherwise occur if several identical objects occur in the scene. Since it is assumed that the camera motion has been compensated the close match is the correct one. To eliminate outliers the median of all shifts is taken as the shift for the whole image.

3.3 *Determination of changes*

To detect any changes that have occurred from time t_1 to time t_2 the difference image between the predicted and the actual image is calculated. The differences are calculated for each band individually. Then absolute values of the differences are summed up. This leaves a single band image describing the changes that have occurred.

3.4 *Elimination of small inaccuracies*

The predicted image can only be as good as the data used for the calculations. The accuracy of the predicted image depends on the matrix describing the camera motion. The predicted image may be incorrect due to inaccurate or outdated information from the controllers or it may be incorrect due to a large difference between the actual distance of the points and the assumed distance.

The summed difference image is processed using morphological operations to eliminate any small inaccuracies that may have occurred in the process of image prediction. This approach has been introduced by Murray and Basu (1994). First a gray level closing operation with a 5×5 mask is used, followed by a gray level opening operation with a 9×9 mask. Finally the image is binarized using a thresholding operation.

4. EXTRACTION OF MOVING OBJECTS

For the following text several sets are defined. Each set may contain several regions or objects. Regions which have been detected as changes in the difference image are called "Moving Regions". An object is called a "Motion Hypothesis" as long as it has not been validated yet.

$\mathrm{MR}(t) =$ Moving Regions at time t
$\mathrm{MH}(t) =$ Motion Hypotheses at time t
$\mathrm{MO}(t) =$ Confirmed Moving Objects at time t

The set of moving objects $\mathrm{MO}(t)$ is the final output of the algorithm. The objects in $\mathrm{MH}(t)$ and $\mathrm{MO}(t)$ also posses a motion vector describing the motion of the object in the image plane.

It is assumed that the sets $\mathrm{MR}(t_1)$, $\mathrm{MH}(t_1)$ and $\mathrm{MO}(t_1)$ are available from the previous iteration of the algorithm. For the first iteration of the algorithm each is initialized with the empty set. Since it is assumed that the camera motion is known, one can also calculate the sets $\widetilde{\mathrm{MR}}(t_2)$, $\widetilde{\mathrm{MH}}(t_2)$ and $\widetilde{\mathrm{MO}}(t_2)$, i.e. the predicted moving regions, hypotheses and objects at time t_2. The objects in $\widetilde{\mathrm{MH}}(t_2)$ and $\widetilde{\mathrm{MO}}(t_2)$ also have been moved according to their motion vector. The sets $\mathrm{MR}(t_2)$, $\mathrm{MH}(t_2)$ and $\mathrm{MO}(t_2)$ are initialized with the empty set at the beginning of each iteration.

4.1 *Determination of Moving Regions*

Due to the use of morphological operations to process the difference image one gets several disconnected regions that might actually belong to the same object. Therefore those regions that belong to the same object need to be combined. Two heuristics are used to combine the regions. The first is the distance between two regions. Two regions are joined into one if the distance between the two closest points of the regions is below some threshold. Let $\mathrm{DR}(t_2)$ be the set of resulting regions.

Assuming that some moving objects have been extracted previously these objects are used as a second heuristic to combine the regions. Using the set of moving objects $\mathrm{MO}(t_1)$ and the regions extracted from the difference image $\mathrm{DR}(t_2)$ the set of moving regions $\mathrm{MR}(t_2)$ is calculated as follows. For every region $R \in \mathrm{DR}(t_2)$ take the convex hull of all points of the objects $O \in \mathrm{MO}(t_2)$ that overlap or are completely contained in R. The resulting convex hull will be placed in $\mathrm{MR}(t_2)$ if it is not empty, otherwise R will be placed in $\mathrm{MR}(t_2)$.

4.2 *Determination of Motion Hypotheses*

It is assumed that every motion hypothesis has a fixed lifetime of several iterations of the algorithm. During its lifetime a motion hypothesis needs to be validated in order to become a moving object. For every object $H \in \widetilde{\mathrm{MH}}(t_2)$ that has not yet exceeded its lifetime the convex hull of all points of the regions $R \in \mathrm{DR}(t_2)$ that overlap or are completely contained in H is taken. The resulting convex hull will be placed in $\mathrm{MH}(t_2)$. If the hypothesis can be validated during its lifetime it is placed in $\mathrm{MO}(t_2)$ instead.

Taking the moving regions $\widetilde{\mathrm{MR}}(t_2)$ and $\mathrm{MR}(t_2)$ new motion hypotheses are calculated. For every $R_1 \in \widetilde{\mathrm{MR}}(t_2)$ and $R_2 \in \mathrm{MR}(t_2)$ a new motion hypothesis H is added to $\mathrm{MH}(t_2)$, which has the same position and size as R_2, if the two Regions R_1 and R_2 are likely to have been caused by the same object. For our algorithm size has been used as a compatibility measure. Note that this possibly adds many incorrect motion hypotheses to $\mathrm{MH}(t_2)$. The motion hypotheses that have been added because of an incorrectly assumed correspondence will be removed later on because the hypothesis will most likely not be validated. The motion vector of H is calculated using the center of mass of both regions and the time difference $t_2 - t_1$. Thus it is assumed that the

Table 1. Denavit-Hartenberg parameters of the pan/tilt camera.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$0°$	L_1	0	$-90° + \theta_1$
2	θ_2	L_2	L_3	0
3	$-90°$	0	L_4	0

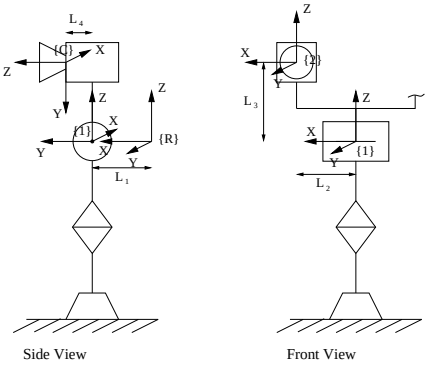

Fig. 3. Camera model with attached frames.

motion of the objects is a smooth one in the image plane and no sudden changes of direction occur. A moving object $O \in \widetilde{\mathrm{MO}}(t_2)$ may become a motion hypothesis if no region $R \in \mathrm{MR}(t_2)$ touches O.

4.3 *Determination of Moving Objects*

For every object $O \in \widetilde{\mathrm{MO}}(t_2)$ the convex hull of all points of the regions $R \in \mathrm{DR}(t_2)$ that overlap or are completely contained in O is taken. The resulting convex hull will be placed in $\mathrm{MO}(t_2)$. If the convex hull is empty the object O is placed back in $\mathrm{MH}(t_2)$ and its lifetime is reset to give it another chance.

5. CALCULATION OF CAMERA MOTION

Let a point with world coordinates ${}^{W}\mathbf{P}$ have coordinates ${}^{C(t_1)}\mathbf{P}$ at time t_1 in the coordinate system of the camera frame $\{C(t_1)\}$. The same point has coordinates ${}^{C(t_2)}\mathbf{P}$ at time t_2. Thus

$$ {}^{W}\mathbf{P} = {}^{W}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} = {}^{W}_{C(t_2)}\mathbf{T} \cdot {}^{C(t_2)}\mathbf{P} \qquad (14) $$

where ${}^{W}_{C(t_1)}\mathbf{T}$ and ${}^{W}_{C(t_2)}\mathbf{T}$ describe the transformation from the corresponding camera frames to world coordinates.

The position of a point at time t_2 can be calculated given the position of the point at time t_1 and the known camera transformations:

$$ {}^{C(t_2)}\mathbf{P} = {}^{W}_{C(t_2)}\mathbf{T}^{-1} \cdot {}^{W}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} \qquad (15) $$

The transformation from the world frame to the camera frame is described as a transformation from the world frame $\{W\}$ to the robot base $\{R(t)\}$ and a transformation from the base frame to the camera frame $\{C(t)\}$. Thus one has

$$ {}^{W}_{C(t_1)}\mathbf{T} = {}^{W}_{R(t_1)}\mathbf{T} \cdot {}^{R(t_1)}_{C(t_1)}\mathbf{T} \qquad (16) $$

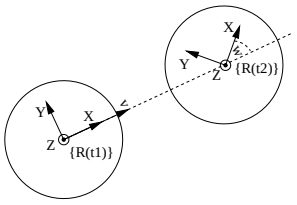

Fig. 4. Movement of the robot base.

and

$$ {}^{W}_{C(t_2)}\mathbf{T} = {}^{W}_{R(t_2)}\mathbf{T} \cdot {}^{R(t_2)}_{C(t_2)}\mathbf{T} \qquad (17) $$

This leads to the following expression for the point ${}^{C(t_2)}\mathbf{P}$ in the camera frame $\{C(t_2)\}$

$$ {}^{C(t_2)}\mathbf{P} = {}^{C(t_2)}_{R(t_2)}\mathbf{T} \cdot {}^{R(t_2)}_{W}\mathbf{T} \cdot {}^{W}_{R(t_1)}\mathbf{T} \cdot {}^{R(t_1)}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} \qquad (18) $$

$$ = {}^{C(t_2)}_{R(t_2)}\mathbf{T} \cdot {}^{R(t_2)}_{R(t_1)}\mathbf{T} \cdot {}^{R(t_1)}_{C(t_1)}\mathbf{T} \cdot {}^{C(t_1)}\mathbf{P} \qquad (19) $$

The individual transformation matrixes can be calculated in a straightforward way for a standard pan/tilt-camera (figure 3) using the Denavit-Hartenberg parameters of table 1. For the camera used here the distances L are as follows: $L_1 = 0.19m$, $L_2 = \pm 0.10m$, $L_3 = 0.065m$ and $L_4 = 0.025m$. The camera system is actually a stereo camera system but only the right or the left camera is used. The mobile robot is able to perform rotatory and translatory movements of its base. Let w be the rotatory and v the translatory velocity of the robot base at time t_1 then the matrix which approximately describes the robot movement (for small δt) is given by:

$$ {}^{R(t_2)}_{R(t_1)}\mathbf{T} = \mathbf{R}_Z(-w(t_2 - t_1)) \cdot \mathbf{D}_X(-v(t_2 - t_1)) \qquad (20) $$

where $\mathbf{D}_X$ is a matrix describing the translatory motion of the robot along the X axis and $\mathbf{R}_Z$ is a matrix describing the rotatory motion of the robot about the Z axis. The frame defined for the robot base can be found in figure 4.

To get a very accurate image prediction one has to use the most up to date information as possible. However a distributed client/server architecture is used for the control of the camera and the robot. Due to the delays involved, it is very difficult if not impossible to get exact data about the current link angles needed for the calculations.

The image time stamps are the most accurate information available since they are set directly by the kernel whenever an image is grabbed. After an image is grabbed it is supplied with the current link angles and link velocities of the camera and the speed of the robot. The current link angles are taken from time t_1 whereas the new link angles are calculated using the link position and velocities from time t_1 and the time difference $(t_2 - t_1)$. It suffices to know only the approximate position of the camera. However it is very important to have the relative motion that has occurred during the time the images have been grabbed.

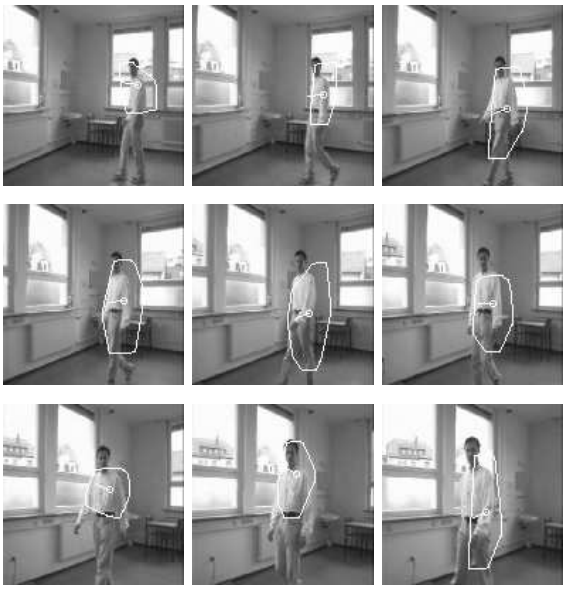

Fig. 5. Sequence and extracted moving object.

6. EXPERIMENTS

A video sequence taken with a RWI B21 mobile robot driving with a linear velocity of $4.6\frac{\text{cm}}{s}$ and an angular velocity of $8\frac{^\circ}{s}$ has been recorded. In other experiments linear velocities of up to $20\frac{\text{cm}}{s}$ and angular velocities of up to $35\frac{^\circ}{s}$ were used. The moving object has been a human walking around in a room. The extracted moving object can be found in figure 5. The object shown here has been extracted off-line from an image sequence. One iteration took about 1.43s for an image of size 128×128 on a 133MHz Pentium for one iteration processing all three bands (red, green & blue).

The algorithm currently takes about 0.269s on a 133MHz Pentium for one iteration on a 64×64 image sequence if all three bands are used. If the moving objects are extracted using a single band gray scale image an iteration of the algorithm takes 0.137s on 64×64 images. This translates to 3.7fps for color and 7.3fps for gray scale images. The first part of the algorithm (change detection) takes about 97% of the time of the whole algorithm. Thus special hardware to process the images and detect the changes would speed up the algorithm considerably.

7. CONCLUSION

In this paper an algorithm which extracts moving objects from a color video sequence has been presented. The algorithm first compensates the known ego-motion of the mobile robot by calculating the camera transformation and predicting the resulting image movement. It has been shown how the information supplied by the control algorithm of a mobile robot can aid in the fast extraction of moving objects.

8. ACKNOWLEDGEMENTS

The author is currently supported by a scholarship according to the Landesgraduiertenförderungsgesetz.

9. REFERENCES

Craig, J.J. (1989). *Introduction to Robotics: Mechanics and Control (2^{nd} ed)*. Addison-Wesley Publishing Company. Reading, MA.

Denzler, J. and H. Niemann (1996). Echtzeitobjektverfolgung mit aktiven Strahlen. In: *Mustererkennung 1996, 18. DAGM-Symposium* (B. Jähne, P. Geißler, H. Haußecker and F. Hering, Eds.). Heidelberg. pp. 84–91.

Ebner, M. (1996). Robuste Extraktion sich bewegender Objekte aus einer mit dynamischer Kamera aufgenommenen Videosequenz. Diplomarbeit. IPVR, Universität Stuttgart.

Frazier, J. and R. Nevatia (1990). Detecting moving objects from a moving platform. In: *Proceedings of the DARPA Image Understanding Workshop, Pittsburgh, PA*. pp. 348–355.

Horn, B.K.P. (1986). *Robot Vision*. The MIT Press. Cambridge, Massachusetts.

Horn, B.K.P. and B.G. Schunck (1981). Determining optical flow. *Artificial Intelligence* **17**, 185–203.

Jain, R.C. (1984). Segmentation of frame sequences obtained by a moving observer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**(5), 624–629.

Jain, R., R. Kasturi and B.G. Schunck (1995). *Machine Vision*. McGraw-Hill, Inc.. New York.

Moravec, H.P. (1977). Towards automatic visual obstacle avoidance. In: *Proceedings of the 5th International Joint Conference on Artificial Intelligence*. Vision–1: p. 584.

Murray, D. and A. Basu (1994). Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(5), 449–459.

Nagel, H.-H. (1986). Image sequences - ten (octal) years - from phenomenology towards a theoretical foundation. In: *Proceedings of the Eighth International Conference on Pattern Recognition, Paris, France, Oct., 1986*. IEEE Computer Society Press. pp. 1174–1185.

Nelson, R.C. (1991). Qualitative detection of motion by a moving observer. *International Journal of Computer Vision* **7**(1), 33–46.

Smith, S.M. and J.M. Brady (1994). A scene segmenter; visual tracking of moving vehicles. *Engineering Applications of Artificial Intelligence* **7**(2), 191–204.

Smith, S.M. and J.M. Brady (1995). Asset-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(8), 814–820.