An Airborne Bayesian Color Tracking System

Felix Woelk, Ingo Schiller and Reinhard Koch

***Abstract*— Mobile tracking systems are of wide interest. For instance sport photographers often hire helicopters to obtain close range images of e.g. aquatic athletes. Besides the high price for hiring a helicopter, good helicopter pilots are not everywhere available. One way to circumvent these problems is the use of a cheap model helicopter. Since flying a model helicopter is a demanding task, there are either two persons needed for the operation of the system (one person for steering the helicopter and the other for the operation of the camera), or alternatively an automatic tracking system can be used. In this work an automatic, image based tracking system is described. The algorithm combines color histograms as similarity measures with a particle filter and achieves fast and robust tracking results. A novel method for fast histogram computation is proposed. The system keeps the object of interest automatically in the viewing field of the camera so that the only task left to the camera operator is to initiate the imaging process. This simple task can be additionally managed by the helicopter pilot so that a single person is sufficient to steer the helicopter and to take the images.**[1]

I. Introduction

Photographers often hire helicopters to take close range shots of e.g. aquatic athletes. Hiring a helicopter is very expensive in the first place. In the second place good helicopter pilots are often not available at a given time or a given location. To circumvent these two problems the use of a cheap tracking system mounted on a model helicopter is suggested. In this paper a prototype for an airborne system for photographing, which can be operated by a single person is described. The major innovation used in this prototype is the automatic tracking system. It is image based and combines color histogram similarity with a fast and robust Bayesian particle filter approach.

The paper is structured as follows: After a short system overview the color histogram similarity measure and the particle filter and its application to the tracking task are described in section II. The specification of the control system in section III is followed by the a description of experimental results in section IV.

System Overview

The system consists of a model helicopter (fig. 1) with a mounted tracking system (fig. 2). The tracking system itself consists of a pan tilt unit (PTU), a rotation sensor, an IEEE1394 tracking camera which is rigidly attached to a high quality imaging or video camera and an embedded PC.

Felix Woelk, Ingo Schiller and Prof. Dr. Reinhard Koch are with the Institute for Computer Science, Christian-Albrechts-Universität zu Kiel, 24118 Kiel, Germany, {woelk, ischiller, rk}@mip.informatik.uni-kiel.de

[1]Hardware used in this work has been partly funded by the photographer Stevie Bootz from Kiel and partly by the Technologie Stiftung Schleswig Holstein.

Fig. 1. Picture of the flying model helicopter with the mounted tracking system. A detailed photo of the tracking system is given in fig. 2.

A lipstick camera with an analogue recording unit is used as high quality video camera. The PC is used to control the PTU on which the cameras are mounted and to process the images. The rotation sensor, which is rigidly attached to the helicopter, is used to detect and compensate fast rotations of the helicopter. The embedded PC is a Pentium M 1.5 GHz. It is capable of handling the image processing and controlling task comfortably at a framerate of 15 Hz.

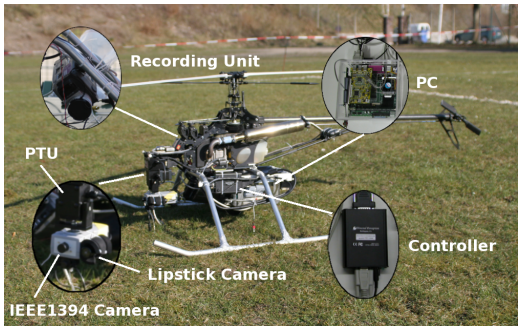

Fig. 2. The tracking system. The prototype of the tracking system does not yet include the high quality imaging camera. See text for detailed description.

II. Object Location

A. Similarity Measure

To keep the object of interest in the viewing space of the camera, the best fit to some user selected template must be found in every image. Athletes usually are non rigid objects and their images are subject to size changes and rotations (f.e. when the athlete performs a loop). A color histogram is used as a rotation invariant template that also can handle the non-rigidity of the athlete. First the color space in which the histogram is computed is described, then a fast, novel algorithm for histogram computation by use

of integral histograms is proposed and finally the similarity measure on histograms used in the system is specified.

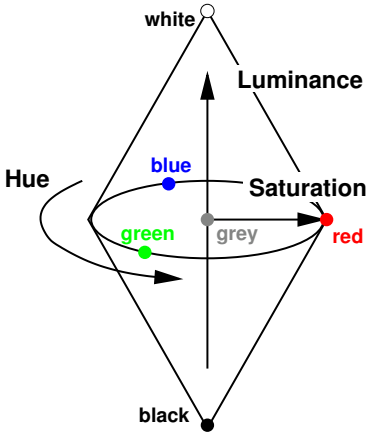

Fig. 3. The Hue Saturation Luminance (HSL) color space. Color information is decoupled from lighting conditions in it. See text for further details.

1) HSL Colorspace: In the Hue Saturation Luminance (HSL) Colorspace (fig. 3) the chromatic information is decoupled from the lighting conditions. The first two parameter describe the base color (hue) and the saturation independent of the lighting conditions. The last parameter (luminance) contains the information about the lighting conditions. The straight forward approach to histogram computation in the HSL color space would result in a 3 dimensional (3D) histogram. Since the main interest is in color information and this information is solely contained in the first two parameters, a 2D histogram built from hue and saturation values only is sufficient. When the saturation falls below some minimal value, no color information is contained in this pixel. The luminance (corresponding to a grey value in this case) is then used to build up an additional grey value histogram. The resulting overall histogram for a given image region contains a 2D color histogram, built from all pixels with saturation above a certain threshold, and a 1D luminance histogram, built from all pixels with saturation below that threshold. For convenience this overall histogram can be represented as a single vector by concatenating all the rows in the 2D histogram matrix and finally appending the 1D luminance histogram.

2) Integral Histogram: The integral histogram reduces a histogram computation (for arbitrary region size) to four vector additions. The basic idea behind the integral histogram images is frequently used in other contexts. In computer graphics it is used under the name of summed area tables and in [14] it is used for fast feature calculation in the context of face detection.

In the integral histogram image a histogram is stored at every pixel position. Every histogram in the integral histogram image is composed from all pixels of the input image which are located above and to the left of the position. In figure 4 the histogram H_α of region A is stored in position α and the histogram H_δ of the joint regions A, B, C and D is stored in position δ. Computing the histogram H_D of region D can now be reduced to four vector additions:

$$H_D = H_\delta + H_\alpha - H_\beta - H_\gamma \tag{1}$$

The computation of the integral histogram image can be improved by using equation 1. The size of region D is set to one pixel. Assume the computation of the integral histogram started row wise at the top left corner of the image and has been conducted up to the point γ. The next unknown entry in the histogram image is hence the histogram H_δ. It can be computed by

$$H_\delta = H_\gamma + H_\beta - H_\alpha + H_D \tag{2}$$

where the histograms H_δ, H_α and H_β are known and the histogram of region D, H_D can be easily computed. It consists of a single entry resulting from the only pixel in region D. The procedures needed to compute the histograms in the first row and the first column can be easily derived by the reader.

3) Bhattacharyya Distance: The histogram similarities, which will be used later in the particle filter algorithm, can be computed in different ways. The following measures have been used in previous work. The Minkowski-form distance is based on the L_p norm, the Bhattacharyya distance measures the statistical separability of histograms, the χ^2 statistics measures the likelihood of one histogram being part of another histogram, the Kullback-Leibler (KL) divergence measures the similarity of two distributions, the Jeffrey-divergence is the symmetric version of KL distance, the histogram quadratic distance is designed to measure weighted histogram similarities and the Earth Mover's distance measures histogram similarity in terms of transfer costs [11].

It has been shown that the Bhattacharyya distance performs favorably compared to the Minkowski-form distances based on L_1 and L_2 norm [3]. In this algorithm,

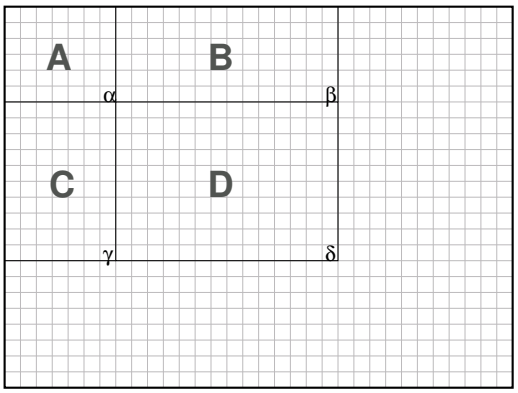

Fig. 4. Histogram computation by using the integral histogram image. In the integral histogram image position α contains the histogram H_α of region A. The histogram H_D of region D can now be computed very efficient: $H_D = H_\delta + H_\alpha - H_\beta - H_\gamma$.

the Bhattacharyya distance D_B is used as a similarity measure. It is calculated between the template histogram $\hat{H} = (\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_n)^T$ (consisting of the entries $\hat{H}_i$ in bin $i \in \{1, \ldots, n\}$) and any other histogram $H(u,v) = (H_1, H_2, \ldots, H_n)^T$ in position $(u,v)^T$ in the image by

$$D_B(\hat{H}, H(u,v)) = \sqrt{1 - \sum_{i=1}^{n} \sqrt{\hat{H}_i \cdot H_i}} \tag{3}$$

This approach requires normalization of the histograms. The normalized histogram $H' = (H'_1, H'_2, \ldots, H'_n)^T$ can be calculated from the original histogram $H = (H_1, H_2, \ldots, H_n)^T$ by

$$H'_i = \frac{H_i}{\sum_{k=1}^{n} H_k} \tag{4}$$

Consider the system state $x = (u, v, s)^T$, i.e. the position $(u,v)^T$ and size s of the best fitting region to the template, as the outcome of a stochastic process. Using the Bhattacharyya distance it is now possible to define a similarity function $p(x) : \mathbb{R}^3 \to \mathbb{R}$

$$p_{\text{Bhat}}(x) = \frac{D_B(\hat{H}, H(x))}{\int D_B(\hat{H}, H(\bar{x}))\ d\bar{x}} \tag{5}$$

In practice, the input space of $p_{\text{Bhat}}(x)$ is not $\mathbb{R}^3$ but a limited, compact interval of it, e.g. the size of the histogram is upper bounded by the image size and lower bounded by the image sensor resolution. Evaluating the similarity over the complete 3D input space would yield a maximum at the position and size of the best fitting region. Due to restricted computational resources, a complete search of the input space is not feasible. Further on, the knowledge of the similarity function over the whole 3D space is unnecessary. It is sufficient to estimate certain characteristics of it, e.g. in practice the modes or the conditional mean of the distribution suffice.

Further stabilization can be achieved by choosing a Bayesian approach incorporating the result from the last time step as a prior into the estimation. A fast Bayesian approach estimating only the necessary characteristics of $p_{\text{Bhat}}(x)$ is the particle filter[2].

B. Particle Filter

The idea of the particle filter has been first published in 1949 by Metropolis and Ulam. Since than and it has been only sporadically mentioned in the literature up to the rediscovery in 1993 by Gordon et. al [2]. A large variety of papers on particle filters has been written since than. A good introduction can be found in [1]. In 1998 Isard introduced this technique into the field of computer vision for tracking tasks [6], [7]. Lately a lot of effort went into improvements of particle filters to overcome certain limitations and problems [4], [5], [8], [13]. The idea of combining color histograms with particle filters for tracking applications has been subject to recent research [10], [9].

[2]If the involved densities were Gaussian and the transformations linear, a better approach to this problem would be the Kalman filter [15]. In our case however the distributions can be of arbitrary form.

Let x_t denote the unobserved state of the system [3] at the discrete time $t \in \mathbb{N}$. Modeling the system as a Markov process results in the conditional transition probability $p(x_{t+1}|x_0, x_1, \ldots, x_t) = p(x_{t+1}|x_t)$. In other words the state of the system in the next time step does only depend on the current state of the system and not on the history. This can be achieved by using auto-regressive second order dynamics [6], which is equivalent to include the velocity in the system state [4]. Let further denote y_t the observation of our system at time t. If we assume a likelihood function $p(y_t|x_t)$, modeling the observation process[5], the posteriori distribution is given by Bayes' law

$$p(x_t|y_t) = \frac{p(y_t|x_t)p(x_t)}{\int p(y_t|x_t)p(x_t)\ dx} \tag{6}$$

where the marginalization $p(y_t) = \int p(y_t|x_t)p(x_t)\ dx$ can be seen as a normalization factor.

Since $p(x_t)$ is at no time known exactly (it is unobserved), the best estimate for $p(x_t)$ can be calculated by using the estimate from the last time step $p(x_{t-1}|y_{t-1})$ and the transition probability $p(x_t|x_{t-1})$ [1] resulting in the prediction equation

$$p(x_t|y_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{t-1})\ dx \tag{7}$$

Plugging this estimate into Bayes' law (eq. 6) results in the update equation

$$p(x_t|y_t) = \frac{p(y_t|x_t)p(x_t|y_{t-1})}{\int p(y_t|x_t)p(x_t|y_{t-1})\ dx} \tag{8}$$

Generally the prediction and update equations cannot be computed in closed form since they require the evaluation of complex and possibly multidimensional integrals.

- Initialize (t=0):
  Generate N independent identical distributed (iid) samples $x_0^{(i)}, i \in \{1, \ldots, N\}$ from the user given initial distribution $p(x_0)$.
- Iterate:
  1) Predict $x_{t+1}^{(i)}$ by sampling from $p(x_{t+1}|x_t^{(i)})$
  2) Evaluate weights[5] $w_{t+1}^{(i)} = p(y_{t+1}|x_{t+1}^{(i)})$
  3) Normalize the weights.
  4) Resample N times with replacement from the samples $x_{t+1}^{(i)}$ according to the weight $w_{t+1}^{(i)}$.
  5) Set $t = t+1$ and repeat iteration (goto 1).

Fig. 5. Description of the particle filter algorithm [1].

Applying Monte Carlo integration techniques leads to the particle filter algorithm (given in fig. 5). In the particle filter

[3]In our case the system state consists of the position and the size of the region which has the smallest Bhattacharyya distance to the template.

[4]Expanding the system state about its velocity can be achieved by additionally storing the system state from the last last time step.

[5]The observation process in this application is modeled as observing the true value plus zero mean normal distributed noise, i.e. $p(y_t|x_t) \approx \exp(-\frac{D_B^2}{2\sigma^2})$

algorithm, the distributions are approximated by a set of N particles $x^{(i)}$ and their weights $w^{(i)}$. To reduce the well

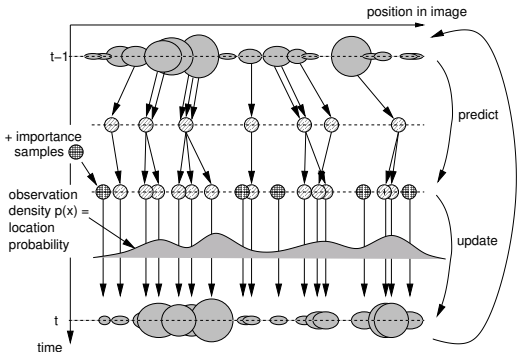

Fig. 6. Graphical representation of a particle filter cycle with importance sampling. The state space dimension is given on the horizontal axis. The vertical axis represents the time. The new samples are selected from the samples of the last timestep (top) according to their weight (size of blobs). After applying the motion model, diffusion is added. These two steps represent the prediction step. The importance samples are added and weights are calculated by evaluating the observation density at the sample positions (update). The weights of the importance samples are corrected according to eq. 9 in this step. The resulting samples are used as input to the next time step. (Figure similar to [1])

known degeneracy problem and to allow reinitialization if the object is lost, step 4 in the iteration process is modified so that a fraction of the samples are chosen by sampling from an importance distribution $g(x)$ [1]. The weights of these importance samples $w^{(i)}_{\text{imp},t}$ must then be corrected to achieve a consistent representation of the posteriori density [7]

$$w^{(i)}_{\text{imp},t+1} = \frac{p(y_{t+1}|x^{(i)}_{t+1})}{g(x)} \tag{9}$$

Fig. 6 shows a graphical representation of one iteration step in the particle filter with the modified resampling step. For a thorough discussion of particle filters see [1].

III. Automatic Control

In order to keep the object of interest in the field of view of the imaging camera, it is necessary to center the object in the tracking camera, which has a different aperture angle from the imaging camera. The standard Proportional Integral Derivative (PID) controller used for this task is described first. Afterwards the incorporation of the rotation sensor, which is used to compensate fast rotations of the helicopter, is described.

A. PID Controller

The PID controller (fig. 7) is a closed-loop controller. It observers the continuous output $u(t)$ of a system (the position of the object of interest in the image) at discrete times n and tries to obtain an output value close to the setpoint (the image center) by changing the system input $y(n)$ (the speed of the PTU). The difference between the system output $u(n)$ and the setpoint is denoted $e(n)$. The resulting output of the system is the input to the PID

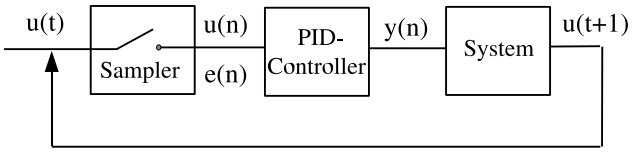

Fig. 7. Systematic view of a PID controller. See text for description.

controller in the next time step (closed-loop). In the time continuous case, the PID controller calculates the system input $y(t)$ via

$$y(t) = K_P e(t) + K_I \int e(t)\ dt + K_D \frac{d}{dt} e(t) \tag{10}$$

where K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain [12]. The discrete form of the PID controller is used. The PID controller has been tuned using the Ziegler-Nichols method [12].

B. Rotation Sensor

For the following considerations a fixed computation time t_C corresponding to the framerate and a fixed aperture angle $2\alpha_A$ of the tracking camera is assumed. The tracking system is guaranteed to fail if the object disappears from the image used by the tracking camera between two frames. It is assumed that the object is initially centered in the image. Two simple cases can be easily analytically analyzed:

1) *No camera rotation and constant relative velocity between camera center and object:*
   An upper bound for the maximum relative velocity $v_{\max}$ is given by
   $$v_{\max} = \tan(\alpha_A)\frac{d}{t_C} \tag{11}$$
   where d is the distance of the object from the camera center. In this estimate it is assumed that the relative velocity is parallel to the image plane (worst case). Using typical values ($\alpha_A = 0.36 \approx 21^\circ$, $t_C = 0.1$s and $d = 10$m), eq. 11 results in $v_{\max} \approx 37\frac{\text{m}}{\text{s}} \approx 136\frac{\text{km}}{\text{h}}$. Typical speeds reached by aquatic athletes (wind driven) are in the order of 20 knots $\approx 11\frac{\text{m}}{\text{s}} \approx 39\frac{\text{km}}{\text{h}}$.
2) *Pure camera rotation plus zero relative velocity:*
   An upper bound for angular velocity $\omega_{\max}$ is given by
   $$\omega_{\max} = \frac{\alpha_A}{t_C} \tag{12}$$
   Again using typical values ($\alpha_A = 0.36 \approx 21^\circ$ and $t_C = 0.1$s), eq. 12 results in $\omega_{\max} \approx 3.6\frac{1}{s} \approx 206\frac{^\circ}{s}$. Angular velocities reached by model helicopters can be significantly higher than $\omega_{\max}$.

Comparing these two upper bounds reveals that system rotations impose a bigger threat to the tracking system than relative velocities. In order to achieve higher acceptable bounds on the angular velocity, one could

- reduce the computational time by using less particles (resulting in poorer results from the particle filter).

However a lower bound on the computation time is given by the computation of the integral image and the maximum framerate of the tracking camera.

- increase the aperture angle of the tracking camera (resulting in smaller images of the object of interest and hence less details in the histogram and therefore also poorer results from the particle filter).
- use an additional rotation sensor measuring the angular velocity of the model helicopter. The angular velocity ω measured by the rotation sensor can be directly used as an additional term into PID controller equation

$$y(t) = K_P e(t) + K_I \int e(t)\ dt + K_D \frac{d}{dt} e(t) + K_\omega \omega \quad (13)$$

where K_ω denotes the angular velocity gain.

The latter approach is used in the tracking system.

IV. Results

After extensive testing on synthetically generated image sequences (not shown) and application of the particle filter on sequences taken during real helicopters rides with manually operated camera (sec. IV-A), the tracking system has been tested in both ground experiments (using the standalone tracking system without the model helicopter) (sec. IV-B) and airborne experiments (tracking unit mounted onto the model helicopter) (sec. IV-C).

A. Experiments without Controlling

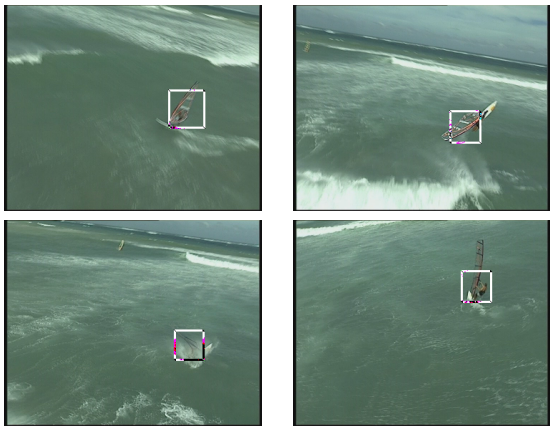

Fig. 8. Some images from sequences taken during a ride on a full size helicopter with manually operated camera. The object is tracked even in the case of extreme rotation and occlusion.

Some images from sequences taken during a ride on a full size helicopter with manually operated camera are shown in fig. 8. In this sequence the camera moves in a relatively smooth way. During the 244 images of the sequence the target has never been lost, even though extreme rotation and occlusion by water occurred.

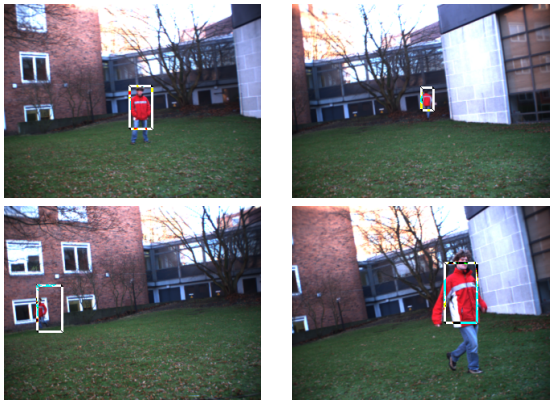

Fig. 9. Some images from a ground sequence. The object is successfully tracked throughout the whole sequence of 484 frames in spite of unsteady camera motion, size changes, occlusion, background clutter and erratic object motion.

B. Ground Experiments

Because experiments with a model helicopter are time consuming, the basic experiment have been conducted while a person was carrying the tracking system. Fig. 9 shows some images of a ground sequence. The target has been successfully tracked over the complete sequence of 484 images through cluttered background even though complete occlusion and rotation occurred. The maximum sample weight is plotted for each frame in fig. 10. The object was occluded from frame 111 to 160. Between frame 160 and 180 it slowly reappeared coming up a staircase. From frame 220 to frame 365 the camera moved very unsteadily. The object was lost due to back light conditions between frame 365 and frame 375. The system recovered very fast after the object loss.

C. Airborne Experiments

The final step in verifying the algorithms is to test the whole system in the real environment. Due to the weight of the equipment (which is near the upper bound of the payload), the experiments have not been made on the open see but on a sporting field. Figure 11 shows some images from the resulting sequences. A person with a red shirt was

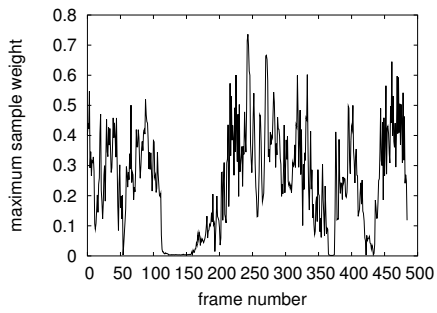

Fig. 10. The maximum sample weight over the 484 frames of the ground sequence shown in fig. 9. See text for a thorough discussion.

Fig. 11. Some images from an airborne sequence. The object is successfully tracked throughout a sequence of approx. 10 minutes length in spite of size changes, illumination changes and erratic object motion. The left column shows the images captured with a high quality analog video camera and the left column shows the images captured with the digital tracking camera. Due to limited bandwidth, only the region around the object could be transmitted via WLAN to a ground station for recording. The size of the bar on the left side of the image indicates the similarity to the template histogram.

selected as target object. The detected object is marked in the picture and the picture is send to a laptop at the ground via WLAN. Due to bandwidth limitation it was not possible to transmit the whole image. The transmitted clipping of the tracking images is shown in the right column of fig. 11. The bar on the left side of the tracking images indicates the similarity to the template histogram. The corresponding image from the analog high quality video camera is shown in the left column. The algorithm works very well even though the object is lost a couple of times (because of very fast object movement too close to the camera) during the 10 minute sequence. If the object re-appeared in the image the algorithm recovered automatically.

D. Timing

Using 800 samples and the integral histogram algorithm in tracking images of size 320×240 with 40 bins in the histogram (4 bins in saturation, 8 bins in hue and 8 bins for the luminance only pixels) results in a framerate of 12 Hz on the embedded PC with a 1.5 GHz Pentium M CPU. The computation time is subject to extreme changes when the integral histogram image is not used for the computation

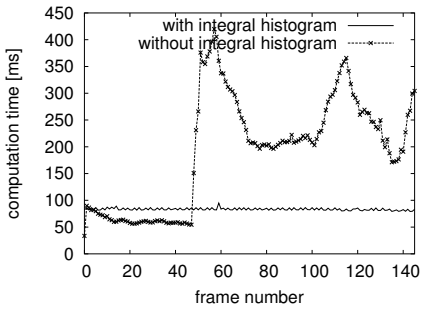

Fig. 12. Comparison of the computation speed of the tracking algorithm with and without the integral histogram algorithm. Due to the novel histogram computation scheme, the computation time remains stable at ≈ 85 ms. When neglecting the fast integral histogram approach (computing every histogram straight forward), the framerate is subject to extreme changes.

of the histograms. The use of the integral histogram allows a much better load balancing as the computation time is nearly constant even for varying histogram size. Therefore, constant frame rate can be achieved, which stabilizes the tracking.

V. Summary

A robust, realtime color histogram based tracking system has been presented. The system has been proven to be operable even in the relatively rough conditions (vibrations, unsteady camera motion, ..) when mounted on a model helicopter.

References

[1] A. Doucet, N.d. Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[2] N.J. Gordon, D.J. Salmond and A.F.M. Smith "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation." *IEE-Proceedings-F*, 140, 107-113, 1993.

[3] B. Huet and E. Hancock, "Structural indexing of infra-red images using statistical histogram comparison", *IWISP*, pp 653-656, 1996.

[4] C. Hue, J.-P. Le Cardre and P. Perez, "Tracking Multiple Objects with Particle Filtering." *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791-812, 2002

[5] Z.Khan, T. Balch and F. Dellaert "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets." *ECCV*, 2004

[6] M. Isard and A. Blake "Condensation – conditional density propagation for visual tracking." *IJCV*, 29(1), pp.5-28, 1998.

[7] M. Isard and A. Blake "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework." *ECCV*, vol. 1 893-908, 1998.

[8] M. Isard and J. McCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker." *ICCV*, 2001.

[9] K. Nummiario, E. Koller-Meier and L.v. Gool, "A Color-based Particle Filter." *GMBV*, 2002.

[10] P. Perez et al., "Color-Based Probabilistic Tracking." *ECCV*, 2002.

[11] A. Shahahrokni and T. Drummond and P. Fua, "Texture Boundary Detection for Tracking Applications." *http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHAHROKNI1/texture.html*

[12] H. Unbehauen, *Regelungstechnik 1/2*, Viehweg, 2000.

[13] J. Vermaak et al., "Maintaining Multi-Modality through Mixture Tracking." *ICCV*, 2003.

[14] P. Viola and M. Jones, "Robust Real-time Object Detection.", *IJCV*, submitted.

[15] G. Welch and G. Bishop, "An Introduction to the Kalman Filter." *TR 95-041*, University of North Carolina, 2001.