Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Weak Models and Cue Integration for Real–time Tracking

D. Kragic and H. I. Christensen
Centre for Autonomous Systems, Numerical Analysis and Computer Science
Royal Institute of Technology, Stockholm, Sweden, {danik,hic}@nada.kth.se

Abstract

Traditionally, fusion of visual information for tracking has been based on explicit models for uncertainty and integration. Most of the approaches use some form of Bayesian statistics where strong models are employed. We argue that for cases where a large number of visual features are available, weak models for integration may be employed. We analyze integration by voting where two methods are proposed and evaluated: i) response and ii) action fusion. The methods differ in the choice of voting space: the former integrates visual information in image space and latter in velocity space. We also evaluate four weighting techniques for integration.

1 Introduction and Motivation

A robust visual tracking with respect to variations in natural environments is one of a key research issues nowadays. We argue that robust tracking may be achieved in an integrated framework by employing a consensus of several visual cues. This idea has been investigated before where a Bayesian framework was used for integration [2], [3]. In [4], *Incremental Focus of Attention* architecture performs tracking in a multi–layered framework. One modality/cue is used at any given moment, processing occurs in a single layer and when the *a–priori* given constraints are met, the layer is changed. Contrary to this serial or, according to [6], *strong coupling* approach, we propose a parallel or a *weak coupling* framework where all cues are used at each time step. Their importance or the effect on the overall result are determined by the assigned weights. *Voting* is here adopted as the underlying integration strategy [5]. Compared to the Bayesian approaches, voting requires no detailed models of the form $p(cue|object)$ which may be difficult or even impossible to determine. A very simple or no model is used to represent this relationship giving it the advantage to operate "model–free" with respect to individual cues. In the simplest case, each estimator may be a classifier that votes for a particular attribute or against it where the level of belief (Dempster-Shafer) or degree of uncertainty (Bayesian) is completely abstracted to give a binary output.

2 Background and Theory

Our tracking algorithm employs the four step *detect–match–update–predict loop*, Fig. 1(a). The objective here is to track a part of an image (a region) between frames. The image position of its center is denoted with $\mathbf{p} = [x\, y]^T$. Hence, the state is $\mathbf{x} = [\, x\, y\, \dot{x}\, \dot{y}\,]^T$ where a piecewise constant white acceleration model is used [10]:

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{v}_k, \quad \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k$$
$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ 0 & \frac{\Delta T^2}{2} \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}, \mathbf{H}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (1)$$

For prediction and estimation, the $\alpha - \beta$ filter is used, [10]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k\hat{\mathbf{x}}_k, \quad \hat{\mathbf{z}}_{k+1|k} = \mathbf{H}\hat{\mathbf{x}}_{k+1|k}$$
$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{W}[\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}], \mathbf{W} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \\ \frac{\beta}{\Delta T} & 0 \\ 0 & \frac{\beta}{\Delta T} \end{bmatrix} \quad (2)$$

2.1 Voting

Voting methods in general, deal with n input data objects, c_i, having associated votes/weights w_i (n input data–vote pairs (c_i, w_i)) and producing the output data–vote pair (y, v) where y may be one of the c_i's or some mixed item. Hence, voting combines information from a number of sources and produces outputs which reflect the consensus of the information. The reliability of the results depends on the information carried by the inputs and, as we will see, their number. A cue is formalized as a mapping from an action space, $\mathbf{A}$, to the interval [0,1], $c : \mathbf{A} \rightarrow [0, 1]$. This mapping assigns a *vote* or a preference to each action $a \in \mathbf{A}$, which, in the context of tracking, may be considered as the position of the target. These votes are used by a *voter* or a *fusion center*, $\delta(\mathbf{A})$. Based on the ideas proposed in [7], [8], we define the following voting scheme:

Definition 2. 1 - ***Weighted Plurality Approval Voting*** *For a group of homogeneous cues, $\mathbf{C} = \{c_1, \ldots, c_n\}$, where n is the number of cues and O_{c_i} is the output of a cue i, a weighted plurality approval scheme is defined as:*

$$\delta(a) = \sum_{i=1}^{n} w_i\, O_{c_i}(a) \quad (3)$$

where the most appropriate action is selected according to:

$$a' = \text{argmax}\{\delta(a) | a \in \mathbf{A}\} \quad (4)$$

2.2 Visual Cues

The cues considered in the integration process are:

Correlation - With the standard sum of squared differences

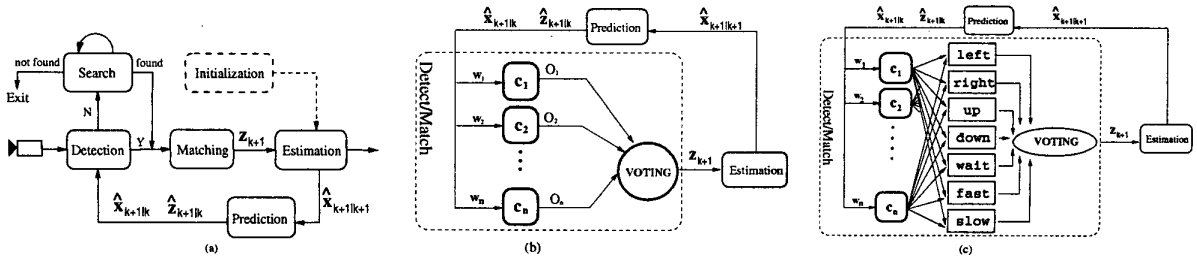


Figure 1: A schematic overview of the a) proposed tracking system, b) *response fusion* and c) *action fusion*.

(SSD), the position of the target is found at the lowest dissimilarity score:

$$SSD(u,v) = \sum_n \sum_m [I(u+m, v+n) - T(m,n)]^2 \quad (5)$$

where $I(u,v)$ and $T(u,v)$ represent the grey level values of the image and the template, respectively.

Color - Color is represented by r and g component in the Chromatic Color space [1].

Motion - Motion detection is based on computation of the temporal derivative using image differencing:

$$M[(u,v),k] = \mathcal{H}[|I[(u,v),k] - I[(u,v),k-1]| - \Gamma] \quad (6)$$

where Γ is a fixed threshold and $\mathcal{H}$ is defined as:

$$\mathcal{H}(x) = \begin{cases} 0 & : \ x \leq 0 \\ x & : \ x > 0 \end{cases} \quad (7)$$

Intensity Variation - In each frame, the following is estimated for all $m \times m$ (details about m are given in Section 3.2) regions inside the tracked window:

$$\sigma^2 = \frac{1}{m^2} \sum_u \sum_v [I(u,v) - \bar{I}(u,v)]^2 \quad (8)$$

where $\bar{I}(u,v)$ is the mean intensity value estimated for the window. For example, for a mainly uniform region, low variation is expected during tracking. The level of texture is evaluated as proposed in [11].

2.3 Weighting

In (Eq. 3), the output from each cue is weighted. Four different weighting methods are evaluated:

1. Uniform weights - Outputs of all cues are weighted equally: $w_i = 1/n$, where n is the number of cues.

2. Texture based weighting - Weights are estimated experimentally and depend on the spatial content of the region. For a highly textured region, we use: color (0.25), image differencing (0.3), correlation (0.25), intensity variation (0.2). For uniform regions: color (0.45), image differencing (0.2), correlation (0.15), intensity variation (0.2).

3. One-step distance weighting - Weighting factor, w_i, of a cue, c_i, at time step k depends on the distance from the predicted image position, $\hat{\mathbf{z}}_{k|k-1}$. Initially, the distance is estimated as $d_i = ||\mathbf{z}_k^i - \hat{\mathbf{z}}_{k|k-1}||$ and errors are estimated as $e_i = d_i / \sum_{i=1}^n d_i$. Weights are than inversely proportional to the error with $\sum_{i=1}^n w_i = 1$.

4. History-based distance weighting - Weighting factor of a cue depends on its overall performance during the tracking sequence. The performance is evaluated by observing how many times the cue was in an agreement with the rest of the cues. The strategy is:

1. For each cue, c_i, examine if $||\mathbf{z}_k^i - \mathbf{z}_k^j|| < d_T$ where $i, j = 1, \ldots, n$ and $i \neq j$. If this is true, a_{ij}=1, otherwise a_{ij}=0. Here, a_{ij}=1 means there is an agreement between the outputs of cues i and j at that voting cycle and d_T represents a distance threshold which is set in advance.
2. Build $(n-1)$ value set for each cue: c_i : $\{a_{ij} | j = 1, \ldots, n \text{ and } i \neq j\}$. Find sum $s_i = \sum_{j=1}^n a_{ij}$.
3. The accumulated values during N tracking cycles, $S_i = \sum_{k=1}^N s_i^k$, indicate how many times a cue, c_i, was in the agreement with other cues. Weights are then simply proportional to this value: $w_i = \frac{S_i}{\sum_{i=1}^n S_i}$ with $\sum_{i=1}^n w_i = 1$.

3 Implementation

We propose two approaches where voting is used for: i) *response fusion*, and ii) *action fusion*. The first approach makes the use of "raw" responses from the employed visual cues in the image which also represents the action space, **A**. Here, the response is represented either by a binary function (yes/no) answer, or in the interval [0,1] (these values are scaled between [0,255] to allow visual monitoring). The second approach uses a different action space represented by a *direction* and a *speed*, see Fig. 2. Compared to the first approach, where the position of the tracked region is estimated, this approach can be viewed as estimating its velocity. Again, each cue votes for different actions from the action space, **A**, which is now the velocity space.

3.1 Initialization

According to Fig. 1(a), a tracking sequence should be initiated by *detecting* the target object. In [4] it is proposed that *selectors* should be employed which are defined as heuristics that selects regions possibly occupied by the target. Based on this ideas, color and image differences are used to detect the target in the first image. These two cues are also used in cases

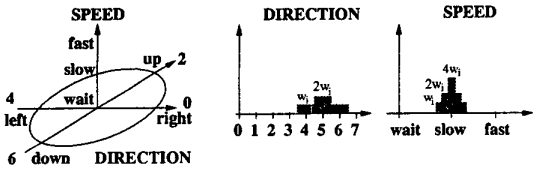


Figure 2: *Action fusion* approach: the desired direction is (*down and left)* with a (*slow*) speed.

where the target has either i) left the field of view, or ii) it was occluded for a few frames.

3.2 Response Fusion Approach

After the target is located, a template is initialized which is used by the correlation cue. In each frame, a color image of the scene is acquired. Inside the window of attention the response of each cue, denoted O_i, is evaluated, see Fig. 1(b). Here, $\mathbf{x}$ represents a position:

Color - During tracking, all pixels whose color falls in the pre–trained color cluster are given value between [0, 255]:

$$0 \leq O_{color}(\mathbf{x},k) \leq 255 \quad \text{with} \quad \mathbf{x} \in [\hat{\mathbf{z}}_{k|k-1} - 0.5\mathbf{x}_w,\ \hat{\mathbf{z}}_{k|k-1} + 0.5\mathbf{x}_w] \tag{9}$$

where $\mathbf{x}_w$ is the size of the window of attention.

Motion- Using (Eq. 6) and (Eq. 7) with $\Gamma = 10$, image is segmented into regions of motion and inactivity:

$$0 \leq O_{motion}(\mathbf{x},k) \leq 255 - \Gamma \quad \text{with} \quad \mathbf{x} \in [\hat{\mathbf{z}}_{k|k-1} - 0.5\mathbf{x}_w,\ \hat{\mathbf{z}}_{k|k-1} + 0.5\mathbf{x}_w] \tag{10}$$

Correlation - Here, the output is given by:

$$O_{SSD}(\mathbf{x},k) = 255e^{(-\frac{\bar{\mathbf{x}}^2}{2\sigma^2})} \quad \text{with} \quad \sigma = 5 \qquad \mathbf{x} \in [\mathbf{z}_{SSD} - 0.5\mathbf{x}_w,\ \mathbf{z}_{SSD} + 0.5\mathbf{x}_w], \bar{\mathbf{x}} \in [-0.5\mathbf{x}_w,\ 0.5\mathbf{x}_w] \tag{11}$$

with Gaussian centered at the peak of the SSD surface.

Intensity variation - Here, (Eq. 8) is used. If a low variation is expected, all pixels inside a $m \times m$ region are given values (255-σ). If a large variation is expected, pixels are assigned σ value directly. m depends on the size of the window of attention with $m = 0.2\mathbf{x}_w$:

$$0 \leq O_{var}(\mathbf{x},k) \leq 255 \quad \text{with} \quad \mathbf{x} \in [\hat{\mathbf{z}}_{k|k-1} - 0.5\mathbf{x}_w,\ \hat{\mathbf{z}}_{k|k-1} + 0.5\mathbf{x}_w] \tag{12}$$

Fusion:

The responses are integrated using (Eq. 3):

$$\delta(\mathbf{x},k) = \sum_i^n w_i O_i(\mathbf{x},k) \tag{13}$$

However, (Eq. 4) can not be directly used since there might be several pixels with same number of votes. Therefore, this equation is slightly modified to accommodate for this:

$$\delta'(\mathbf{x},k) = \begin{cases} 1: & \text{if } \delta(\mathbf{x},k) \text{ is } \operatorname{argmax}\{\delta(\mathbf{x}',k)|\mathbf{x}' \\ & \in [\hat{\mathbf{z}}_{k|k-1} - 0.5\mathbf{x}_w, \hat{\mathbf{z}}_{k|k-1} + 0.5\mathbf{x}_w]\} \\ 0: & \text{otherwise} \end{cases} \tag{14}$$

Finally, the new measurement $\mathbf{z}_k$ is given by the mean value (first moment) of $\delta'(\mathbf{x},k)$, i.e., $\mathbf{z}_k = \bar{\delta}'(\mathbf{x},k)$.

3.3 Action Fusion Approach

Here, the action space is defined by a direction d and speed s, see Fig. 2. Both the direction and the speed are represented by histograms of discrete values where the direction is represented by eight values, see Fig. 1(c):

$$\begin{aligned} &\text{LD}\begin{bmatrix}-1\\1\end{bmatrix}, \text{L}\begin{bmatrix}-1\\0\end{bmatrix}, \text{LU}\begin{bmatrix}-1\\-1\end{bmatrix}, \text{U}\begin{bmatrix}0\\-1\end{bmatrix}, \\ &\text{RU}\begin{bmatrix}1\\-1\end{bmatrix}, \text{R}\begin{bmatrix}1\\0\end{bmatrix}, \text{RD}\begin{bmatrix}1\\1\end{bmatrix}, \text{D}\begin{bmatrix}0\\1\end{bmatrix} \\ &\text{with L-left, R-right, D-down, U-up} \end{aligned} \tag{15}$$

Speed is represented by 20 values with 0.5 pixel interval which means that the maximum allowed displacement between successive frames is 10 pixels (this is easily made adaptive based on the estimated velocity). There are two reasons for choosing just eight values for the direction: i) if the update rate is high or the inter–frame motion is slow, this approach will still give a reasonable accuracy and hence, a smooth performance, and ii) by keeping the voting space rather small there is a higher chance that the cues will vote for the same action. Accordingly, each cue will vote for a desired direction and a desired speed. As presented in Fig. 2 a *neighborhood voting scheme* is used to ensure that slight differences between different cues do not result in an unstable classification. (Eq. 2) is modified so that:

$$\mathbf{H} = \begin{bmatrix}0&0&1&0\\0&0&0&1\end{bmatrix} \text{ and } \mathbf{W} = \begin{bmatrix}\alpha\Delta T & 0 & \beta & 0\\ 0 & \alpha\Delta T & 0 & \beta\end{bmatrix}^T \tag{16}$$

In each frame, the following is estimated for each cue:

Color - The response of the color cue is first estimated according to (Eq. 9) and followed by:

$$\mathbf{a}_{color}(k) = \frac{\sum_{\mathbf{x}} O_{color}(\mathbf{x},k)\mathbf{x}(k)}{\sum_{\mathbf{x}} \mathbf{x}(k)} - \hat{\mathbf{p}}_{k|k-1} \quad \text{with} \quad \mathbf{x} \in [\hat{\mathbf{p}}_{k|k-1} - 0.5\mathbf{x}_w,\ \hat{\mathbf{p}}_{k|k-1} + 0.5\mathbf{x}_w] \tag{17}$$

where $\mathbf{a}_{color}(k)$ represents the desired action and $\hat{\mathbf{p}}_{k|k-1}$ is the predicted position of the tracked region. Same approach is used to obtain $\mathbf{a}_{motion}(k)$ and $\mathbf{a}_{var}(k)$.

Correlation - The minimum of the SSD surface is used as:

$$\mathbf{a}_{SSD}(k) = \operatorname*{argmin}_{\mathbf{x}}(SSD(\mathbf{x},k)) - \hat{\mathbf{p}}_{k|k-1} \tag{18}$$

Fusion:

After the desired action, $\mathbf{a}_i(k)$, for a cue is estimated, the cue produces the votes as follows:

$$\text{direction } d_i = \mathcal{P}(\text{sgn}(\mathbf{a}_i)),\ \text{speed } s_i = ||\mathbf{a}_i|| \tag{19}$$

where $\mathcal{P} : \mathbf{x} \rightarrow \{0,1,\ldots,7\}$ is a scalar function that maps the two–dimensional direction vectors (see (Eq. 15)) to one–dimensional values representing the bins of the direction histogram. Now, the estimated direction, d_i, and the speed, s_i, of a cue, c_i, with a weight, w_i, are used to update the direction and speed of the histograms according to Fig. 2 and (Eq.3). The new measurement is then estimated by multiplying the actions from each histogram which received the maximum number of votes according to (Eq. 4):

$$\mathbf{z}_k = \mathcal{S}(\operatorname*{argmax}_d HD(d)) \operatorname*{argmax}_s HS(s) \tag{20}$$

where $S : x \rightarrow \{\left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right], \ldots, \left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right]\}$. The update and prediction steps are then performed using (Eq. 16) and (Eq. 2). The reason for choosing this particular representation instead of simply using a weighted sum of first moments of the responses of all cues is, as it has been pointed out in [8], that arbitration via vector addition can result in commands which are not satisfactory to any of the contributing cues.

4 Experimental Evaluation

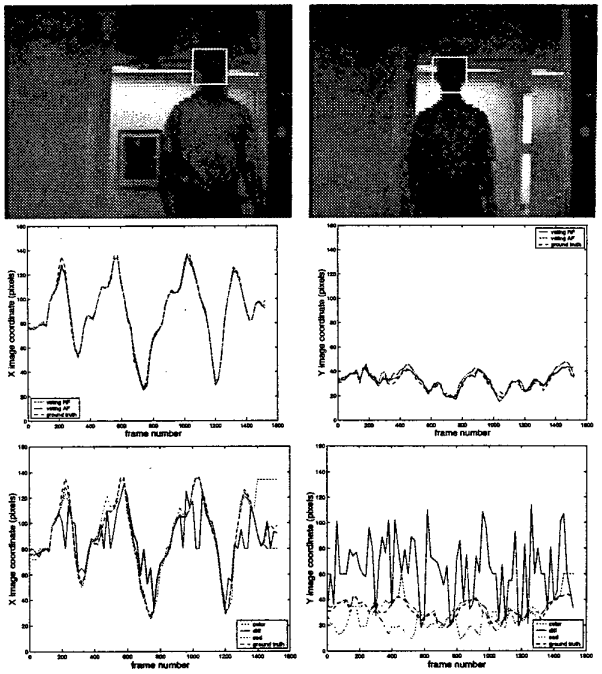
A series of tracking experiments has been performed to experimentally evaluate the proposed approaches. The sequences are obtained in an everyday environment with changing lighting conditions and background clutter. The original size of the images was 320×240 and a standard CCD camera with 6mm focal length was used. Tracking runs at frame-rate on a PC 333MHz Pentium using a standard Matrox Meteor frame–grabber card. The main objectives of the experiments were: i) evaluation of fusion approaches with respect to the weighting methods, and ii) performance of the fusion approaches versus the performance of each cue.

The results are discussed through *accuracy* and *reliability*. The accuracy is expressed using an error measure which is a distance between the ground truth (chosen manually using a reference point on the object) and the currently estimated position of the reference point. The results are summarized through the mean square error and standard deviation in pixels. The measure of the reliability is on a yes/no basis depending on if a cue (or the fused system) successfully tracks the target during a single experiment. The tracking is successful if the object is kept inside the window of attention during the entire test sequence.

4.1 Experiment 1.

Here, the effect of weighting was evaluated. The results are obtained for 10 sequences and for each sequence 3 different sizes of the window of attention were used: 25×25, 35×35 and 45×45 pixels. The reason for this was to test the ability of the system to cope with the background clutter: if the target is small compared to the size of the window of attention, a large portion of the window will belong to the background and therefore the content of the window will change and affect the response of each cue. The target undergoes arbitrary 3D motion.

Accuracy (Table 1) - Here, the distance measure is used as an error indicator. The overall results are presented in Table 1 for the proposed fusion approaches. The results show that the best accuracy is achieved with fixed weights using the *texture based weighting* and the *uniform weighting*. The *one–step* distance weighting gives a reward to a cue each time when the cue performs satisfactorily and there is no ability to determine the overall performance of the cues during the sequence. It was expected that this problem would be solved using the *history based* weighting but, on the other hand, temporal smoothing results in a slow weight assignment dynamics. One solution



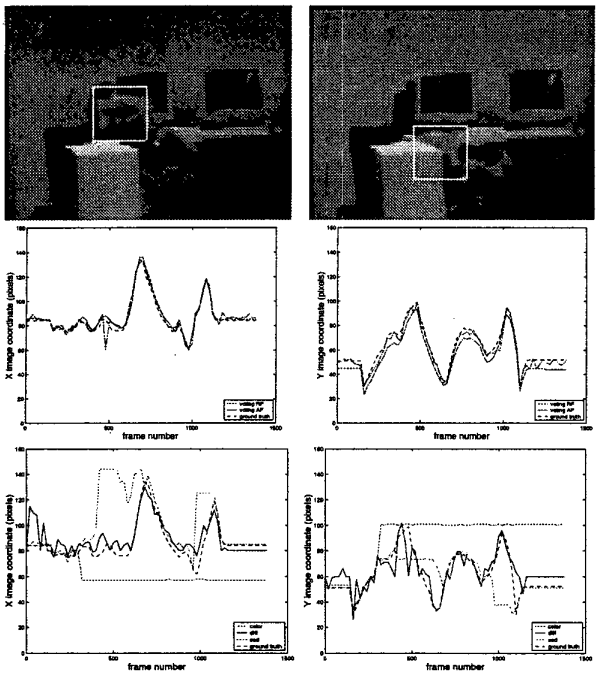
	RF Voting		AF Voting		Color		diff		SSD	
	x	y	x	y	x	y	x	y	x	y
mean	-0.7	0.1	0.2	2	-1.3	2.5	-2.7	31.1	3	-11
std	2.4	2.5	2.5	2.6	4.8	4.5	17.3	25	13.2	11

Figure 3: A comparison between ground truth, voting approaches and individual cues during a person tracking.

to this problem might be to change the model and instead of using all frames up to the current one, apply a temporal windowing approach. This would allow the use of the immediate history to evaluate the performance of each cue.

Comparing the performance for fusion approaches shows that action fusion approach had higher standard deviation (14 pixels for texture based weighting). The reason for this is the choice of the underlying voting space. For example, if the color cue shows a stable performance for a number of frames, its weight will be high compared to the other cues (or it might have been set to a high value from the beginning). In some cases, two colors are used at the same time. When an occlusion occurs, the position of the center of the mass of the color blob will change fast (and sometimes in different directions) which results in abrupt changes in both direction and speed. The other method, *response fusion*, on the other hand, does not suffer from this which results in a lower standard deviation value.

In many cases it is, however, more important to retain the tracking at the cost of a lower accuracy. For that purpose the *reliability* measure is important.



	RF Voting		AF Voting		Color		Diff		SSD	
	x	y	x	y	x	y	x	y	x	y
mean	1.5	-6.4	1.5	-1.7	-24	28	3.2	2.5	14.9	-2.4
std	2	1.9	4	3	19.4	20.8	8.3	9.5	22	16

Figure 4: A comparison between ground truth, voting approaches and individual cues in case of occlusions.

	Uniform Weights		Texture Weighting		One-step Dist. Weighting		Hist. Based Weighting	
	mse	std	mse	std	mse	std	mse	std
RF	10	9	7	6	17	14	17	14
AF	9	14	8	14	13	14	15	14

Table 1: Qualitative results (pixels) for 30 sequences and all weighting techniques.

Reliability (Table 2) - Here, the influence of choice of the weight assignment technique on the success rate of the response and action fusion approaches is discussed. As for the accuracy, the reliability was estimated for 30 test runs and the percentage of the success is presented. Ranking the results shows that the *texture based weighting* performed most reliably - the target was successfully tracked during 27 test runs.

Comparing the overall results, *texture weighting* approach resulted in both the highest accuracy and reliability. *Uniform weighting*, although very accurate according to the results in Table 1, performed worst in terms of reliability.

	Uniform Weights	Texture Weighting	One-step Dist. Weighting	Hist. Based Weighting
RF	76.7 %	90 %	83.3 %	80 %
AF	43.3 %	73.3 %	66.7 %	66.7 %

Table 2: The influenceof the weight assignment techniques on the success rate.

4.2 Experiment 2.

This experiment evaluated the performance of the proposed voting approaches as well as the performance of individual cues with respect to three sensor–object configurations typically used in visual–servoing systems: i) static sensor/moving object ("stand–alone camera system"), ii) moving sensor/static object ("eye–in–hand camera" servoing toward a static object), and iii) moving sensor/moving object (camera system on a mobile platform or eye–in–hand camera servoing toward a moving object). The results are presented as in the previous experiment, with respect to the accuracy and reliability. The two fusion approaches as well as the individual cues have been tested with respect to the ability to cope with occlusions of the target and to regain tracking after the target has left the field of view for a number of frames. The results are presented for correlation, color and image differences since the intensity variation cue can not be used alone for tracking.

Accuracy (Table 3) - The best accuracy is achieved using the *response fusion* approach. Although the *mse* is similar for the *action fusion* approach in cases of *static sensor/moving object* and *moving sensor/static object* configurations, *std* is higher. The reason for this is, as in the previous experiment, the choice of the underlying voting space. The comparison of the performance of the fusion approaches and the performance of the individual cues shows the necessity for fusion. Image differences alone can not be used in cases of *moving sensor/static object* and *moving sensor/moving object* configurations since there is no ability to differ between the object and the background. During most of the sequences the target undergoes 3D motion which results with scale changes and rotations not modeled by SSD. It is obvious that these factors will affect this cue significantly resulting with a large error as demonstrated in the table. This problem may be solved by using a better model (see [9]). It can also be seen that the color cue performed best of the individual cues.

In the case of *moving sensor/static object*, after the tracking is initialized the color cue "sticks" to the object during the sequence and, since the background varies a little, the best accuracy is achieved compared to other configurations. During other two configurations the background will change containing also the color same as the target's. This distracts the color tracker resulting in increased error. The error is larger in the case of *static sensor/moving object* compared to *moving sensor/moving object* since in the test sequences the background included the target's color more often.

Reliability (Table 4) - Since the accuracy is obtained using the *texture based weighting* the reliability for the *action* and *response* fusion will be same as presented in Table 2 for this weighting technique. In Table 4, the obtained reliability

	static sensor/ moving object		moving sensor/ static object		moving sensor/ moving object	
	mse	std	mse	std	mse	std
RF	7	7	4	3	9	10
AF	7	9	4	10	13	25
Color	15	16	10	6	10	14
Diff	23	26	failed	failed	failed	failed
SSD	25	27	12	13	17	21

Table 3: Qualitative results for various sensor-object configurations (in pixels).

results are ranked showing that color performs most reliably compared to other individual cues. In certain cases, especially when the influence of the background is not significant, this cue will perform satisfactorily. However, it will easily get distracted if the background takes a large portion of the window of attention and includes the target's color.

Image differencing will depend on the size of the moving target with respect to the size of the window of attention and variations in lighting. In structured environments, however, this cue may perform well and may be considered in cases of a single moving target where the size of the target is small compared to the size of the image (or window of attention).

Fig. 3 shows two example images and the tracking accuracy for the proposed fusion approaches and for each of the cues individually. The plots and the table show the deviation from the ground truth value (in pixels). A significant instability is demonstrated for image differencing indicating that this cue can not be used alone for tracking. It may also be seen that the color cue performed really well. This is not surprising since many of the face- or people–tracking systems rely strongly on this cue. Very little texture and significant changes in scale implies that correlation cue is very likely to fail. Similar results are shown in Fig. 4 for a case where the target is a package of raisins. During this sequence, a number of occlusion occurs (as demonstrated in the images), but the plots demonstrate a stable performance of the fusion approaches during the whole sequence. The color cues is, however, "fooled" by the box which is the same color as the target. The plots demonstrate how this cue fails around frame 300 and never regains tracking after that. These two examples clearly demonstrate that tracking by fusion is more superior than any of the individual cues.

	# success	# failure	%
RF Voting	27	3	90
AF Voting	22	8	73.3
Color	18	12	60
SSD	12	18	40
Diff.	7	23	23.3

Table 4: Success rate for individual cues and fusion approaches.

5 Summary and Conclusions

We have developed a visual tracking system where the consensus of simple visual cues facilitates both robust and real–time performance. Response and action fusion in a voting based framework are proposed. The approaches differ in the choice of the underlying voting spaces. Special emphasis was put on the evaluation of the cues' weights where four methods have been used. The experimental evaluation considered the effect of the weighting technique to the performance of the fusion approaches. It has been shown that the most accurate and reliable results are obtained using the *texture based weighting* where the weights are set in advance and kept constant during tracking. The system was also evaluated with three most common camera–object configurations: i) static sensor/moving object, ii) moving sensor/static object and iii) moving sensor/moving object. Experimental evaluation has shown that the best performance was obtained using the *response fusion* approach. On 30 test runs this approach was successful in 90% of cases compared to 73.7% for the *action fusion* approach. To demonstrate the necessity for the fusion, color, differencing and correlation were also evaluated individually. It has been shown that color performed best. Color is commonly used in cases where the object is uniform in color and, given some assumptions about the environment, it should be considered as a strong tracking candidate. The results are presented for a monocular system and our future work will consider binocular cues in the fusion process.

References

[1] D. Kragic. "Visual Servoing for Manipulation: Robustness and Integration Issues", *PhD thesis*, Royal Institute of Technology, Stockholm, Sweden, 2001.

[2] C. Rasmussen and G. Hager, "Joint probabilistic techniques for tracking objects using multiple visual cues", *IEEE IROS*, p 191–196, 1998.

[3] Y. Shirai and R. Okada and T. Yamane, "Robust visual tracking by integrating various cues", *in M. Vincze and G. Hager, eds, 'Robust Vision for Manipulation'*, p 53–66, 2000.

[4] K. Toyama and G. Hager, "Incremental focus of attention for robust visual tracking", *CVPR*, p 189 –195, 1996.

[5] B. Parhami, "Voting algorithms", *IEEE Trans. on Reliability* **43**(3),p 617–629, 1994.

[6] J. Clark and A. Yuille, *Data fusion for sensory information precessing systems*, Kluwer Academic Publisher, 1990.

[7] D. Blough and G. Sullivan, "Voting using predispositions", *IEEE Trans. on Reliability* **43**(4), p 604–616, 1994.

[8] J. Rosenblatt and C. Thorpe, "Combining multiple goals in a behavior-based architecture", *IEEE IROS*, p 136–141, 1995.

[9] G. Hager and K. Toyama, "The XVision system: A general–purpose substrate for portable real–time vision applications", *Comp. Vision and Im. Understanding* **69**(1), p 23–37, 1996.

[10] Y. Bar-Shalom and Y. Li, *Estimation and Tracking:Principles, techniques and software*, Artech House.

[11] J. Shi and C. Tomasi, "Good features to track", *CVPR'94*, p 593–600