Pattern Recognition, 26th DAGM Symposium, Springer-Verlag, Berlin, pages 359–366

Multi-Step Entropy Based Sensor Control for Visual Object Tracking

Benjamin Deutsch[1⋆], Matthias Zobel[1⋆], Joachim Denzler[2], and Heinrich Niemann[1]

[1] Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg
91058 Erlangen, Germany
{deutsch, zobel, niemann}@informatik.uni-erlangen.de
[2] Arbeitsgruppe Rechnersehen, Universität Passau
94030 Passau, Germany
denzler@fmi.uni-passau.de

Abstract. We describe a method for selecting optimal actions affecting the sensors in a probabilistic state estimation framework, with an application in selecting optimal zoom levels for a motor-controlled camera in an object tracking task. The action is selected to minimize the expected entropy of the state estimate. The contribution of this paper is the ability to incorporate varying costs into the action selection process by looking multiple steps into the future. The optimal action sequence then minimizes both the expected entropy and the costs it incurs.
This method is then tested with an object tracking simulation, showing the benefits of multi-step versus single-step action selection in cases where the cameras' zoom control motor is insufficiently fast.

1 Introduction

This paper describes a method for selecting *optimal actions* which affect the sensors in a probabilistic state estimation framework. The contribution of this paper is the ability to incorporate *varying costs* into the action selection process by looking *multiple steps* into the future.

Probabilistic state estimation systems continuously estimate the current state of a dynamic system based on observations they receive, and maintain this estimate in the form of a probability density function. Given the possibility to affect the observation process with certain *actions*, what are the optimal actions, in an information theoretic sense, that the estimation system should choose to influence the resulting probability density?

One sample application is the selection of optimal camera actions in motor-operated cameras for an active object tracking task, such as pan and tilt operations or zooming. We examine focal length selection as our sample application, using an extended Kalman filter for state estimation.

⋆ This work was partly funded by the German Research Foundation (DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.

Previous work in the areas of object recognition [10, 4, 3] have shown that an active viewpoint selection process can reduce uncertainty. For object tracking, active focal length selection is used to keep the target's scale constant [6, 11]. Yet the focus of these works is not to find the optimal zoom level.

The information theoretic solution described in [5], which this work is based on, uses the *entropy* of the estimated state distribution. This system calculates the *expected entropy* for each action, and then chooses the action where the expected entropy is lowest.

However, this approach only works if all actions are considered equal. If the actions incur costs which may depend on the last action, examining the expected benefit of just a single action is no longer sufficient. In the example of focal length selection, the zoom lens motor has only a finite speed. A too high zoom level can cause the object to be lost when it approaches the edges of the camera image faster than the zoom motor can follow.

The solution is to obtain the best *sequence* of future actions, and to calculate the costs and benefits of the sequence as a whole. In our case of a motorized zoom lens, the tracker is able to reduce the focal length in advance, in order for the low focal length to actually be available in the time frame where it is needed.

In simulated experiments with slow zoom motors, up to 82% less object loss was experienced, as compared to the original single-step method. This reduced the overall state estimation error by up to 56%.

The next section contains a short review of the Kalman filter and the notation used in this paper. Section 3 simultaneously reviews the single-step method from [5] and shows how to extend it to multiple steps, the main contribution of this paper. The method is evaluated in section 4, and section 5 concludes the paper and gives an outlook for future work.

2 Review: Kalman Filter

As in [5], we operate on the following discrete-time dynamic system: At time t, the state of the system is described in the state vector $\boldsymbol{x}_t \in \mathbb{R}^n$, which generates an observation $\boldsymbol{o}_t \in \mathbb{R}^m$. The state change and observation equations are

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, t) + \boldsymbol{w} \quad , \quad \boldsymbol{o}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{a}_t) + \boldsymbol{r} \tag{1}$$

where $\boldsymbol{f}(\cdot,\cdot) \in \mathbb{R}^n$ is the state transition function and $\boldsymbol{h}(\cdot,\cdot) \in \mathbb{R}^m$ the observation function. $\boldsymbol{w}$ and $\boldsymbol{r}$ are normal zero-mean error processes with covariance matrices $\boldsymbol{W}$ and $\boldsymbol{R}$.

The parameter $\boldsymbol{a}_t \in \mathbb{R}^l$ is called the *action* at time t. It summarizes all the parameters which affect the observation process. For object tracking, $\boldsymbol{a}_t$ might include the pan, tilt and the focal length of each camera. The action is performed *before* the observation is made.

The task of the state estimator is to continuously calculate the distribution $p(\boldsymbol{x}_t | \langle \boldsymbol{o} \rangle_t, \langle \boldsymbol{a} \rangle_t)$ over the state, given the sequence $\langle \boldsymbol{o} \rangle_t$ of all observations and the sequence $\langle \boldsymbol{a} \rangle_t$ of all actions taken up to, and including, time t.

Assuming the action is (for now) known and constant, the Kalman filter [8], a standard algorithm, can be used for state estimation. Since the observation function is based on the non-linear perspective projection model, an *extended Kalman filter* [1] is necessary. A full description of the extended Kalman filter is beyond the scope of this paper. We use the following notation for the filter: $\widehat{\boldsymbol{x}}_t^-$ and $\widehat{\boldsymbol{x}}_t^+$ are the *a priori* and *a posteriori* state estimate means at time t. $\boldsymbol{P}_t^-$ and $\boldsymbol{P}_t^+$ are the covariance matrices for the a priori and a posteriori state estimates. The extended Kalman filter performs the following steps for each time-step t:

1. State mean and covariance prediction:

$$\widehat{\boldsymbol{x}}_t^- = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{t-1}, t-1) \quad , \quad \boldsymbol{P}_t^- = \boldsymbol{f}_t^{\boldsymbol{x}} \boldsymbol{P}_{t-1} \boldsymbol{f}_t^{\boldsymbol{x}\,T} + \boldsymbol{W} \quad . \tag{2}$$

2. Computation of the filter gain:

$$\boldsymbol{K}_t = \boldsymbol{P}_t^- \boldsymbol{h}_t^{\boldsymbol{x}\,T}(\boldsymbol{a}_t) \left(\boldsymbol{h}_t^{\boldsymbol{x}}(\boldsymbol{a}_t) \boldsymbol{P}_t^- \boldsymbol{h}_t^{\boldsymbol{x}\,T}(\boldsymbol{a}_t) + \boldsymbol{R} \right) \quad . \tag{3}$$

3. State mean and covariance update by incorporating the observation

$$\widehat{\boldsymbol{x}}_t^+ = \widehat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t \left(\boldsymbol{o}_t - \boldsymbol{h}(\widehat{\boldsymbol{x}}_t^-, \boldsymbol{a}_t) \right) \quad , \quad \boldsymbol{P}_t^+(\boldsymbol{a}_t) = \left(\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{h}_t^{\boldsymbol{x}}(\boldsymbol{a}_t) \right) \boldsymbol{P}_t^- \quad . \tag{4}$$

$\boldsymbol{f}_t^{\boldsymbol{x}}$ and $\boldsymbol{h}_t^{\boldsymbol{x}}(\boldsymbol{a}_t)$ denote the Jacobians of the state transition and observation functions. Since the observation Jacobian $\boldsymbol{h}_t^{\boldsymbol{x}}(\boldsymbol{a}_t)$ depends on the selected action $\boldsymbol{a}_t$, the a posteriori state covariance does, too. In cases where no observation is made in a time step, the a posteriori state estimate is equal to the a priori one.

3 Multi-step optimal actions

The method described in [5] uses the entropy of the state distribution to select the next action for a single step in the future. The single-step approach works well if the optimal action can be performed at each time-step. Often, however, there will be real-world constraints on which actions are possible; for example, cameras with a motorized zoom lens can only change their focal lengths at a finite maximal speed. In general, we say that an action, or a sequence of actions, incurs a *cost*. This cost must be subtracted from the expected benefits of the actions to find the truly optimal actions.

In the case of focal length selection, the single-step method will often select a large focal length when the object is in the center of the camera image. Once the object moves towards the edge, a lower focal length is needed in order not to lose the object; this focal length may be to far for the zoom motors. The multi-step method, evaluating a sequence of actions, will detect the need for a low focal length sooner, and will start reducing the focal length ahead of time.

To evaluate an action, we use the *entropy* [2] of the state distribution as a measure of uncertainty. This measure was used in [5] to select a single action. We will show how this method can be expanded to a sequence of actions.

To evaluate a sequence of actions, we measure the entropy of the state distribution at the *horizon*. The horizon k is the number of steps to be looked

ahead, starting at time step t. For the single-step variant, $k = 1$. We denote the sequences of *future* actions and observations, occurring between time steps $t+1$ and $t+k$, as $\langle \boldsymbol{a} \rangle^k$ and $\langle \boldsymbol{o} \rangle^k$, respectively.

The entropy of the a posteriori state belief $p(\widehat{\boldsymbol{x}}_{t+k} | \langle \boldsymbol{o} \rangle_{t+k}, \langle \boldsymbol{a} \rangle_{t+k})$ is

$$H(\boldsymbol{x}^+_{t+k}) = -\int p(\boldsymbol{x}_{t+k} | \langle \boldsymbol{o} \rangle_{t+k}, \langle \boldsymbol{a} \rangle_{t+k}) \log(p(\boldsymbol{x}_{t+k} | \langle \boldsymbol{o} \rangle_{t+k}, \langle \boldsymbol{a} \rangle_{t+k})) d\boldsymbol{x}_{t+k} \ . \quad (5)$$

This gives us information about the final a posteriori uncertainty, provided actions $\langle \boldsymbol{a} \rangle^k$ were taken and observations $\langle \boldsymbol{o} \rangle^k$ were observed.

However, to determine the optimal actions *before* the observations are made, this measure cannot be used directly. Instead, we determine the *expected entropy*, given actions $\langle \boldsymbol{a} \rangle^k$, by averaging over all observation sequences:

$$H(\boldsymbol{x}_{t+k} | \langle \boldsymbol{o} \rangle^k, \langle \boldsymbol{a} \rangle^k) = \int p(\langle \boldsymbol{o} \rangle^k | \langle \boldsymbol{a} \rangle^k) H(\boldsymbol{x}^+_{t+k}) \ d \ \langle \boldsymbol{o} \rangle^k \quad . \quad (6)$$

This value is called the *conditional entropy* [2]. The notation $H(\boldsymbol{x}_t | \boldsymbol{o}_t, \boldsymbol{a}_t)$ is misleading, but conforms to that used in information theory textbooks. The only free parameter is the action sequence $\langle \boldsymbol{a} \rangle^k$. The optimal action sequence can then be found by minimizing the conditional entropy.

In the case of a Gaussian distibution, as is used throughout the Kalman filter, the entropy takes the following closed form:

$$H(\boldsymbol{x}_{t+k} | \langle \boldsymbol{o} \rangle^k, \langle \boldsymbol{a} \rangle^k) = \int p(\langle \boldsymbol{o} \rangle^k | \langle \boldsymbol{a} \rangle^k) \left(\frac{n}{2} + \frac{1}{2} \log \left((2\pi)^n | \boldsymbol{P}^+_{t+k}(\langle \boldsymbol{a} \rangle^k) | \right) \right) d\langle \boldsymbol{o} \rangle^k. \quad (7)$$

We note that only $p(\langle \boldsymbol{o} \rangle^k | \langle \boldsymbol{a} \rangle^k)$ depends on the integrand $\langle \boldsymbol{o} \rangle^k$, the covariance $\boldsymbol{P}^+_{t+k}(\langle \boldsymbol{a} \rangle^k)$ does not. This allows us to place everything else outside the integration, which then intergrates over a probability density function and is therefore always 1. Therefore, we only need to obtain the a posteriori covariance matrix $\boldsymbol{P}^+_{t+k}$ to evaluate an action sequence, which means stepping through the Kalman filter equations k times. Since we do not have any future observations $\boldsymbol{o}$, the state estimate mean $\widehat{\boldsymbol{x}}^-$ can only be updated with the expected observation $\boldsymbol{h}(\widehat{\boldsymbol{x}}^-, \boldsymbol{a})$, which reduces equation (4) to $\widehat{\boldsymbol{x}}^+ = \widehat{\boldsymbol{x}}^- + \boldsymbol{0}$. The state estimate mean allows us to calculate all used Jacobians for equations (2) and (3), which give us all covariance matrices $\boldsymbol{P}^-$ and $\boldsymbol{P}^+$ for any future time step.

In cases where an observation is not guaranteed, the final entropy is based on either the a posteriori or the a priori covariance matrix. The conditional entropy must take this into account. We define an observation to be either *visible* or *non-visible*. For example, in the case of object tracking, an observation is visible if it falls on the image plane of both cameras, and non-visible otherwise. It is important to note that a non-visible observation is still an element of the set of all observations. For a single step, splitting the observations into visible and non-visible ones results in the following entropy:

$$H(\boldsymbol{x}_t | \boldsymbol{o}_t, \boldsymbol{a}_t) = \int\limits_{\{\boldsymbol{o}_t \ \text{visible}\}} p(\boldsymbol{o}_t | \boldsymbol{a}_t) H_v(\boldsymbol{x}^+_t) d\boldsymbol{o}_t + \int\limits_{\{\boldsymbol{o}_t \ \neg\text{visible}\}} p(\boldsymbol{o}_t | \boldsymbol{a}_t) H_{\neg v}(\boldsymbol{x}^-_t) d\boldsymbol{o}_t \quad (8)$$

In the Kalman filter case, where $H_v(\widehat{\boldsymbol{x}}_t^+)$ and $H_{\neg v}(\widehat{\boldsymbol{x}}_t^-)$ do *not* depend on $\boldsymbol{o}_t$, they can again be moved outside the integration. The remaining integrations now reflect the probability of a visible (w_1) or non-visible (w_2) observation:

$$H(\boldsymbol{x}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) = w_1 \cdot H_v(\boldsymbol{x}_t^+) + w_2 \cdot H_{\neg v}(\boldsymbol{x}_t^-) \tag{9}$$

w_1 and w_2 can be solved efficiently using the Gaussian error function [5].

In the multi-step case with a horizon of k, there are 2^k different cases of visibility, since an observation may be visible or not at each time step, and hence 2^k different possible entropies must be combined. If we can calculate the probability and the a posteriori entropy at step $t+k$ for each case, we can again obtain the conditional entropy by a weighted sum:

$$\begin{aligned} H(\boldsymbol{x}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) = {} & w_{vv\ldots v} H_{vv\ldots v}(\boldsymbol{x}_t) + w_{vv\ldots n} H_{vv\ldots n}(\boldsymbol{x}_t) \\ & + \ldots + w_{nn\ldots n} H_{nn\ldots n}(\boldsymbol{x}_t) \end{aligned} \tag{10}$$

where $vv\ldots v$ denotes the case where every time step yields a visible observation, $vv\ldots n$ denotes all visible except for the last, and so on. For such a sequence of visibilities, the probabilities and covariance matrices can be calculated by using the a priori or a posteriori covariance from the previous step as the starting point, and proceeding as in the single-step case.

This can be summarized in a recursive algorithm: For time step l, starting at $l = 1$, the Kalman filter equations use the current action $\boldsymbol{a}_{t+l}$ to produce the correct state mean ($\widehat{\boldsymbol{x}}_{t+l}^+$) and covariance ($\boldsymbol{P}_{t+l}^+$, $\boldsymbol{P}_{t+l}^-$) predictions for both cases of visibility, as well as the probabilities w_1 and w_2 for each case. If $l = k$, the conditional entropy is calculated as in equation (9), using entropies obtained from both covariance matrices through equation (7). Otherwise, this procedure is repeated twice for time $l+1$: once using $\boldsymbol{P}_{t+1}^+$ as its basis for the visible case, and once using $\boldsymbol{P}_{t+1}^-$. Both repetitions (eventually) return a conditional entropy for all steps beyond l, and these are combined according to w_1 and w_2 into the conditional entropy for time step l to be returned.

4 Experiments

This algorithm was evaluated in a simulated object tracking system. Current computational restrictions make a meaningful evaluation in a real-world environment impossible, since the insufficient speed of the zoom motors, a key aspect of the problem, is no longer present.

The following simulated setup, as shown in figure 1, was used: The target object follows a circular pathway. The sensors are two cameras with parallel lines of sight and a variable focal length. The cameras are 200 units apart. The center of the object's path is centered between the two cameras, at a distance of 1500 units, its radius is 200 units.

Simulations were performed with horizons of 1, 2, 3 and 4, and with zoom motor speeds 3, 4 and 5 motor steps per time step, for a total of 12 different experiments. Each experiment tracked the object for 10 full rotations in 720

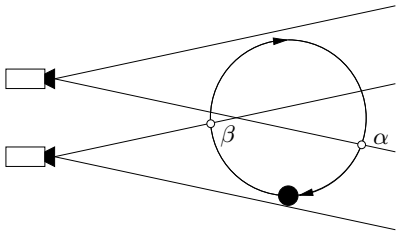

Fig. 1. Simulation setup. The object moves on a circular path. At points α and β, object loss may occur due to limited zoom motor speed.

time steps. For comparison, one experiment was also conducted with fixed focal lengths, and one with unlimited motor speed. In our implementation, a Pentium processor at 2.8 GHz takes less than two minutes for a horizon length of 1 (including output). An experiment with horizon length 4 takes about 6 hours. This implementation interates over the entire action space, without concern for efficiency. Section 5 lists several enhancements with the potential for drastic speed increases to possibly real-time levels.

Figure 2 (left) shows the number of time steps with visible observations, out of a total of 720, for each experiment. The lower the value, the longer the object was lost. The object was typically lost near points α or β in figure 1, at which the object approaches the border of a camera's image plane faster than the zoom motor can reduce the focal length.

Figure 2 (right) shows the actual focal lengths selected by the lower camera in figure 1. Two cycles from the middle of the experiments are shown. The experiments being compared both use a motor zoom speed of 3, and a horizon length of 1 and 4. Additionally, the focal lengths which occur when the zoom motor speed is unlimited are shown. One can see that a larger horizon produces similar focal lengths to a single-step system, but it can react sooner. This is visible between time steps 190 and 210, where the four-step lookahead system starts reducing the focal length ahead of the single-step variant. This results in reduced object loss. The plateaus at time steps 170 and 240 result from the object being lost in the other camera, increasing the state uncertainty.

Table 1, lastly, shows the mean state estimation error, as compared to the ground truth state. The advantage of a multi-step system is greatest in the case of a slow zoom motor (top row), where the increased probability of a valid observation more than makes up for the slight increase in information which the single-step system obtains with its larger focal lengths. This advantage diminishes once the zoom motors are fast enough to keep up with the object. The second-to-last row shows the mean error for a horizon of 1 and an unlimited motor speed. This is the smallest error achievable by using variable focal lengths. The last row contains the mean error for the largest fixed focal length which suffered no object loss. An active zoom can reduce this error by up to 45%, but only if the zoom motor is fast enough to avoid most object loss.

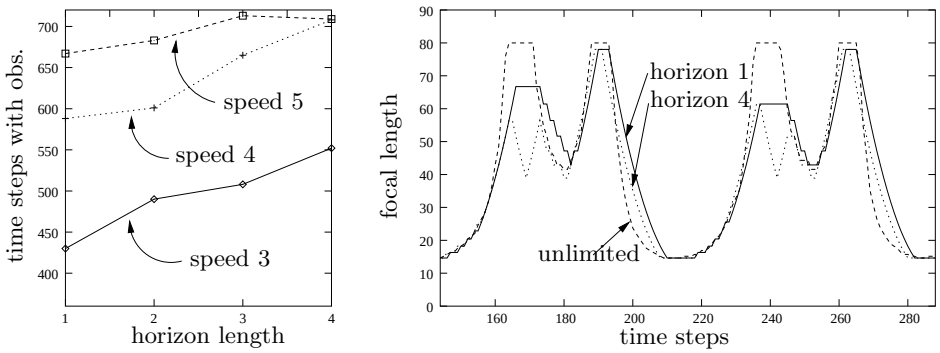

Fig. 2. *Left:* Number of time steps with observations, from a total of 720, for each experiment. Lower values mean greater object loss. *Right:* Focal lengths for two object cycles at a zoom motor speed of 3 and horizons of 1 and 4. The focal lengths from an unlimited motor speed are also shown.

5 Conclusion and Outlook

The methods presented in this paper implement a new and fundemental method for selecting information theoretically optimal sensor actions, with respect to a varying cost model, by predicting the benefit of a given sequence of actions several steps into the future. For the example of focal length selection, we have shown that, given a small action range, this multi-step approach can alleviate the problems that the single-step method faces. In our experiments, we were able to reduce the fraction of time steps with no usable observation by over 80%, which in turn reduced the mean state estimation error by up to 56%.

Future work will focus on reducing the computation time, to enable meaningful real-time experiments, and finally real-time applications, of multi-step action selection. For example, the results from common subexpressions, i.e. the first calculations for two action sequences with a common start, can be cached.

Another optimization is to test only a subset of all possible action sequences, with optimization methods which only rely on point evaluation. Application dependent analysis of the topology of the optimization criterion, such as axis independence and local minimality, may allow more specialized optimization methods. The efficiency may also be improved by intelligently pruning the evaluation tree, for example using methods from artificial intelligence research, such as alpha-beta pruning [9], or research in multi-hypothesis Kalman filters [1].

Though this paper only outlined the procedure for use with a Kalman filter, the method should be general enough to apply to other estimation systems, for example particle filters [7]. This is non-trivial, since this work makes use of the fact that the entropies do not depend on the actual value of the observations. This is no longer the case with more general state estimators.

Multiple camera actions have also been studied in object recognition [3] using reinforcement learning. The parallels between the reinforcement learning methods and this work will be investigated.

Zoom motor speed	horizon 1	horizon 2	horizon 3	horizon 4
3 steps	52.5	33.7	30.3	23.3
4 steps	21.2	20.4	17.1	16.1
5 steps	16.9	16.9	15.9	16.1
unlimited	15.1			
fixed	27.7			

Table 1. Mean error, in world units, for each of the 12 experiments. The last two rows show the results for an unlimited zoom motor speed, and a fixed focal length. A variable focal length approach is always superior to a fixed one, except for the special case of slow zoom motors. These cases can be caught by a multi-step lookahead.

Lastly, these methods need to be evaluated for more general cost models, based on the "size" or "distance" of an action and not just on its feasibility.

References

1. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, Boston, San Diego, New York, 1988.
2. T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons, New York, 1991.
3. F. Deinzer, J. Denzler, and H. Niemann. Viewpoint Selection – Planning Optimal Sequences of Views for Object Recognition. In *Computer Analysis of Images and Patterns – CAIP 2003*, LNCS 2756, pages 65–73, Heidelberg, August 2003.
4. J. Denzler, C.M. Brown, and H. Niemann. Optimal Camera Parameter Selection for State Estimation with Applica tions in Object Recognition. In *Mustererkennung 2001*, pages 305–312, Heidelberg, 2001.
5. J. Denzler, M. Zobel, and H. Niemann. Information Theoretic Focal Length Selection for Real-Time Active 3-D Object Tracking. In *International Conference on Computer Vision*, pages 400–407, Nice, France, 2003.
6. J. Fayman, O. Sudarsky, E. Rivlin, and M. Rudzsky. Zoom tracking and its applications. *Machine Vision and Applications*, 13(1):25–37.
7. M. Isard and A. Blake. Condensation — conditional density propagation for visual tracking. 29(1):5–28, 1998.
8. R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–44, 1960.
9. D.E. Knuth and R.W Moore. An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4):293–326, 1975.
10. Lucas Paletta and Axel Pinz. Active object recognition by view integration and reinforcement learnin g. *Robotics and Autonomous Systems*, 31, Issues 1-2:71–86, April 2000.
11. B. Tordoff and D.W. Murray. Reactive zoom control while tracking using an affine camera. In *Proc 12th British Machine Vision Conference, September 2001*, volume 1, pages 53–62, 2001.