Journal of Theoretical and Applied Information Technology
31st August 2014. Vol. 66 No.3
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEEP EXTREME TRACKER BASED ON BOOTSTRAP PARTICLE FILTER

[1] ALEXANDER A S GUNAWAN, [2] MOHAMAD IVAN FANANY, [3] WISNU JATMIKO
[1] Bina Nusantara University, Mathematics Department, School of Computer Science, Jakarta, Indonesia
[2, 3] Universitas Indonesia, Faculty of Computer Science, Depok, Indonesia
E-mail: [1] aagung@binus.edu, [2] ivan@cs.ui.ac.id, [3] wisnuj@cs.ui.ac.id

ABSTRACT

Visual tracking in mobile robots have to track various target objects in fast processing, but existing state-of-the-art methods only use specific image feature which only suitable for certain target objects. In this paper, we proposed new approach without depend on specific feature. By using deep learning, we can learn essential features of many of the objects and scenes found in the real world. Furthermore, fast visual tracking can be achieved by using Extreme Learning Machine (ELM). The developed tracking algorithm is based on bootstrap particle filter. Thus the observation model of particle filter is enhanced into two steps: offline training step and online tracking step. The offline training stage is carried out by training one kind of deep learning techniques: Stacked Denoising Autoencoder (SDAE) with auxiliary image data. During the online tracking process, an additional classification layer based on ELM is added to the encoder part of the trained. Using experiments, we found (i) the specific feature is only suitable for certain target objects (ii) the running time of the tracking algorithm can be improved by using ELM with regularization and intensity adjustment in online step, (iii) dynamic model is crucial for object tracking, especially when adjusting the diagonal covariance matrix values. Preliminary experimental results are provided. The algorithm is still restricted to track single object and will extend to track multiple object and will enhance by creating the advanced dynamic model. These are remaining for our future works.

Keywords: *visual tracking, mobile robots, bootstrap particle filter, deep learning, extreme learning machine*

1. INTRODUCTION

The key requirement for many applications in mobile robots is ability to track various target objects in video sequences in real-time. Currently, many existing tracking methods have been introduced for certain tasks. However, many of them have weakness regarding two main requirements. First, they make basic assumptions about the environment and object appearances. For example in [1] the intention is that the machines can operate indoors and outdoors, by day and night, and thus experience various weather and illumination. But it is assumed that the target object of the application is only human. Second, most visual tracking algorithms are computationally expensive. In mobile robots, on average only 10% of the computational resources of the robot are available for a tracking task and GPU parallelization is restricted, because of their high power consumption [2].

In this research, we overcome two above requirements by introducing new approach in tracking single object in video sequences. The proposed tracker does not depend on specific feature. By using deep learning, we can learn essential features of many of the objects and scenes found in the real world. Furthermore, fast visual tracking can be achieved by using Extreme Learning Machine (ELM) during online tracking step.

The developed tracking algorithm is based on bootstrap particle filter. Then the observation model of bootstrap particle filter is enhanced into two steps: offline training step and online tracking step. The offline training stage is carried out by training one kind of deep learning techniques: Stacked Denoising Autoencoder (SDAE) with auxiliary image data. During the online tracking process, an additional classification layer based on ELM is added to the encoder part of the trained. The developed tracker is called as Deep Extreme Tracker (DET).

In this paper, first we discuss the Bayesian framework and its implementation based on bootstrap particle filter. Then, we consider Deep Extreme Tracker (DET) and discuss its foundation

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

theory about Stacked Denoising Autoencoder (SDAE) and regularized ELM. Finally, we discuss the experiment results dealing with accuracy and running time.

2. TRACKING AS BAYESIAN PROBLEM

Based on Bayes' theorem, the problem of tracking moving objects can be described [3] as follows:

Prediction:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1})\ p(x_{t-1} \mid z_{1:t-1})\ dx_{t-1} \quad (1)$$

Update:

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\ \ p(x_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \quad (2)$$

with x_t is estimated hidden state in discrete time index t, and $z_{1:t} = \{z_1 \ldots z_t\}$ is a set of observations.

The integral in (1) is called as the Chapman-Kolmogorov equation. The solution of this integral gives the predicted state of the elements of x_t using the transition density $p(x_t \mid x_{t-1})$, given all the measurements up to time t−1 and the state at time t−1. Upon receipt of measurement z_t at time t, the predicted state is corrected by likelihood factor $p(z_t \mid x_t)$ and normalized using the denominator $p(z_t \mid z_{1:t-1})$ to estimate the distribution $p(x_t \mid z_{1:t})$.

3. PARTICLE FILTER FOR OBJECT TRACKING

For numerical implementation of equation (1) and (2), it is used the particle filter which approximates the posterior distribution using a finite set of weighted samples or particles. Instead of solving the optimal solution to an approximate model, the particle filter gives discrete approximation to the exact model posterior. It is preferred by many researchers to deal with the estimation problem when the system is nonlinear and non-Gaussian. The basic idea behind the particle filter is Monte Carlo simulation [4] in which the posterior density is approximated by a set of particles with associated weights. The easiest of particle filter approaches to implement is the bootstrap particle filter (BPF). In the BPF, the importance density is just simplified as transition density $p(x_t \mid x_{t-1})$. For detail explanation of tracking as Bayesian problem and its implementation using BPF can be looked in [5].

4. DEEP EXTREME TRACKER

In this research, we proposed an algorithm based on bootstrap particle filter and using deep learning and Extreme Learning Machine (ELM) to enhance the observation model. The developed tracker is called as Deep Extreme Tracker (DET), and its observation model can be divided into two steps: offline training step and online tracking step. The offline training stage is carried out by training one kind of deep learning techniques that is Stacked Denoising Autoencoder (SDAE) with auxiliary image data [6]. During the online tracking process, an additional classification layer based on ELM is added to the encoder part of the trained. More explanations are described in the next sub section

4.1 Transition Model

The location of a target object in an image frame can be represented by the six parameters of an affine transformation $\vec{x}_t = (x_t, y_t, s_t, \theta_t, \alpha_t, \phi_t)$, which denote x, y translation, scale, rotation angle, aspect ratio, and skew direction at time t [7]. Transition model which represents dynamics between states in this space is modeled by Brownian motion. Each parameter in current state of target object $\vec{x}_t$ is modeled independently by a Gaussian distribution around its past state in $\vec{x}_{t-1}$, and thus the motion between frames is itself an affine transformation. In precise equation, the transition model is:

$$p(\vec{x}_t \mid \vec{x}_{t-1}) = N(\vec{x}_t \mid \vec{x}_{t-1}, \Psi) \quad (3)$$

where Ψ is a diagonal covariance matrix whose elements are the corresponding variances of affine parameters, that is: $\Psi = (\sigma_x^2, \sigma_x^2, \sigma_s^2, \sigma_\theta^2, \sigma_\alpha^2, \sigma_\phi^2)$. These parameters establish the kind of motion of target object and this generic dynamical model assumes it does not change over time. With suitable values in the diagonal covariance matrix Ψ and more particles, it is possible to track the object with higher precision at the cost of increased computation. The values of Ψ are supposed to depend on target object dynamics.

4.2 Observation Model

For the reason that the goal of observation model is to use a representation to describe the "thing" that we are tracking, it is created effective image representation which can be learn automatically based on deep learning techniques.

The state variable $\vec{x}_t$ describes the affine motion parameters (and thus the location) of the target object at time t. Given a set of observed images $I = \{I_t^1 \ldots I_t^n\}$, the observation model is aimed to estimate the value of the hidden state variable $\vec{x}_t$. The observation model is used to measure the observation likelihood of the particles. In this research, the observation model is divided into two steps: offline training step and online tracking step and it is built based on the state of the art of machine learning.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For offline step, it is used SDAE, recent variant of deep learning technique based on the autoencoder. Deep Learning hypothesizes that in order to learn high-level representations of data a hierarchy of intermediate representations are needed. In the vision case the first level of representation could be Gabor like filters, the second level could be line and corner detectors, and higher level representations could be objects and concepts [8]. Recently, deep learning architectures have been used successfully to give very promising results for some complicated tasks, including image classification [9]. The key to success is to make use of deep architectures to learn richer invariant features via multiple nonlinear transformations. We consider that visual tracking can also benefit from deep learning for the same reasons.

The main drawback of SDAE is the needed computation time relatively long, thus it is not a problem in offline learning step, but will be barrier for implementation in online tracking step. In this research, it is used ELM in online step to increase its computation time. Furthermore, ELM can make the tracker algorithm more robust. Technically, it means the confidence threshold τ can be decreased carefully without damaging the tracking quality. In online step, the confidence of each particle is determined by making a simple forward pass through the network. The implementation of observation model can be described as:

1) Initialization

In this research, the initialization of the particles is done by putting the rectangular box on target object in first frame. This box has to be chosen carefully to represent all features of the target object by defining its horizontal and vertical center, and also its width and height. Some negative examples are collected from the background at a short distance from the target object.

Furthermore, it is also need to be adjusted the diagonal covariance matrix Ψ with suitable values based on target object dynamics. Practically, all challenges in visual tracking, e.g. partial and full occlusion, fast motion, out-of-view etc, can be overcome by choosing these values carefully.

2) Offline Training Step

For offline training, it is used the Tiny Images dataset [10] as auxiliary data. The dataset was collected from the web using seven search engines, covering many of the objects and scenes found in the real world. From the almost 80 million tiny images each of size 32×32, it is sampled 1 million images randomly and converted to grayscale for offline training. Therefore, each image is represented by a vector of 1024 dimensions corresponding to 1024 pixels. The feature value of each dimension is then linearly scaled to the range [0, 1].

For offline step, it is used SDAE, recent variant of deep learning technique based on the autoencoder. SDAE learns to recover a data sample from the corrupted version of sample. In that way, robust features are learned since the neural network contains a hidden layer with more units than the input units. The whole structure of the SDAE is depicted in Fig. 1.

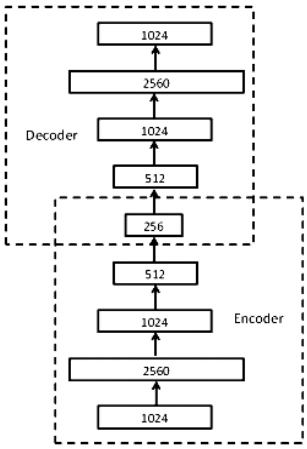


Figure 1. Stacked Denoising Autoencoder

It is observed in the research that employing SDAE can be optimized using important parameter such as dropout and sparsity target. To optimize the algorithm, we have explored these parameters.

a) Dropout

When a large feed forward neural network is trained on a small training set, it typically performs poorly on held-out test data [11]. This "over fitting" is greatly reduced by randomly omitting some of the feature detectors on each training case. This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Random "dropout" gives improvements in the training step and also accelerates its convergence.

Nevertheless, it is observed in this research that dropout level is related to the instability in object tracking. Thus improvements in the training step because of introducing the dropout level have to pay by increasing the threshold τ. It has to be done because the tracker should not forget to tune the network along the tracking process in maintaining tracking stability.

b) Sparsity target

In training step, to repair hidden units that have developed very large weights early in the training and are either always firmly on or always firmly off

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is to use a sparsity target. Also, discriminative performance is sometimes improved by using features that are only rarely active [12]. Sparse activities of the binary hidden units can be achieved by specifying a sparsity target which is the desired probability of being active, p << 1. An additional penalty term is then used to encourage the actual probability of being active, q, to be close to p. However, it is observed in this research that introducing sparsity target does not have large influences in object tracking. It might only have impacts on a long sequence of object tracking.

3) Online Tracking Step

It is clear that the learning speed of feed forward neural networks is in general far slower than required and it has been a major bottleneck in object tracking [13]. Two key reasons behind may be: (1) the slow gradient-based learning algorithms are extensively used to train neural networks, and (2) all the parameters of the networks are tuned iteratively by using such learning algorithms.

For the above reason, on the online step our tracker employ Extreme Learning Machine (ELM) for calculating the additional classification layer which added to the encoder part of the trained in offline step. In theory, this algorithm tends to provide good generalization performance at extremely fast learning speed. The experimental results based on a few artificial and real benchmark function approximation and classification problems show that ELM can produce good generalization performance in most cases and can learn much faster for learning feed forward neural networks.

The main different of our implementation comparing to other generic ELMs is in the input layer. In DET, it is not used a set of random number for input layer, but utilized the end result of offline training step. The overall network architecture is shown in Fig. 2.

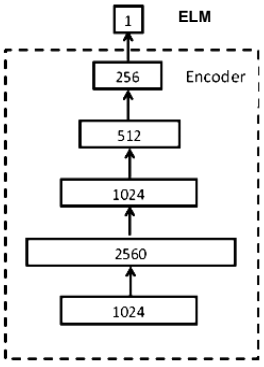


Figure 2. Network For Online Tracking

It is observed in the research that employing ELM in the additional classification layer also has a main downside especially in facing dark illumination. The problem might come from the singularity in matrix contained many zeros (that means black color). To remedy this algorithm instability, we also employ regularization in ELM and intensity adjustment in image sequences.

a) Regularization

Generic ELM tends to generate over-fitting model and its performance is affected seriously when outliers exist in the dataset. The problems can be handled using regularization techniques and hence make the computation more reliable. For the case where the number of training samples is huge, [14] gives the alternative regularization solution, that is:

$$\beta = \left(\frac{\mathrm{I}}{c} + H^T H \right)^{-1} H^T T \tag{4}$$

where β is additional classification layer, I is identity matrix, H is the end result matrix of offline training, T is target value (positive or negative value). Coefficient c is regularization factor and is set 1e6 in the research.

b) Image Processing

In the research, to enhancing grayscale images is used intensity adjustment that increases the contrast of the image by mapping the values of the input intensity image to new values such that, by default, 1% of the data is saturated at low and high intensities of the input data. Employing this image processing technique, our tracker maintains tracking robustness in all video sequences. It is observed the intensity adjustment also has influence in improving other tracker. Another alternative technique is histogram equalization that enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram (uniform distribution by default).

5. EXPERIMENTAL RESULTS

In order to evaluate the proposed method, it is done the experiments using a video camera to track the objects. The experiments are implemented on Intel i3 2.53 [GHz] CPU (without GPU) and 2 [GB] RAM. The experiments are done using 1000 particles and facing to various challenges in seven video sequences that is illumination variation, partially or fully occlusions, object deformation, fast motion, image blur and out-of-view. The used dataset are: woman [6], car4, davidin [7], person, partialocc, fullocc [15], ballocc (own generated video). For initialization, the rectangular box on target object in first frame is chosen carefully to

Journal of Theoretical and Applied Information Technology
31st August 2014. Vol. 66 No.3
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

represent all features of the target object. And the diagonal covariance matrix is adjusted with suitable values based on target object dynamics. Finally, we empirically compare DET with Incremental Visual Tracker (IVT) [7] and Deep Learning Tracker (DLT) [6] using above seven challenging video sequences. IVT, which based on principal component analysis (PCA), is regarded as representation of common visual trackers that based in specific feature.

5.1 Accuracy

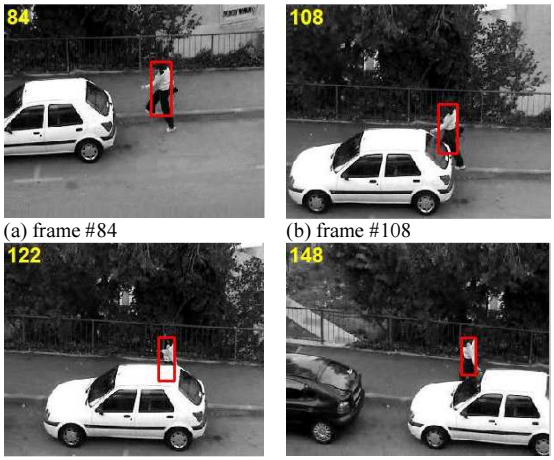
Because of all important dynamics parameters have been carefully set up, both DET and DLT can track all video sequences very well. Thus, accuracy is not main issue in both trackers. The important point is that there is the big difference of the confidence threshold value. For DLT, the threshold has to be set at 0.9 in order to track all video sequences except some last frames. On the other hand, the threshold is only 0.5 for DET and can fully track all video sequences correctly. For information, if the maximum confidence of all particles in a frame is below the threshold, the whole network will be tuned again.

Nevertheless, IVT cannot track all video sequences well. From experiment results, it can be seen IVT only suitable for certain target objects. The suitable characteristics of target object are slow motion (in partialocc) and relative permanent like face (in davidin). In order to compare the accuracy of trackers, it is listed the tracking performance of each sequence in table I.

Table I. Comparison Of Accuracy On 7 Video Sequences.

DATA SET	TRACKING PERFORMANCE			#FRAME
	DLT	IVT	DET	
WOMAN	✓	×	✓	550
CAR4	✓	×	✓	659
DAVIDIN	✓	✓	✓	770
PERSON	✓	×	✓	948
PARTIALOCC	✓	✓	✓	306
FULLOCC	✓	×	✓	454
BALLOCC	✓	×	✓	145

In Fig. 3, it is illustrated the performance of both DLT and DET in facing challenging events in some video sequences. Fig. 3 shows the performance of DET in handling partially occlusion (frame #84 – frame #148) using woman dataset. As shown in that figure, DET can overcome this partially occlusion challenge. In addition, DLT also perform well in facing this challenge.



(a) frame #84 (b) frame #108

(c) frame #122 (d) frame #148

Figure 3. Partially Occlusion In Woman Dataset

Fig. 4 shows the performance of DET in handling 3D object deformation (frame #405 – frame #500) using davidin dataset. As shown in that figure, DET can overcome this object deformation challenge. In beginning, DLT also perform well in facing this challenge, but DLT cannot track well more in davidin dataset after frame #704.

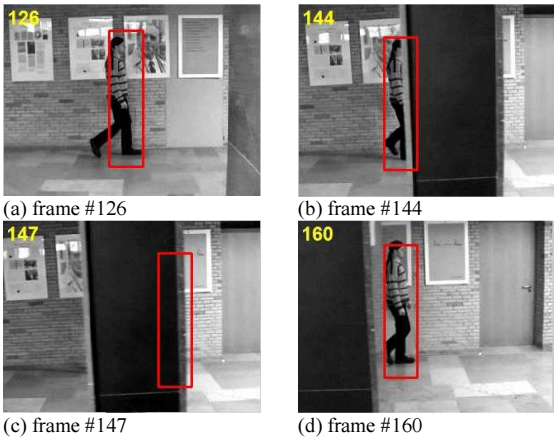
(a) frame #405 (b) frame #426

(c) frame #460 (d) frame #500

Figure 4. 3D Object Deformation In Davidin Dataset

Fig. 5 shows the performance of DET in handling fully occlusion (frame #126 – frame #160) using fullocc dataset. As shown in that figure, DET can overcome this fully occlusion challenge. In addition, initially DLT also perform well in facing this challenge, but DLT cannot track well more in fullocc dataset after frame #366.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a) frame #126 (b) frame #144

(c) frame #147 (d) frame #160

Figure 5. Fully occlusion in fullocc dataset

5.2 Running Time

In order to compare the performance time of both DLT and DET, it is listed the running time of each sequence in table II. The running time is calculated in frame per second (fps) and in addition it is also listed number of frame of each sequence.

Table Ii. Comparison Of Running Time On 7 Video Sequences.

DATA SET	FPS		#FRAME	DET BETTER ?
	DLT	DET		
WOMAN	0.77	1.49	550	✓
CAR4	1.11	1.51	659	✓
DAVIDIN	0.78	1.55	770	✓
PERSON	0.74	1.52	948	✓
PARTIALOCC	1.09	1.52	306	✓
FULLOCC	0.93	1.45	454	✓
BALLOCC	0.57	1.46	145	✓

6. CONCLUSIONS

In this paper, we have successfully enhanced the observation model of tracking algorithm using combination of deep learning and ELM. The offline training stage is carried out by training one kind of deep learning techniques: Stacked Denoising Autoencoder (SDAE) with auxiliary image data. And during the online tracking process, an additional classification layer based on Extreme Learning Machine (ELM) is added to the encoder part of the trained. There are three main conclusions: (i) the specific feature is only suitable for certain target objects (ii) the running time of the tracking algorithm can be improved by using ELM with regularization and intensity adjustment in online step, (iii) dynamic model is crucial for object tracking, especially for adjusting the diagonal covariance matrix values.

The algorithm is still restricted to track single object and will extend to track multiple object and will enhance by creating the advanced dynamic model. These are remaining for our future works.

REFERENCES:

[1] R. Mosberger and H Andreasson, "An inexpensive monocular vision system for tracking humans in industrial environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 5850-5857.

[2] A. Kolarov et al., "Vision-based hyper-real-time object tracker for robotic applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, 2012, pp. 2108-2115.

[3] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*.: Cambridge University Press, 2011.

[4] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*.: Wiley, 2012.

[5] Alexander A S Gunawan and Ito Wasito, "Nonretinotopic Particle Filter for Visual Tracking," *Journal of Theoretical and Applied Information Technology*, vol. 63, no. 1, pp. 104-111, 2014.

[6] Naiyan Wang and Dit-Yan Yeung, "Learning a Deep Compact Image Representation for Visual Tracking," in *NIPS*, Lake Tahoe, Nevada, USA, 2013, pp. Proceedings of Twenty-Seventh Annual Conference on Neural Information Processing Systems.

[7] David Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental Learning for Robust Visual Tracking," *the International Journal of Computer Vision, Special Issue: Learning for Vision*, 2007.

[8] Rasmus Berg Palm, "Prediction as a candidate for learning deep hierarchical models of data," Informatics and Mathematical Modelling, Technical University of Denmark, Kongens Lyngby, Master Thesis 2012.

[9] A. Krizhevsky, I. Sutskeve, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[10] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: A large data set for nonparametric objectand scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[11] G. E Hinton, N Srivastava, A Krizhevsky, I Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

[12] G. Hinton, "A practical guide to training restricted Boltzmann machines," *In Neural Networks: Tricks of the Trade*, pp. 599–619, 2012.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing*, vol. 70, pp. 489-501, 2006.

[14] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012.

[15] Dominik A. Klein. (2010) BoBoT - Bonn Benchmark on Tracking. [Online]. http://www.iai.uni-bonn.de/~kleind/tracking/