Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 3276103, 10 pages
https://doi.org/10.1155/2017/3276103

Research Article

Robust Visual Tracking Using the Bidirectional Scale Estimation

An Zhiyong,[1,2] Guan Hao,[1] and Li Jinjiang[2]

[1]*Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 201203, China*
[2]*Key Laboratory of Intelligent Information Processing in Universities of Shandong, Shandong Institute of Business and Technology, Yantai 264005, China*

Correspondence should be addressed to An Zhiyong; azytyut@163.com

Received 21 August 2016; Accepted 18 December 2016; Published 19 January 2017

Academic Editor: Francisco Valero

Copyright © 2017 An Zhiyong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Object tracking with robust scale estimation is a challenging task in computer vision. This paper presents a novel tracking algorithm that learns the translation and scale filters with a complementary scheme. The translation filter is constructed using the ridge regression and multidimensional features. A robust scale filter is constructed by the bidirectional scale estimation, including the forward scale and backward scale. Firstly, we learn the scale filter using the forward tracking information. Then the forward scale and backward scale can be estimated using the respective scale filter. Secondly, a conservative strategy is adopted to compromise the forward and backward scales. Finally, the scale filter is updated based on the final scale estimation. It is effective to update scale filter since the stable scale estimation can improve the performance of scale filter. To reveal the effectiveness of our tracker, experiments are performed on 32 sequences with significant scale variation and on the benchmark dataset with 50 challenging videos. Our results show that the proposed tracker outperforms several state-of-the-art trackers in terms of robustness and accuracy.

1. Introduction

Visual tracking has drawn significant attentions in computer vision with various applications such as activity analysis, video surveillance, and auto control systems. Despite significant progress in recent years, it is still difficult due to baffling factors in complicated situations such as scale variations, partial occlusion, background clutter, deformation, and fast motion.

In recent years, many object tracking algorithms have been proposed. Among those trackers, the tracking-by-detection algorithms have achieved excellent performance by learning a discriminative classifier. Bolme et al. [1] presented a minimum output sum of squared error (MOSSE) filter on the gray images when initialized using a single frame. The MOSSE tracker is robust to variations in lighting, pose, and nonrigid deformations while running with a speed reaching several hundred frames per second. The MOSSE tracker can be performed efficiently because of using the fast Fourier transform (FFT). Recently, many new algorithms [2–5] based on the MOSSE filters have been proposed in researches in field of object tracking.

Unlike tracking-by-detection algorithm, the TLD tracking framework [6] was presented that decomposed the tracking task into three subtasks: tracking, learning, and detection. The TLD tracker used a P-N learning algorithm to improve classification performance in long-term tracking. In recent years, sparse representation has been successfully applied to visual tracking. Zhang et al. [7] proposed the multitask tracking framework based on the particle filters and the multitask sparse representation. Wang et al. [8] proposed the WLCS tracker based on the particle filters and weighted local cosine similarity which measures the similarities between the target template and candidates. Guo et al. [9] proposed the max confidence boosting tracker that allows ambiguity in the tracking and effectively alleviates the drift problem. Zhang and Song [10] proposed the weighted multiple instance learning tracker that considers the sample importance into the tracker framework. Wang et al. [11] proposed the structure constrained grouping based on the

Bayesian inference framework which casts visual tracking as foreground superpixels grouping problem. However, those algorithms cannot handle the scale changes well in object tracking.

In this paper, we decomposed the tracking task into two subtasks: translation filter and scale filter with a complementary scheme. The translation filter relies on a temporal context regression model. As for scale filter, our key idea is using both the forward and backward tracking information to construct the scale filter. The information from the last frame to the current frame can be defined as the forward tracking information, while the information from the current frame to the last frame can be regarded as the backward tracking information. Therefore, the scale filter can be estimated by the bidirectional scale estimation, including the forward scale and the backward scale. The proposed approach achieves state-of-the-art performance on both scale variation dataset and the benchmark dataset.

2. Related Work

Our tracking algorithm is based on the correlation filter model. The classic correlation filter algorithm is the MOSSE tracker. The MOSSE tracker takes randomly affine-transformed ground truths as training set when initializing its correlation filter. However, in MOSSE tracker, the correlation filter is only used to detect the position of target.

Based on the correlation filter, Henriques et al. [2] proposed the CSK tracker algorithm that used the circulant matrices theory to learn a kernelized least-squares classifier. The CSK tracker runs at hundreds of frames per second. Furthermore, Henriques et al. [3] proposed the kernelized correlation filter (KCF) using the multichannel HOG features based on the CSK tracker. However, the CSK and KCF trackers imply the low accuracy in the scale variations sequences because the algorithms are limited to target translation. Li and Zhu [12] proposed the SAMF tracker that extends KCF to handle scale changes by sampling with several predefined scale perturbations. In fact, by sampling predefined scale variations, the SAMF tracker is not flexible enough to deal with fast and abrupt scale changes.

Zhang et al. [4] proposed a kernelized correlation filter to predict the target scale variation that often produces the inaccurate scale estimation in a complicated environment. Danelljan et al. [5] proposed the discriminative scale space tracker (DSST) that learnt the translation filter and scale filter for the tracking target. The scale filter in DSST only uses the forward tracking information to search the reasonable scale while neglecting the backward information. This makes the scale unstable usually in complex scenes.

3. Our Approach

Here we describe our approach. Section 3.1 presents translation tracking filter. In Section 3.2 we describe scale tracking filter using the bidirectional scale estimation.

3.1. Translation Tracking Filter. The MOSSE tracker only uses the image intensity features to design the translation filter. Since the HOG feature is a strong local analysis feature, it has been widely used in visual tracking. So we use both the HOG feature and intensity feature to build the translation filter. Here we build the translation filter using the ridge regression similar to the DSST and MOSSE. Given n features $f = \{f_1, f_2, \ldots, f_i, f_n\}$, the translation tracking filter h^i is obtained by minimizing the sum of squared errors with the regularization term

$$\min \left\| \sum_{i=1}^{n} f_i * h^i - g \right\|^2 + \lambda \sum_{i=1}^{n} \left\| h^i \right\|^2, \tag{1}$$

where g is the desired correlation outputs. Here g can be designed as the 2D Gaussian shaped peak centered on the target in training image. In this paper, the upper case variables F_i, G and the filter H^i denote the Fourier transform of their lower case counterparts separately. The solution to (1) in the Fourier transform domain is [5]

$$H^i = \frac{\overline{G} \odot F_i}{\sum_{i=1}^{n} F_i \odot \overline{F_i} + \lambda}, \tag{2}$$

where $\odot$ is the element-wise product and the bar $\overline{G}$ represents complex conjugation. To obtain a robust approximation, the translation filter H_t^{trans} can be updated by the numerator $H_{t,1}^i$ and denominator $H_{t,2}$.

$$H_{t,1}^i = \eta H_{t-1,1}^i + (1-\eta)\, \overline{G} \odot F_i \tag{3}$$

$$H_{t,2} = \eta H_{t-1,2} + (1-\eta) \sum_{i=1}^{n} F_i \odot \overline{F_i}, \tag{4}$$

where t is the index of frame and η is the learning rate. Given the features F^Z of an image patch Z in the new frame, the confidence score can be calculated as

$$y = \frac{\sum_{i=1}^{n} \overline{H_{t,1}^i} \odot F^Z}{H_{t,2} + \lambda}. \tag{5}$$

So the new target location can be estimated by searching for the location of the maximum confidence score.

3.2. Scale Tracking Filter. In this section, a 1-dimensional correlation filter is constructed to estimate the target scale. The purpose of scale tracking filter is to locate the appropriate scale for the target in the current frame. Let $W \times R$ be the target size and N is the number of scales $S = \{a^n \mid n \in ([-(N-1)/2], \ldots, [(N-1)/2])\}$, where a denotes the scale factor. For each $a^n \in S$, we extract an image patch of size $a^n W \times a^n R$ centered around the estimated location. At the same time, we use the HOG features to construct the scale pyramid.

Here the training features at each scale level are set to a k-dimensional feature descriptor. Given the training samples $f^S = \{f_1^S, \ldots, f_j^S, \ldots, f_k^S\}$, the scale filter h^j is obtained by minimizing the sum of squared errors with the regularization term

$$\min \left\| \sum_{j=1}^{k} f_j^S * h^j - g \right\|^2 + \lambda \sum \left\| h^j \right\|^2, \tag{6}$$

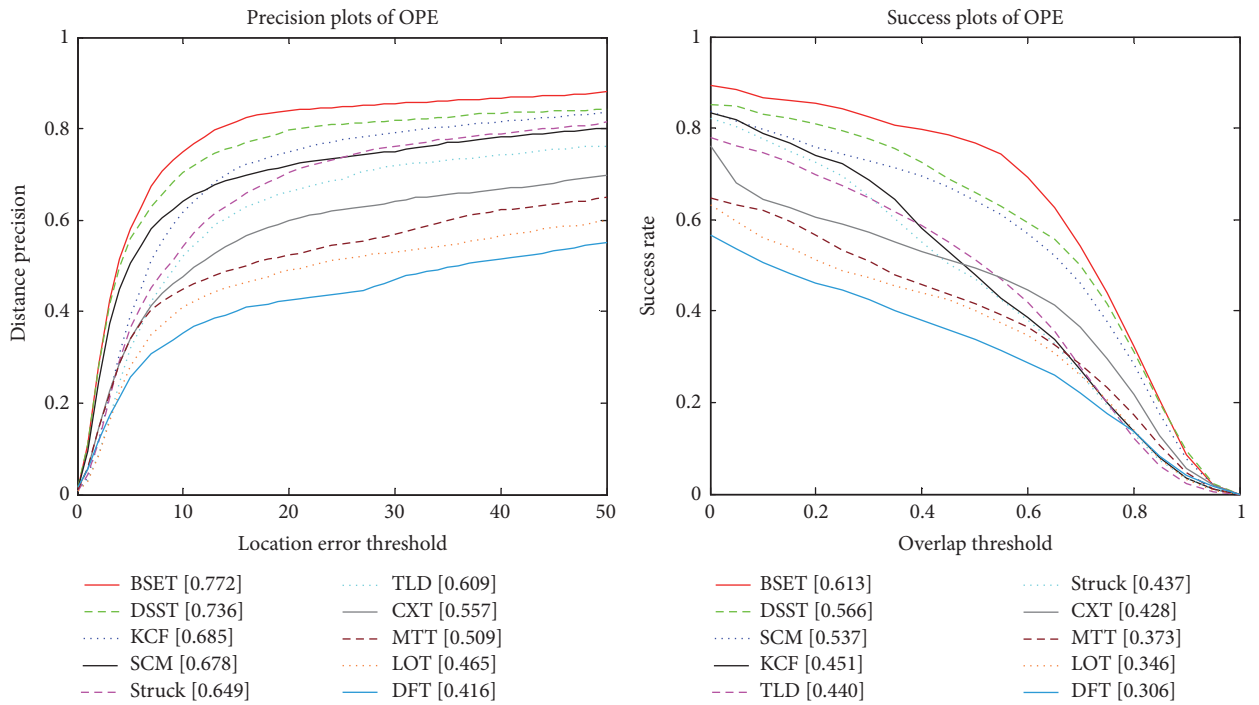


Figure 1: Precision and success plots over all the 32 sequences.

where g is as the 1-dimensional Gaussian shaped peak centered around the current scale. Then the solution to (6) in the Fourier transform domain is

$$H^j = \frac{\overline{G} \odot F_j^S}{\sum_{j=1}^k F_j^S \odot \overline{F_j^S} + \lambda}, \tag{7}$$

where the upper case variables $F_j^S, \overline{G}$ and the filter H^j denotes the Fourier transform of their lower case counterparts separately. Assume the scale filter H_{t-1}^{scale} in the last frame be expressed by the numerator A_{t-1}^j and denominator B_{t-1}. Given the features F^t of an image patch in the current frame, the maximum scale confidence score can be calculated as

$$y_t^F = \max\left(\frac{\sum_{j=1}^k \overline{A_{t-1}^j} \odot F^t}{B_{t-1} + \lambda}\right), \tag{8}$$

where t is the index of frame. So the new target scale can be estimated by searching for the scale of the maximum confidence score. Let $S^F = c \times S_{t-1}$ denote the corresponding scale to y_t^F. Here S_{t-1} denotes the target scale in the last frame and c denotes the relevant factor between the new target scale in the current frame and the target scale in the last frame. Then the scale S^F can be set to the forward scale for the new target since it uses the last frame's filter H_{t-1}^{scale} to estimate the target scale in the current frame.

In order to improve the performance, we compute a backward scale for the same target. Based on the new target location and the forward scale S^F, we can obtain a new scale filter H_{s^F} that includes the numerator A^j and denominator B. It is noted that the scale filter H_{s^F} only uses the new target information in the current frame. Given the features F^{t-1} of an image patch in the last frame, the max confidence score can be calculated as

$$y_{t-1}^I = \max\left(\frac{\sum_{j=1}^k \overline{A^j} \odot F^{t-1}}{B + \lambda}\right). \tag{9}$$

Let $S_{t-1}^I = d \times S^F$ denote the corresponding scale to y_{t-1}^I in the last frame and let d denote the backward relevant factor. Then the new target scale can be expressed by $S^I = S_{t-1}^I / d$. The scale S^I can be set to the backward scale estimation since it uses the current frame's filter H_{s^F} to estimate scale.

In most cases, the backward scale S^I is equal to forward scale S^F. However, there are always estimating errors for the backward and forward scales in practical situation. Here the trend of scale change can be defined as two types: increase and decrease. When the scale value is more than 1, the scale trend is regarded as the increase. While the scale value is less than 1, we can assume that the scale trend is the decrease. The forward scale's trend is often inconsistent with the backward scale's trend in some situations. The main reason is that the scale filter H_{t-1}^{scale} includes the history information while the scale filter H_{s^F} only uses the current scale information. In fact, the scale filter H_{t-1}^{scale} with the history information can produce inaccurate trend estimates for scale. Thus, we should set the backward scale to the final scale when the forward scale's trend is inconsistent with the backward scale's trend.

Figure 2: A visualization of the tracking results of five visual trackers on challenging sequences: car24, carscale, human5, RedTeam, and skating1.

So a conservative strategy is adopted to integrate two above scales

$$S_t = \begin{cases} S^I, & \text{if } S^F < 1 \,\&\&\, S^I \geq 1 \\ S^I, & \text{if } S^F \geq 1 \,\&\&\, S^I < 1 \\ S^F, & \text{else.} \end{cases} \tag{10}$$

When the forward scale's trend is identical to the backward scale's trend, we can set the forward scale to the final scale since the filter H_{t-1}^{scale} includes abundant scale information. So the scale filter can be updated by the numerator A_t^j and denominator B_t

$$A_t^j = \eta A_{t-1}^j + (1-\eta) F_j^t \odot \overline{G} \tag{11}$$

$$B_t = \eta B_{t-1} + (1-\eta) \sum_{j=1}^{k} F_j^t \odot \overline{F_j^t}, \tag{12}$$

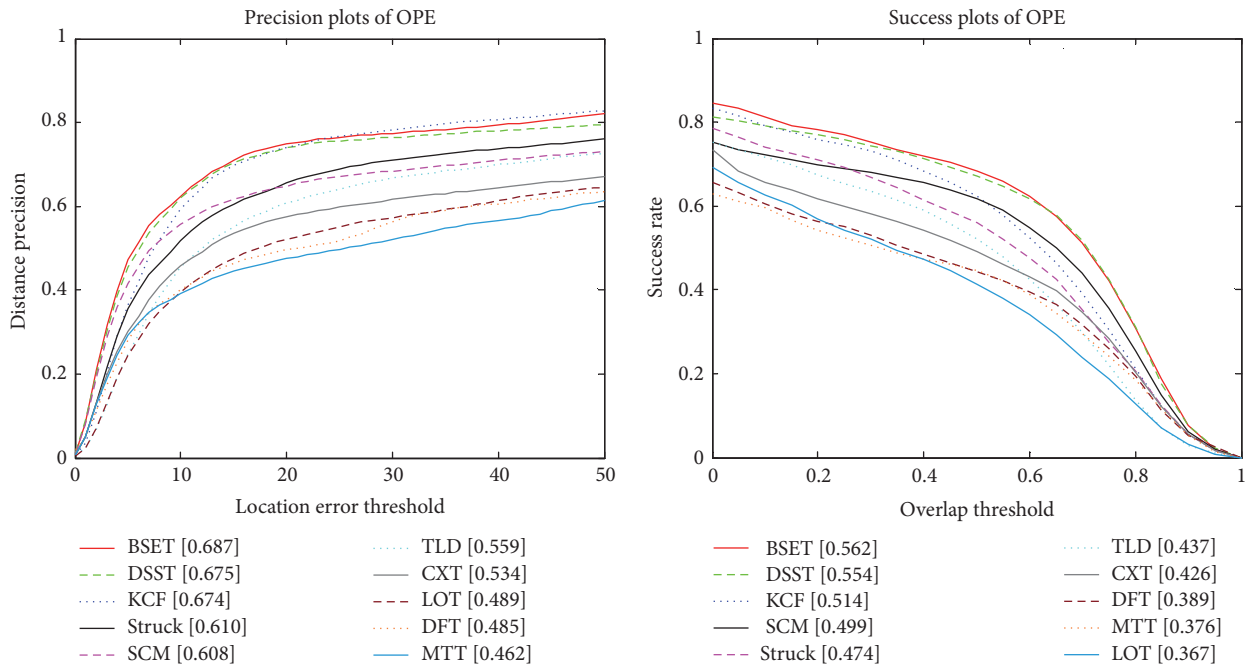


Figure 3: Evaluation on OTB-50 dataset.

where η is the learning rate. It is noted that the features F_j^t in (11) and (12) are based on the final scale S_t. It is effective to update the scale filter since the accurate scale S_t can improve the performance of scale filter to a great extent. We present an outline of our method in Algorithm 1.

Algorithm 1 (proposed tracking algorithm).

Input.

Image I_t

Previous target position and scale $(x_{t-1}, y_{t-1}, S_{t-1})$.

Translation model H_{t-1}^{trans}: the numerator $H_{t-1,1}^i$ and denominator $H_{t-1,2}$.

Scale model H_{t-1}^{scale}: the numerator A_{t-1}^j and denominator B_{t-1}.

Output.

Estimated target position and scale (x_t, y_t, S_t).

Updated translation model $H_t^{\text{trans}}(H_{t,1}^i, H_{t,2})$ and scale model $H_t^{\text{scale}}(A_t^j, B_t)$.

Translation Estimation.

(1) Extract the target features at the previous target position (x_{t-1}, y_{t-1}) and scale S_{t-1} from I_t.

(2) Compute the translation filter H^{trans} using (2).

(3) Set (x_t, y_t) to the target position that maximizes the confidence score in (5).

Scale Estimation.

(4) Extract the target features around from I_t at target position (x_t, y_t) and scale S_{t-1}.

(5) Compute the forward scale S^F using the scale filter H_{t-1}^{scale}.

(6) Compute the scale filter H_{s^F} and the backward scale S_t^I.

(7) Integrate the forward scale S_t^F and the backward scale S_t^I as the final scale S_t using (10).

Model Update.

(8) Extract the target features from I_t at the target position (x_t, y_t) and scale S_t.

(9) Update the translation model H_t^{trans}: updated by the numerator $H_{t,1}^i$ in (3) and denominator $H_{t,2}$ in (4).

(10) Update the scale model H_t^{scale}: updated by the numerator A_t^j in (11) and denominator B_t in (12).

4. Experiments

We name our proposed tracker "BSET" *(Bidirectional Scale Estimation Tracker)*. All trackers in this paper are implemented in Matlab 2013 on an Intel I5-3210 2.50 GHz CPU with 4 GB RAM. The regularization parameter λ is set to 0.01 and the learning rate η is set to 0.025 in formulas. The number of scales is set to 33 with a self-adaptive scale factor. Given a target of size $M \times N$, the self-adaptive scale factor can be set to

$$a = \begin{cases} 1.04, & \text{if } \min(M, N) \le 20 \\ 1.02, & \text{if } \max(M, N) \ge 100 \\ 1.03, & \text{else.} \end{cases} \quad (13)$$

Table 1: Per-video overlap precision (OP) (%) on the 32 sequences. "∗" indicates the best performance, "∗∗" indicates the second best ones, and "#" indicates the third best ones.

	BSET	DSST	KCF	Struck	SCM	TLD	MTT	LOT	DFT	CXT
biker	**47.2***	26.8	25.4	25.4	**44.4**$^{\#}$	31.0	31.0	43.0	25.4	**46.5****
blurCar2	**100.0***	**100.0***	94.7	93.8	14.4	**99.8****	13.3	28.7	17.4	**95.0**$^{\#}$
boy	**100.0***	**100.0***	**99.2****	**97.5**$^{\#}$	43.9	93.5	48.7	65.0	48.3	49.7
car24	**100.0***	17.3	17.3	17.0	**100.0***	**48.3****	**100.0***	**37.2**$^{\#}$	7.2	**100.0***
car4	**100.0***	**100.0***	36.4	39.8	**97.3****	**79.2**$^{\#}$	31.1	4.9	25.8	29.9
carScale	**88.5***	**84.5****	44.4	43.3	65.1	43.7	56.7	46.4	44.8	**78.2**$^{\#}$
couple	10.7	10.7	24.3	54.3	10.7	**100.0***	**61.4****	**56.4**$^{\#}$	8.6	**56.4**$^{\#}$
crossing	**100.0***	**100.0***	**95.0****	**94.2**$^{\#}$	**100.0***	51.7	22.5	30.8	64.2	34.2
dog1	**100.0***	**100.0***	65.1	65.3	84.7	67.3	78.5	**98.6**$^{\#}$	52.1	**99.8****
doll	**99.7***	**99.7***	55.2	68.8	**98.7****	62.4	50.4	87.1	35.0	**97.5**$^{\#}$
dudek	**99.0***	**98.1****	97.6	**98.0**$^{\#}$	97.6	84.2	92.9	61.8	80.1	92.4
fleetface	**70.0****	66.5	**66.9**$^{\#}$	66.6	**70.6***	56.7	54.6	57.9	55.6	64.6
freeman1	**77.3****	35.3	16.3	21.5	**80.7***	21.2	16.0	18.7	17.8	**25.8**$^{\#}$
freeman3	**60.0**$^{\#}$	31.3	27.8	20.0	**93.0****	58.0	47.8	7.4	33.0	**93.7***
freeman4	16.6	**41.7***	18.4	15.5	**24.4**$^{\#}$	**26.9****	18.0	13.4	18.0	18.0
girl	87.4	30.6	74.2	**98.0***	**88.2**$^{\#}$	76.4	**93.2****	54.6	25.2	64.2
Human5	**99.7***	24.3	23.6	34.1	41.5	**51.2****	35.1	**42.5**$^{\#}$	7.6	28.1
Human8	**100.0***	**100.0***	**30.5**$^{\#}$	13.3	**100.0***	13.3	11.7	**92.2****	13.3	10.9
ironman	**34.9***	**13.3**$^{\#}$	**15.1****	4.8	**13.3**$^{\#}$	6.6	8.4	8.4	3.6	3.0
liquor	40.9	40.9	**98.1***	40.6	32.1	**58.2**$^{\#}$	20.1	**96.3****	22.9	21.0
matrix	**18.0**$^{\#}$	**18.0**$^{\#}$	13.0	12.0	**30.0***	7.0	**29.0****	4.0	6.0	4.0
RedTeam	**96.2***	**70.0****	37.6	39.8	39.1	28.3	**51.4**$^{\#}$	10.1	27.0	45.3
shaking	**100.0***	**100.0***	1.4	16.7	**89.6****	40.0	1.1	7.7	**82.5**$^{\#}$	10.7
singer1	27.6	**100.0***	27.6	29.9	**100.0***	**99.1****	**34.5**$^{\#}$	24.2	27.6	32.2
skating1	**98.0***	**52.3****	36.3	37.0	**42.3**$^{\#}$	22.8	13.5	28.0	16.3	12.0
soccer	14.5	**38.8****	**39.3***	15.6	**23.7**$^{\#}$	12.2	18.1	21.7	21.9	12.8
surfer	**89.4**$^{\#}$	29.0	39.9	15.7	40.7	**89.9****	5.6	37.2	3.7	**96.3***
toy	**97.0****	**90.0**$^{\#}$	43.2	49.1	22.1	73.4	53.1	21.4	46.9	**97.4***
trellis	**96.8****	**97.7***	84.0	78.4	**85.4**$^{\#}$	47.3	19.3	31.6	51.8	80.8
walking	**99.8***	**99.8***	51.5	56.6	**95.9**$^{\#}$	38.3	**95.9**$^{\#}$	**96.8****	55.1	21.8
walking2	**100.0***	**100.0***	38.0	**43.4**$^{\#}$	**100.0***	34.0	**99.2****	39.0	38.2	39.8
woman	88.3	**93.3**$^{\#}$	**93.6***	**93.5****	85.8	16.6	19.8	8.4	**93.5****	20.6
Average OP	**76.8***	**65.9****	47.8	46.8	**64.2**$^{\#}$	51.2	41.6	40.0	33.6	49.5

The strategy can adjust the scale factor with the size of target adaptively and overcomes the problem of scale slow increase for the small target. When the size of target is large, we should set a small value to the scale factor since this can adapt the scale change responsively. To assess the overall performance of BSET, a large benchmark dataset (OTB-50) [13] is adopted that contains 50 videos with many challenging attributes such as scale variation, in-plane rotation, low resolution, and background clutter.

We adopt the 32 sequences annotated with "scale variation" as the scale variation dataset. Two criteria, the center location error (CLE) as well as the overlap precision, are employed to the scale dataset in our paper. The CLE can be defined as the average Euclidean distance between the ground truth and the estimated center location of the target. Overlap precision (OP) [5] is defined as the percentage of frames where the bounding box overlap surpasses a threshold $t \in [0, 1]$. We compare our method with the nine state-of-the-art trackers: DSST [5], KCF [3], Struck [14], SCM [15], TLD [6], MTT [7], LOT [16], DFT [17], and CXT [18], which have been shown to provide excellent performance in literatures. In addition, we provide two kinds of plots: precision plot and success plot [13] to evaluate all trackers, where trackers are ranked using the area under curve (AUC).

4.1. Experiment 1: Robust Scale Estimation. The scale variation dataset includes the 32 sequences and those sequences

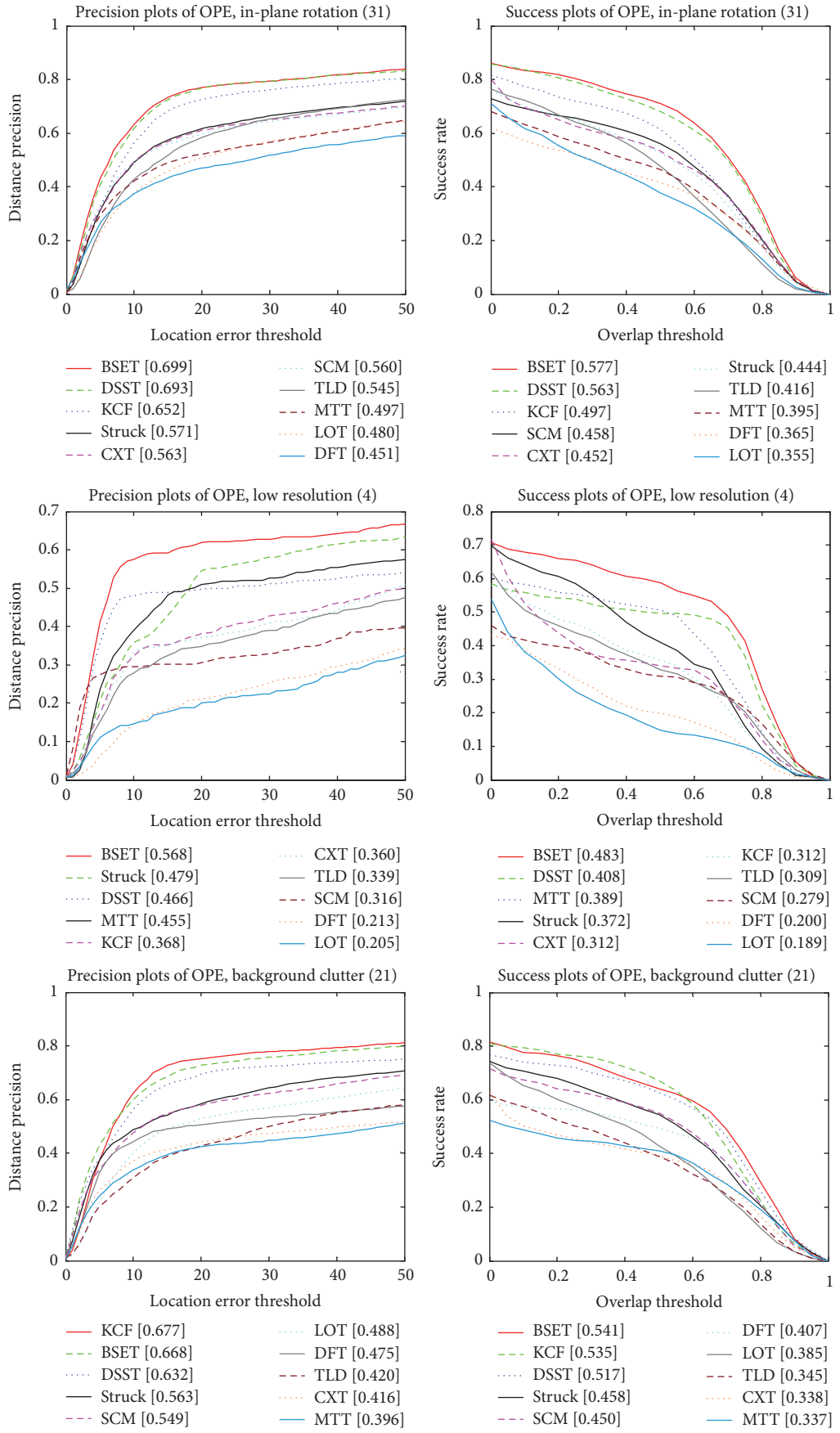


Figure 4: Evaluation on three tracking challenges of in-plane rotation, low resolution, and background clutter.

Table 2: Per-video CLE (in pixel) on the 32 sequences. "∗" indicates the best performance, "∗∗" indicates the second best ones, and "#" indicates the third best ones.

	BSET	DSST	KCF	Struck	SCM	TLD	MTT	LOT	DFT	CXT
biker	**35.4**$^{\#}$	68.9	71.2	**25.0****	72.6	66.4	76.8	69.3	74.7	**20.0***
blurCar2	**2.8***	**2.9****	6.8	10.4	125.6	**6.5**$^{\#}$	141.8	53.6	152.4	7.2
boy	**1.9***	**2.1****	**2.9**$^{\#}$	3.8	51.0	4.5	12.8	66.0	106.3	7.4
car24	**1.7****	**1.6***	4.1	119.7	**1.9****	3.0	2.6	100.0	165.6	3.5
car4	**1.7***	**1.9****	9.9	8.7	**4.3****	12.8	22.3	167.3	61.9	58.1
carScale	**18.5****	**19.1**$^{\#}$	**16.1***	36.4	33.4	22.6	87.6	101.2	75.8	24.5
couple	129.9	125.1	47.6	**11.3****	109.6	**2.5***	**27.8**$^{\#}$	37.1	108.6	41.8
crossing	**1.3***	**1.5****	2.2	2.8	**1.6**$^{\#}$	24.3	57.1	36.7	22.3	23.4
dog1	**3.9***	**4.3**$^{\#}$	**4.2****	5.7	7.0	**4.2****	**4.3**$^{\#}$	4.6	41.2	4.9
doll	**2.4***	**3.0****	8.4	8.9	**3.4**$^{\#}$	6.0	110.3	6.3	59.5	4.7
dudek	13.4	13.3	**11.4****	**11.4****	**10.8***	18.1	14.1	85.1	18.7	**12.8**$^{\#}$
fleetface	**27.6**$^{\#}$	33.6	**26.4****	**23.0***	**27.6**$^{\#}$	41.2	69.1	33.7	68.0	45.1
freeman1	**9.1**$^{\#}$	**7.6****	94.9	14.3	**6.9***	39.7	117.8	86.9	10.4	26.8
freeman3	**5.3**$^{\#}$	16.6	19.3	16.8	**3.2***	29.3	15.6	40.5	32.6	**3.6****
freeman4	57.4	**20.0***	**27.1**$^{\#}$	48.7	37.7	39.2	**23.5****	38.6	57.5	65.6
girl	**5.0**$^{\#}$	11.0	11.9	**2.6***	**2.6***	9.8	**4.3****	22.8	24.0	11.0
human5	**5.0***	302.5	175.5	**6.9**$^{\#}$	9.3	**5.3****	8.3	90.3	259.1	200.9
human8	**2.2****	**2.4**$^{\#}$	3.8	63.8	**1.9***	66.0	76.0	4.0	73.8	67.3
ironman	**53.8***	205.9	194.9	127.7	163.5	**93.2****	215.3	**98.7**$^{\#}$	239.7	162.7
liquor	98.4	98.5	**5.3***	91.0	99.2	**37.6**$^{\#}$	543.4	**8.5****	221.1	131.8
matrix	83.0	**70.2**$^{\#}$	76.4	194.5	**48.2***	**57.2****	75.4	73.5	105.8	151.6
RedTeam	**2.5***	**2.8**$^{\#}$	3.8	4.3	4.1	35.5	**2.7****	71.9	50.3	16.8
shaking	**7.6***	**8.2****	112.5	30.7	**11.0**$^{\#}$	37.1	97.9	82.6	26.3	129.2
singer1	8.5	**3.5****	12.8	14.5	**2.7***	**8.0**$^{\#}$	36.2	141.4	18.8	11.4
skating1	**6.9***	**8.4**$^{\#}$	**7.7****	82.9	16.4	145.5	293.3	110.5	174.2	129.8
soccer	**13.0***	**20.3**$^{\#}$	**15.4****	71.4	77.6	77.1	84.2	42.2	139.5	89.2
surfer	**4.0**$^{\#}$	20.1	8.7	9.0	14.8	**3.8****	37.0	26.1	150.9	**3.1***
toy	8.4	**7.9**$^{\#}$	**7.8****	11.4	50.8	8.9	12.8	53.0	31.9	**6.1***
trellis	**2.7****	**2.6***	7.8	**6.9**$^{\#}$	7.0	31.1	68.8	47.7	44.9	7.0
walking	**2.5**$^{\#}$	**1.7***	4.0	4.6	**2.5**$^{\#}$	10.2	3.5	**2.4****	5.9	205.7
walking2	**3.0****	**3.2**$^{\#}$	29.0	11.2	**1.7***	44.6	4.0	64.9	29.1	34.7
woman	10.6	9.7	10.1	**4.2***	**7.9****	139.9	137.3	117.1	**8.5**$^{\#}$	72.5
Average CLE	**19.7***	34.4	**32.5**$^{\#}$	33.9	**31.8****	35.3	77.6	62.0	83.1	55.6

also have challenging problems such as illumination variation, motion blur, background clutter, and occlusion. Table 1 shows the Per-video overlap precision at a threshold 0.5 compared with 9 state-of-the-art trackers. From Table 1, we find that the BSET algorithm provides better or similar performance on 18 out of the 32 sequences. The BSET algorithm performs well with an average OP of 0.768, which outperforms the DSST algorithm by 10.9%. Table 2 shows Per-video CLE compared with 9 state-of-the-art trackers. From Table 2, we find that the BSET algorithm provides better or similar performance on 12 out of the 32 sequences. The SCM tracker performs well with average CLE of 31.8 pixels. But the BSET tracker achieves lower average CLE of 19.7 pixels.

In addition, we report precision plots and success plots of OPE in Figure 1. In precision and success plots, our approach tracks scale more accurately on the scale variation dataset than other state-of-the-art trackers. This also indicates that using bidirectional scale estimation can improve the accuracy of scale estimation. In the precision plot, the precision score of the BSET algorithm is 0.772, which outperforms the DSST algorithm by 3.6%. Meanwhile, in the success plot, the proposed BSET algorithm achieves the AUC of 0.613, which outperforms the DSST algorithm by 4.7%. Figure 2 shows a visualization of the tracking results of our approach and the state-of-the-art visual trackers DSST [5], KCF [3], Struck [10], and TLD [6] on five challenging sequences: car24,

carscale, human5, RedTeam, and skating1. Our algorithm provides promising results compared to existing trackers on these sequences. Our algorithm performs at 13.7 frames per second on scale variation dataset that basically meets the requirement of engineering applications.

4.2. Experiment 2: Evaluation on Benchmark Dataset. Here we evaluate all the trackers on the OTB-50 dataset that includes 50 sequences (including 51 tracking targets). Resultant precision plots and success plots are shown in Figure 3, which shows our tracker is superior comparing to state-of-the-art trackers on the benchmark dataset. Among the trackers, the DSST method achieves the precision score of 0.675 and success score of 0.554 while the BSET algorithm achieves the precision score of 0.687 and success score of 0.562.

In addition, we report results for three challenging attributes in Figure 4. In the precision plot, the proposed BSET algorithm outperforms the DSST and KCF algorithms in in-plane rotation and low resolution. Meanwhile, in background clutter, the precision score of the BSET algorithm is 0.668 which is close to the KCF 0.677. In the success plot, the DSST method performs with overall success in in-plane rotation (0.563) and low resolution (0.408) while the BSET algorithm achieves success rate of 0.577 and 0.483, respectively. The KCF method performs well with overall success in background clutter (0.535) while the BSET algorithm achieves the success rate of 0.541. However, the BSET algorithm cannot handle well the fully occluded (e.g., freeman4). The main reason is that the accuracy of the backward scale will be poor in the fully occluded. In summary, the BSET algorithm adapts to scale variation, in-plane rotation, low resolution, and background clutter.

5. Conclusion

In this paper, we propose an effective tracking algorithm using the bidirectional scale estimation in a tracking-by-detection framework. We decomposed the tracking task into two subtasks: translation filter and scale filter to estimate the target objects. The translation filter is constructed using the ridge regression and multidimensional features. The scale filter is constructed by the bidirectional scale estimation, including the forward and the backward scales. Experimental results show that our proposed tracker performs favorably against several state-of-the-art trackers on the scale variation dataset.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors are grateful to the support by National Natural Science Foundation of China (61373079, 61472227) and National Natural Science Foundation of Shandong Province (ZR2013FL018, ZR2015FL019, and ZR2014FL007).

References

[1] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2544–2550, San Francisco, Calif, USA, June 2010.

[2] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, pp. 702–715, Florence, Italy, October 2012.

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[4] L. Zhang, Y. Wang, H. Sun, Z. Yao, and S. He, "Robust visual correlation tracking," *Mathematical Problems in Engineering*, vol. 2015, Article ID 238971, 13 pages, 2015.

[5] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the 25th British Machine Vision Conference (BMVC '14)*, pp. 1–11, Linköping, Sweden, September 2014.

[6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[7] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 367–383, 2013.

[8] D. Wang, H. Lu, and C. Bo, "Visual tracking via weighted local cosine similarity," *IEEE Transaction on Systems, Man, and Cybernetics Part B*, vol. 45, no. 9, pp. 1838–1850, 2015.

[9] W. Guo, L. Cao, T. X. Han, S. Yan, and C. Xu, "Max-confidence boosting with uncertainty for visual tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1650–1659, 2015.

[10] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.

[11] L. Wang, H. Lu, and D. Wang, "Visual tracking via structure constrained grouping," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 794–798, 2015.

[12] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV 2014 Workshops*, vol. 8926 of *Lecture Notes in Computer Science*, pp. 254–265, Springer, 2015.

[13] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[14] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, Barcelona, Spain, November 2011.

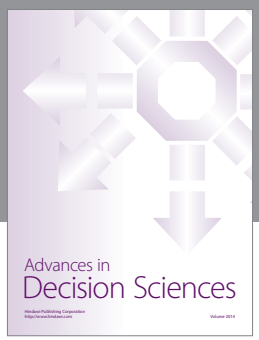
[15] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1838–1845, Providence, RI, USA, June 2012.

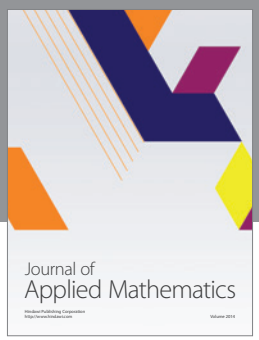
[16] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of the IEEE Conference on Computer*

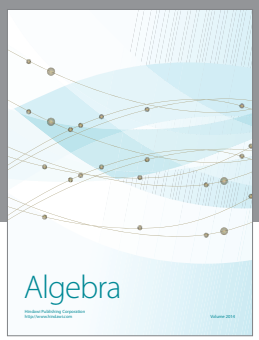
Vision and Pattern Recognition (CVPR '12), pp. 1940–1947, June 2012.

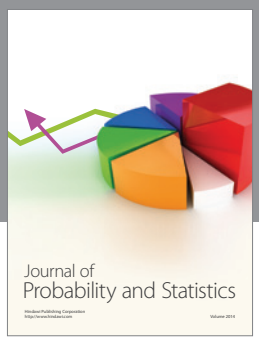
[17] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, Providence, RI, USA, June 2012.

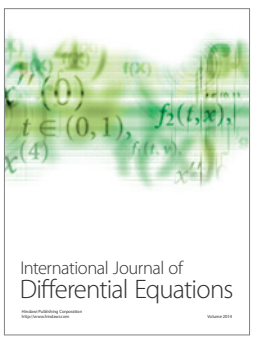
[18] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: exploring supporters and distracters in unconstrained environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1177–1184, Colorado Springs, Colo, USA, June 2011.

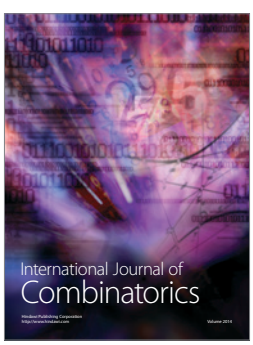


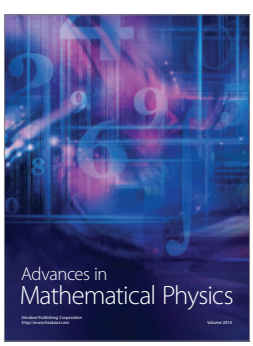


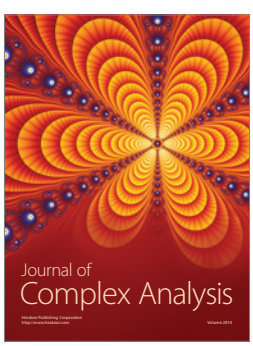


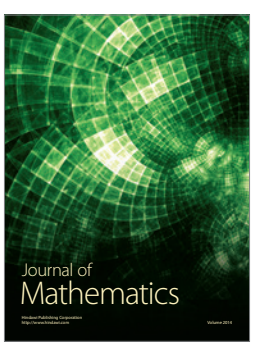


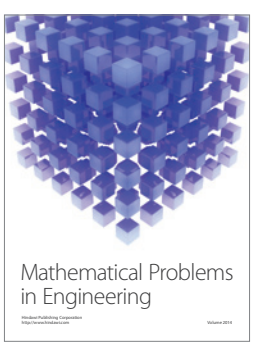


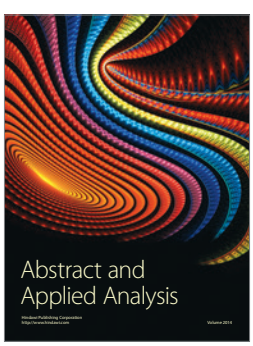


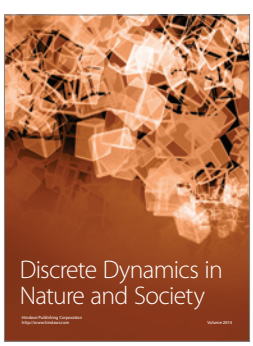


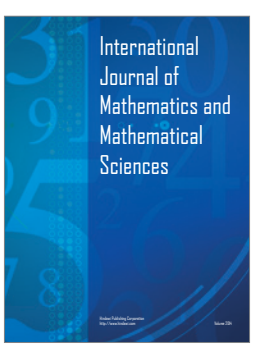


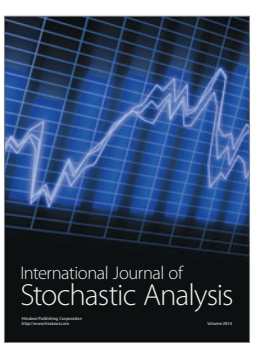












Hindawi

Submit your manuscripts at
https://www.hindawi.com