Robust Visual Tracking Using the Time-Reversibility Constraint

Hao Wu[†], Rama Chellappa[†], Aswin C. Sankaranarayanan[†] and Shaohua Kevin Zhou[‡]
[†]Center for Automation Research, University of Maryland, College Park, MD 20742
{wh2003, rama, aswch }@cfar.umd.edu
[‡]Integrated Data Systems Department, Siemens Corporate Research, Princeton, NJ 08540
kzhou@scr.siemens.com

Abstract

Visual tracking is a very important front-end to many vision applications. We present a new framework for robust visual tracking in this paper. Instead of just looking forward in the time domain, we incorporate both forward and backward processing of video frames using a novel time-reversibility constraint. This leads to a new minimization criterion that combines the forward and backward similarity functions and the distances of the state vectors between the forward and backward states of the tracker. The new framework reduces the possibility of the tracker getting stuck in local minima and significantly improves the tracking robustness and accuracy. Our approach is general enough to be incorporated into most of the current tracking algorithms. We illustrate the improvements due to the proposed approach for the popular KLT tracker and a search based tracker. The experimental results show that the improved KLT tracker significantly outperforms the original KLT tracker. The time-reversibility constraint used for tracking can be incorporated to improve the performance of optical flow, mean shift tracking and other algorithms.

1. Introduction

Visual tracking has become more and more important in computer vision research owing to its wide applications in visual surveillance and motion analysis. In physics, motion refers to the act of changing location from one place to another, relative to a reference point, as measured by a particular observer in a particular frame of reference; therefore, the goal of visual tracking is to find and describe the relative position change of the moving object according to the recorded video frames. Given that the kinematic and dynamic information of the object can not be observed from the video directly, researchers try to infer the motion information through some observable properties of the object under motion, among which the appearance of the object is probably the most widely used. There are several tracking algorithms that have been developed based on the assumption that the appearance of the object is unchanged or the change can be explained using some object motion models. Most matching based tracking algorithms fall in this category, along with the popular KLT [11][17][15] and the mean shift tracking algorithm [5][6]. To improve the tracking performance, various 2D or 3D appearance models have been developed to handle the appearance change during tracking. Researchers have also incorporated additional knowledge, like camera calibration, scene geometry, and scene illumination for appearance based tracking. Another kind of tracking algorithms like the CONDENSATION [8] and those based on particle filtering [7] directly incorporate the dynamic model of the object motion into tracking methods. These algorithms generally involve an object state transition model and an observation model, which combines both motion and appearance models of the object. The dynamic models used in these algorithms are generally loosely defined, which are good for general object motion, not just for a specific motion model, like constant speed, constant acceleration or a random walk.

Researchers have taken various approaches to exploit and incorporate more information about the true object motion, like adaptive speed or trajectory prediction models [19]; however, to the best of our knowledge, a fundamental characteristic called *time-reversibility* of object motion has been ignored in most of the past and current works. The intuition relies on the idea that, since all the targets of interest are macroscopic solid objects in the physical world and obey physical laws of classical mechanics that are time-symmetric, all the motion process of the targets should be time-reversible; which means that the time-reversed process satisfies the same dynamical equations as the original process. However, most of the existing tracking algorithms only look forward in the time domain instead of looking bidirectionally during tracking.

If we look at the tracking problem as a black box shown in Figure 1, the inputs of the black box are the observations

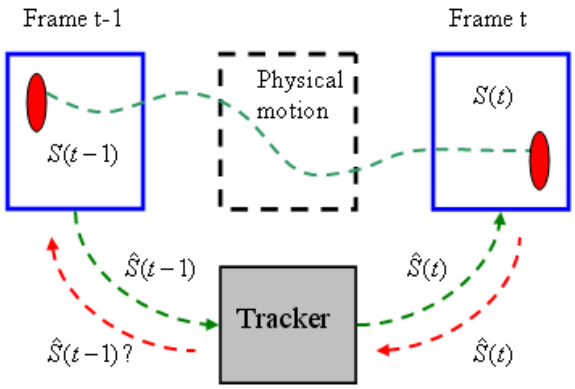

Figure 1. An illustration of visual tracking

and object state at $t-1$, the output is the object state at the current frame t. The process of how the object evolves from the previous state to the current state is not totally obvious to the observers due to the limited information contained in the video data. We claim that irrespective of the nature of object motion, if we switch the time axis in the physical world, we can expect that the object will go back to the exactly same state at time $t-1$. Hence, if the tracking strategy does capture the true object motion during this time interval, which implies that the object state is well estimated at time t, then using the same tracking strategy, the backward estimated object state at time $t-1$ should be identical to the forward state at time $t-1$. On the contrary, if the tracking algorithms does not preserve the time-reversibility characteristic of the motion process, it is very possible that it has failed to track the object or may fail soon. In practice, it is very unlikely for the forward tracking algorithms to maintain the time-reversibility property with time. A similar idea is used for evaluating the performance of tracking algorithms in [18].

In this paper, we present a new framework for robust visual tracking. Instead of just looking forward in the time domain, we simultaneously perform both the forward and backward tracking using the time-reversibility constraint. The new framework reduces the possibility of the tracker getting stuck in local minima and significantly improves the tracking robustness and accuracy. We illustrate the improvements due to the proposed approach for the popular KLT tracker and a search based tracker. The experimental results show that the improved KLT tracker significantly outperforms the original KLT tracker without demanding additional computational resources. The proposed time-reversibility constraint is general enough to be incorporated into most of the current tracking algorithms, as well as for computing optical flow.

Before proceeding further, we clarify the notion of time-reversibility and distinguish it from backward processing or smoothing. Some algorithms also consider backward tracking [16][2]; however, they perform forward and backward tracking separately and then merge the results in order to get better performance. CONDENSATION and particle smoothing methods [10][9][7] consider both the past and future observations to get a smoothed estimate, which also involves backward processing. There are some other works using backward tracking in video coding area [13][14]; however, these strategies are not focused on exploiting the notion of time-reversibility in tracking. The consistent image registration method proposed by Christensen [3] and Christensen and Johnson [4] is probably the closest to exploiting time-reversibility in spirit.

In Section 2, we will first present a new framework for tracking using the time-reversibility constraint; then the new KLT tracker using the time-reversibility constraint will be described in Section 3; detailed experimental results and analysis on the new KLT algorithm, compared to the original KLT tracker, will be given in Section 4; conclusions and potential future works are in Section 5.

2. The New Tracking Framework with the Time-Reversibility Constraint

Many of the current tracking algorithms maximize a likelihood function or minimize a cost function. Without loss of generality, we can formulate the tracking problem using the Bayesian framework as follows: determine

$$\hat{S}_t = \arg\max_{S_t} P(S_t \mid \hat{S}_{1,\ldots,t-1}, Y_{1,\ldots,t}) \tag{1}$$

where S_t is the state of the object at time t. The state can be one or more templates, state vectors or probability density functions that describe the status of the object at a specific time; Y_t is the observation at time t; usually it is the image frame or some abstracted features. The above framework can represent most of the current forward-looking tracking algorithms; however, as stated above, this framework ignores the relationship between the forward and backward tracking results, which exhibits an important characteristic called time-reversibility present in all kinds of object motions including constant, abrupt or smoothly changing motions. Hence, we present the new tracking framework below:

$$\begin{aligned}\hat{S}_t = \arg\max_{S_t} & P(S_t \mid \hat{S}_{1,\ldots,t-1}, Y_{1,\ldots,t}) \\ & + P(S^b_{t-1} \mid S_t, Y_{t,t-1}) + \lambda L(\hat{S}_{t-1}, S^b_{t-1})\end{aligned} \tag{2}$$

where S^b_{t-1} is the backward estimate at time $t-1$ and $L(\cdot)$ is the likelihood between the forward and backward estimation results; the first term is the same as in the forward-only tracking approach; the second term is the dual term representing tracking backward from time t to $t-1$; but sometimes it is not enough to achieve or approach time-reversibility with only the first two terms. To improve the

performance further, we need to explicitly add a constraint derived from the difference between the forward estimate $\hat{S}_{t-1}$ and the backward estimate S^b_{t-1}, which yields the third term in the above equation.

Since the likelihood functions usually have multiple local minima, especially for high dimensional data like images, the tracker may get stuck in some local minima in practice. We believe that the incorporation of these additional constraints reduces a fair number of local minima.

In traditional Bayesian smoothing algorithms, researchers aim to compute the following:

$$\hat{S}_t = \arg\max_{S_t} P(S_t \mid \hat{S}_{1,\ldots,t-1}, Y_{1,\ldots,T}) \tag{3}$$

where T is larger than t. We compare Bayesian smoothing and the proposed tracking framework with the time-reversibility constraint as below:

- First, Bayesian smoothing requires future data to improve the performance while there is no need for more data in the tracking algorithms using the time-reversibility constraint. The time-reversibility constraint provides more modeling constraints.
- Second, Bayesian smoothing requires much more computation than Bayesian filtering, not only due to more data being processed but also due to the optimization process which looks at all the interactions between each state and observation variables. For particle smoothing, in general, the computational complexity is $O(N^2)$ while for particle filtering it is only $O(N)$ [10]. On the contrary, the tracking algorithms using the time-reversibility constraint generally do not lead to more computational load unless an exhaustive search method is used in the optimization process.

There is no contradiction between the time-reversibility constraint and the Bayesian smoothing strategy. In Bayesian smoothing, the performance improvement is due to the backward-flow of information from future data while time-reversibility means that the entropy involved in the motion process is non-increasing. In practice, time-reversibility is only approximately satisfied due to noise or partial observations. Bayesian filtering or smoothing tries to minimize the information decrease during tracking [12], which in effect is similar to using the time-reversibility constraint.

3. The New KLT with the Time-Reversibility Constraint

The basic idea of the Kanade-Lucas-Tomasi (KLT) feature tracker first appeared in Lucas and Kanade's paper [11] in 1981; it was fully developed by Tomasi and Kanade [17] in 1991. In 1994, Shi and Tomasi [15] presented a KLT based method to select good features to track. In the past decade, KLT is probably the most widely used feature tracker in many applications, such as structure from motion, and computation of optical flow. In this section, we improve the original KLT algorithm using the time-reversibility constraint presented in section 2.

3.1. The Original KLT

The original KLT algorithm assumes that the intensity of the features remains constant when a camera moves, that is, $I(x, y) = J(x+\xi, y+\eta)$ assuming that the motion between two consecutive frames can be described as pure translation. This leads to the following objective function:

$$(\hat{\xi}, \hat{\eta}) = \arg\min_{\xi,\eta} \int\int_W [J(x+\xi, y+\eta) - I(x, y)]^2 * w(x, y)dxdy \tag{4}$$

Later, a symmetric expression is used to derive the solution [1]:

$$(\hat{\xi}, \hat{\eta}) = \arg\min_{\xi,\eta} \int\int_W [J(\mathbf{p} + \frac{\mathbf{d}}{2}) - I(\mathbf{p} - \frac{\mathbf{d}}{2})]^2 * w(\mathbf{p})d\mathbf{p} \tag{5}$$

where $\mathbf{p} = (x, y)^T$ and $\mathbf{d} = (\xi, \eta)^T$ and the weighting function $w(\mathbf{p})$ is usually set to 1 over the domain W. For simplicity we will omit all the function variables and $w(\mathbf{p})$ in the following. We also use the discrete form for the integrals involved in the derivations. Using a first order Taylor expansion to linearize the above nonlinear objective function and setting the derivative with respect to $\mathbf{d}$ to zero, we get:

$$\sum\sum_W (J - I + \mathbf{g}^T\mathbf{d})\mathbf{g} = 0 \tag{6}$$

$$\mathbf{g} = (\nabla\frac{I + J}{2})^T \tag{7}$$

This can be rearranged as:

$$\mathbf{Zd} = \mathbf{e} \tag{8}$$

where

$$\mathbf{Z} = \sum\sum_W \mathbf{g}\mathbf{g}^T \tag{9}$$

$$\mathbf{e} = \sum\sum_W (I - J)\mathbf{g} \tag{10}$$

We notice that the final solution seems to smooth the forward and backward tracking results by simply averaging them. In general, the forward KLT results will differ from the backward KLT results due to asymmetry in image information.

The symmetric expression in (4) may be confused with the time-reversibility constraint proposed in this paper. However, we can see from (4) that after switching to symmetric expression, the objective function tries to minimize the difference between $I(\mathbf{p}-\frac{\mathbf{d}}{2})$ and $J(\mathbf{p}+\frac{\mathbf{d}}{2})$ while the original purpose is to find the best match for the feature centering at $I(\mathbf{p})$. The notion of symmetry used here does not imply time-reversibility and vise versa.

3.2. The New KLT using the Time-Reversibility Constraint

3.2.1 The Derivation of the New KLT algorithm

Following the original definitions in KLT, we propose a new objective function for KLT using the time-reversibility constraint.

$$\begin{aligned}(\hat{\mathbf{d}},\hat{\mathbf{d}}^{\mathbf{b}}) &= \arg\min_{\mathbf{d},\mathbf{d}^{\mathbf{b}}} \int\int_W [J(\mathbf{p}+\mathbf{d})-I(\mathbf{p})]^2 w(\mathbf{p}) d\mathbf{p} \\ &\quad + \int\int_W [I(\mathbf{p}+\mathbf{d}+\mathbf{d}^{\mathbf{b}})-J(\mathbf{p}+\mathbf{d})]^2 w(\mathbf{p}) d\mathbf{p} \\ &\quad + \lambda(\mathbf{d}+\mathbf{d}^{\mathbf{b}})^T(\mathbf{d}+\mathbf{d}^{\mathbf{b}}) \qquad (11)\end{aligned}$$

where $\mathbf{d}^{\mathbf{b}}$ is the backward displacement vector when tracking from t to $t-1$. Assuming $\mathbf{d}$ and $\mathbf{d}^{\mathbf{b}}$ are small, we can perform the following approximations using the first order Taylor expansion:

$$\begin{aligned}I(\mathbf{p}+\mathbf{d}+\mathbf{d}^{\mathbf{b}}) &\approx I(\mathbf{p})+\nabla I(\mathbf{d}+\mathbf{d}^{\mathbf{b}}) \\ J(\mathbf{p}+\mathbf{d}) &\approx J(\mathbf{p})+\nabla J\mathbf{d} \qquad (12)\end{aligned}$$

By setting the derivatives with respect to $\mathbf{d}$ and $\mathbf{d}^{\mathbf{b}}$ to zero respectively, we have the following constraints which are given in their discrete forms:

$$\begin{aligned}0 &= \sum\sum_W [H(\nabla H-\nabla J)^T + (\nabla H^T\nabla I)\mathbf{d}^{\mathbf{b}}] \\ &\quad + \sum\sum_W [(\nabla J^T\nabla J+\nabla H^T\nabla H)\mathbf{d}] + \lambda(\mathbf{d}+\mathbf{d}^{\mathbf{b}}) \\ 0 &= \sum\sum_W [H\nabla I^T + (\nabla I^T\nabla H)\mathbf{d}] \\ &\quad + \sum\sum_W (\nabla I^T\nabla I)\mathbf{d}^{\mathbf{b}} + \lambda(\mathbf{d}+\mathbf{d}^{\mathbf{b}}) \qquad (13)\end{aligned}$$

where $H=I-J$. Solving these equations, we finally get:

$$\begin{aligned}\mathbf{U}\mathbf{d} &= \varepsilon \\ \mathbf{U} &= AD^{-1}C+\lambda D^{-1}C-\frac{1}{2}B \\ \varepsilon &= (A+\lambda I)D^{-1}(V-W)+\frac{1}{2}(S-R) \qquad (14)\end{aligned}$$

where the definitions of A,B,C,D,V,W,S,R are given below:

$$\begin{aligned}A &= \sum\sum_W (\nabla I)^T\nabla I; B=\sum\sum_W (\nabla I)^T\nabla J; \\ C &= \sum\sum_W (\nabla J)^T\nabla J; D=\sum\sum_W (\nabla J)^T\nabla I; \\ R &= \sum\sum_W I(\nabla I)^T; S=\sum\sum_W J(\nabla I)^T; \\ V &= \sum\sum_W I(\nabla J)^T; W=\sum\sum_W J(\nabla J)^T; \qquad (15)\end{aligned}$$

If we re-write the original KLT equation using the above defined variables, we get:

$$\begin{aligned}\mathbf{Z} &= \frac{1}{2}(A+B+C+D); \\ \mathbf{e} &= \frac{1}{2}(R-S+V-W); \qquad (16)\end{aligned}$$

3.2.2 Comparison to the Original KLT

By comparing the new KLT and the original KLT, we find that although all the variables appear in both the original and the new KLT equations, the interactions between these variables are different. The original KLT just linearly combines the forward-only and backward-only mappings while in the new KLT process, the forward and backward are performed simultaneously, actually improving the performances of each other.

We also note that the new KLT has almost the same computational cost as the original one. In practice, the new KLT requires lesser computations than the original one because the required number of iterations is lower to achieve the same performance.

An interesting observation is that even when $\lambda=0$, the new KLT still has a completely different expression from the original KLT. When $\lambda=0$, $\mathbf{U}$ and ε are:

$$\begin{aligned}\mathbf{U} &= AD^{-1}C-\frac{1}{2}B \\ \varepsilon &= AD^{-1}(V-W)+\frac{1}{2}(S-R) \qquad (17)\end{aligned}$$

This is due to the second term in (11). In our experiments, we find that the new KLT outperforms the original KLT even when λ equals to 0. The reason is that optimizing the first two terms in (11) still involves the interaction between forward and backward processing, implicitly enforcing the time-reversibility constraint. In fact, the second and third terms in (11) both improve the tracking performance. We will further study their contributions in section 4.

3.3. Good Features to Track

Shi and Tomasi [15] presented a method to select good features to track. The method is based on the rationale that

if both the eigenvalues of the matrix R as defined above are larger than some threshold, which implies that the KLT equation can be robustly solved, then the feature in general will exhibit complex textures and be good for tracking.

Since the new KLT has the same form as the original one, we can also judge if a feature is good for tracking or not. However, in both symmetric KLT and the proposed KLT, the corresponding matrix contains information from two images, so the explicit physical interpretation of the eigenvalues is hard to see. Both the original KLT matrix and the new KLT matrix show good properties in practice in terms of their stability in solving the equations. We studied the condition number of the both matrices, which is in general an indication of good stability if the value is close to 1. Experimental results show that the condition number of the new KLT matrix is on the average closer to 1 than the original KLT. This provides an explanation for the improvement due to the new constraint.

As different tracking algorithms lead to different matrices, so evaluating if a feature is good or not for tracking is not completely determined by the characteristic of the feature itself. A bad feature in one algorithm can be good in the other. In the experiments, we find that the proposed new KLT can track some features well while the original KLT fails in tracking them.

4. Experimental Results and Discussions

We implemented the new KLT algorithm based on the latest C code of the original KLT algorithm which is widely used and can be downloaded from the website: http://www.ces.clemson.edu/ stb/klt/.

4.1. Performance Evaluation on Clean Sequences

4.1.1 With No Ground Truth

First, we compare the results on the sequence contained in the KLT code package, which we call 'the Table sequence'. In the Table sequence the camera exhibits a rotation from left to right. To fairly evaluate the performance, we used the same set of parameters for both algorithms. We also disabled some thresholds for removing bad features during the tracking procedure unless they are out of the image region.

We selected 200 feature points for both algorithms using the same method contained in the package which is based on [15], so the starting feature points are the same for both algorithms. The iteration number is set to 10 which is sufficient for both algorithms to converge. Figure 2 shows the results for both algorithms at the starting and ending frames together with some enlarged details. The feature points are colored as red dots. We circled the points where the two algorithms differ significantly, say, by more than 3 pixels. By visual inspection, we found that the new KLT kept tracking well on those points while the original KLT lost track of them.

Figure 2. Tracking results of the original KLT (left column) and the proposed KLT (right column) using the time-reversibility constraint at the starting and ending frames. The green circled points on the left side differ significantly with the yellow circled points on the right side. It is easy to see that the new KLT keeps tracking those points well while the original KLT fails to track them.

4.1.2 With Generated Ground Truth

To further quantitatively evaluate the performance, we use the first frame of the Table sequence to generate a new sequence with random translations in both x and y directions. Therefore all the features points have the same motion as the generated random translations. Two-level image pyramids are used in both algorithms. The results of the algorithms are then compared with ground truth. The effect of different λ has been studied on two generated sequences with different motions. We summarize the quantitative results in Table 1 and 2. For the sequence with translation distributed between 0 to 12, the mean error curves with respect to λ are plotted in Figure 3 with red and blue lines for the original and new KLT respectively. As we can see, the new KLT using the time-reversibility constraint consistently outperforms the original KLT in different λ values. The average improvement is more than 35%.

In this sequence, the change due to λ is quite small be-

error	$\lambda =$	0	0.05	0.2	2	20
n		1427	1427	1427	1427	1426
mean	original KLT	0.2351	0.2351	0.2351	0.2351	0.2353
	new KLT	0.1429	0.1424	0.1435	0.1487	0.1429
variance	original KLT	1.7419	1.7419	1.7419	1.7419	1.7431
	new KLT	0.7935	0.7936	0.7928	0.7983	0.7898

Table 1. The generated random translation is uniform between 0 and 12 pixels in both directions, where we can see the best performance is achieved when $\lambda = 0.05$. The value of λ is normalized by the size of the feature window. n is the number of points involved in the experiment. The value of error is in pixel unit.

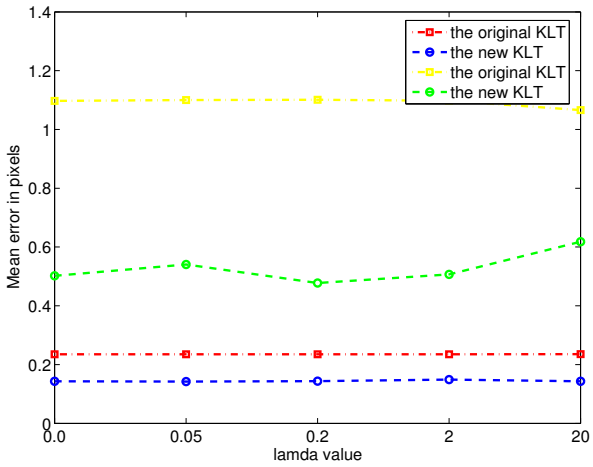

Figure 3. The mean errors of the original KLT and the new KLT. The red and blue lines are the results for the sequence with translation uniformly distributed from 0 to 12; the yellow and green lines are the results for the sequence with translation uniformly distributed from 0 to 20.

tween $\lambda = [0, 20]$. Comparing the error at $\lambda = 0$ and others, there is no big change while the performance is still much better than the original KLT. This result shows that the second term in (11) does contribute to improving the results. Both the average errors of the original and new KLT methods on this sequence are quite small, primarily due to the motion being small translation. From the results, we probably can say that the improvement on good sequences mainly comes from the second term.

We generated another sequence with increased translation magnitude. We can see the improvement due to the third term in Table 2. We find the value of λ to achieve the best performance increased in the new sequence with larger motion. The mean errors of both algorithms are shown in yellow and green curves respectively in Figure 3. This result tells us that the third term helps more when the original KLT has more difficulties in tracking than in the situations where it is easier to track. Figure 4 shows the tracking results of both algorithms at the ending frame, where λ used in the new KLT equals to 0.2.

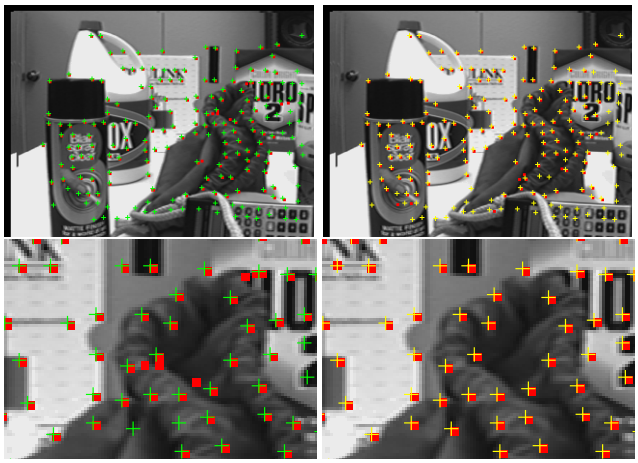

Figure 4. Tracking results of the original KLT (left column) and the proposed improved KLT (right column) at the ending frame on the generated sequence with known ground truth. The green and yellow cross signs provide the ground truth positions of the feature points. The up-left corner of the small red block should overlay on the center of the cross if tracking were perfect.

error	$\lambda =$	0	0.05	0.2	2	20
n		1235	1232	1231	1233	1237
mean	original KLT	1.0973	1.1001	1.1009	1.0974	1.0659
	new KLT	0.5019	0.5405	0.4777	0.5066	0.6181
variance	original KLT	40.2333	40.3281	40.3601	40.2967	39.2340
	new KLT	9.9528	13.7966	9.2691	9.4448	11.3291

Table 2. The generated random translation is uniform between 0 and 20 pixels in both directions, where we can see the best performance is achieved at $\lambda = 0.2$. The value of λ is normalized by the size of the feature window. n is the number of points involved in the experiment.The value of error is in pixel unit.

4.2. Performance Comparison on Noisy Sequences

We add Gaussian noise to the sequence with random translations uniformly distributed between 0 and 12. The variance of the zero-mean noise is 0.005 which is normalized by the range of the image intensity. Table 3 shows the results of both algorithms for different λ; and the mean errors are plotted in Figure 5. It is seen that the best performance is achieved at $\lambda = 40$, which confirms the above conclusion that larger value of λ should be used for more difficult sequences. This is because the values of the first two intensity evaluation terms increase for more difficult sequences; thus to make the third term comparable to the first two terms, a larger λ is needed. Figure 6 provides the tracking results of both algorithms, where λ equals to 40 in the proposed new KLT.

4.3. Speed Comparison

The new KLT using the time-reversibility constraint is a real-time algorithm, which does not add noticeable increase in computational cost, compared to the original KLT. This can be expected from the similar linear equations solved in

error	$\lambda =$	0	2	20	40	60
n		1311	1310	1282	1269	1264
mean	original KLT	1.3192	1.3158	1.3395	1.3331	1.3210
	new KLT	0.9666	0.9278	0.9180	0.9165	0.9306
variance	original KLT	11.9968	12.0254	12.3895	12.1590	11.7519
	new KLT	2.4290	2.3080	3.3098	3.4253	3.5468

Table 3. Results on the noisy sequence with random translation uniformly distributed from 0 and 12 pixels in both directions, where we can see the best performance is achieved at $\lambda = 40$. The value of λ is normalized by the size of the feature window. n is the number of points involved in the experiment. The value of error is in pixel unit.

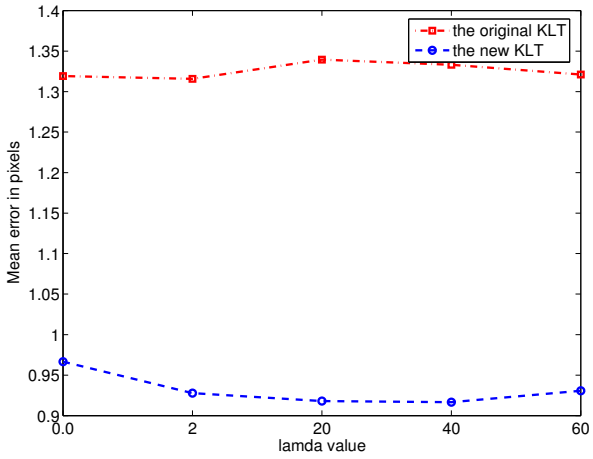

Figure 5. The mean error of the original KLT and the new KLT. Gaussian noise is added in the sequence with translations uniformly distributed from 0 to 12.

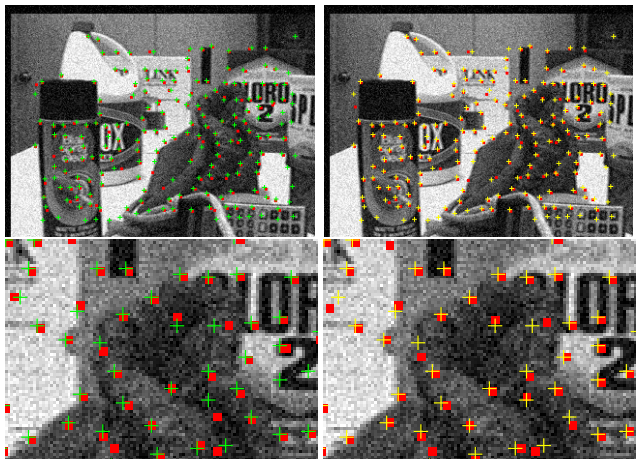

Figure 6. The tracking results of the original KLT (left column) and the improved KLT (right column) on a noisy sequence at the ending frame. The green and yellow cross points provide the ground truth positions of the feature points. The up-left corner of the small red block should overlay on the center of the cross if tracking were perfect.

both the original and new KLT. We plot the mean error of both algorithms under different iteration numbers in Figure 7. It can be seen that the new KLT even converges faster

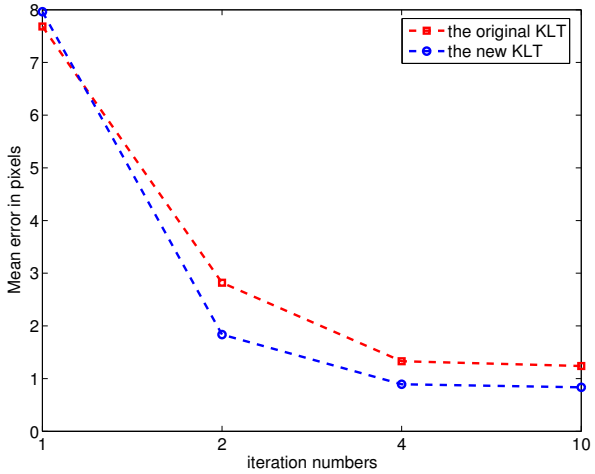

Figure 7. The mean error of the original KLT and the new KLT under different iteration numbers from 1 to 10.

than the original KLT although only by a small factor.

4.4. Good Features to Track

We studied the condition number in the original KLT matrix $\mathbf{Z}$ and the new KLT matrix $\mathbf{U}$. The condition number here is defined as the absolute ratio of the maximal eigenvalue to the minimal eigenvalue of a matrix. When solving a linear equation like $\mathbf{M}x = \mu$, if the condition number of $\mathbf{M}$ is large, even a small error in μ may cause a large error in x. In the original Table sequence, the average condition number of the original KLT matrix is about 8.9171 while the condition number of the new KLT matrix is about 2.6621. This is based on the evaluations of 200 points across 10 frames. And from Figure 8, we also find that the condition number increases in the original KLT while it remains nearly constant in the new KLT using the time-reversibility constraint. We believe that this explains the performance improvement due to the new constraint, from the view of numerical computation stability.

4.5. Additional Results on Large Object Tracking

To test the improvement of the new tracking strategy using the time-reversibility constraint on tracking large objects, we performed an exhaustive search based tracking on a generated very noisy sequence. The objective functions used are the same as the original KLT and the new KLT. The difference is in that in KLT, a gradient decent like search method is used while an exhaustive search method is used here. The results are shown in Figure 9. We can see that using the time-reversibility constraint, the block can be tracked well while it can not be tracked without such a constraint. Thus we believe that combining the time-reversibility constraint with some large-object tracking algorithms will improve the tracking performance too.

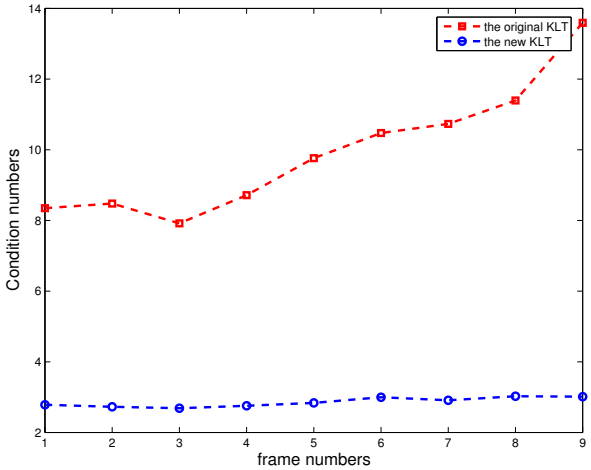

Figure 8. The average condition number of the matrices for 200 points at each frame during tracking.

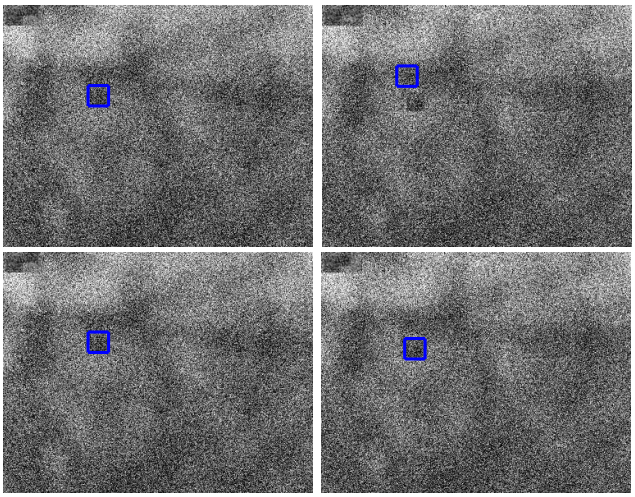

Figure 9. Tracking the black block by searching in a neighborhood around the object using a fixed appearance model. The top row shows the result without the time-reversibility constraint at the starting and ending frames, which fails to track the block. The bottom row shows the result using the time-reversibility constraint at the corresponding frames, which obtained good tracking of the block.

5. Conclusions

In this paper, we present a new framework for visual tracking algorithms using the time-reversibility constraint, which has not been studied before. We applied this idea to the popular KLT feature tracking algorithm and developed a new KLT algorithm using the time-reversibility constraint. Extensive experiments are performed to compare the performance between the original KLT and the new KLT. The results show that the performance of the new KLT algorithm has been significantly improved. A simple experiment on tracking large object is also given, which shows that the proposed strategy is very promising for tracking large objects. The work on improving other tracking algorithms, such as mean shift tracking, and optical flow methods will be studied in the future.

References

[1] S. Birchfield. Derivation of kanade-lucas-tomasi tracking equation. *Unpublished*, January 1997.

[2] T. Caljon, V. Enescu, P. Schelkens, and H. Sahli. An offline bidirectional tracking scheme. *Advanced Concepts for Intelligent Vision Systems*, 2005, Antwerpen.

[3] G. Christensen. Consistent linear-elastic transformations for image matching. *Information Processing in Medical Imaging, LCNS 1613*, pages 224–237, June 1999.

[4] G. Christensen and H. Johnson. Consistent image registration. *IEEE Transactions on Medical Imaging*, pages 568–582, July 2001.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transaction on Pattern Analysis Machine Intellegence*, pages 564–575, 2003.

[7] A. Doucet, N. D. Freitas, and N. Gordon. *Sequential Monto Carlo Methods in Practice*. New York, NY: Springer-Verlag, 2001.

[8] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[9] M. Isard and A. Blake. A smoothing filter for condensation. *Proceedings of the 5th European Conference on Computer Vision*, 1406:767–781, 1998.

[10] M. Klass, M. Briers, N. D. Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: If i had a million particles. *Proc. IEEE Int. Conf. on Machine Learning.*, 2006.

[11] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[12] S. K. Mitter and N. J. Newton. Information and entropy flow in the kalman-bucy filter. *Journal of Statistical Physics*, (12), January 2005.

[13] T. Shanableh and M. Ghanbari. Backward tracking of b-pictures bidirectional motion for interframe concealment of anchor pictures. *Proc. IEEE Int'l Conf. on Image Processing*, 3:396–399, 2000.

[14] T. Shanableh and M. Ghanbari. The importance of the bi-directionally predicted pictures in video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):402–414, March 2001.

[15] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[16] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Bidirectional tracking using trajectory segment analysis. *Proc. IEEE Int'l Conf. on Computer Vision*, 1:717–724, 2005.

[17] C. Tomasi and T. Kanade. Detection and tracking of point features. *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.

[18] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. In situ evaluation of tracking algorithms using time reversed chain. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[19] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38, Dec. 2006.