A Direct 3D Object Tracking Method Based on Dynamic Textured Model Rendering and Extended Dense Feature Fields

Leisheng Zhong, Ming Lu, Li Zhang, Tsinghua University

Overview

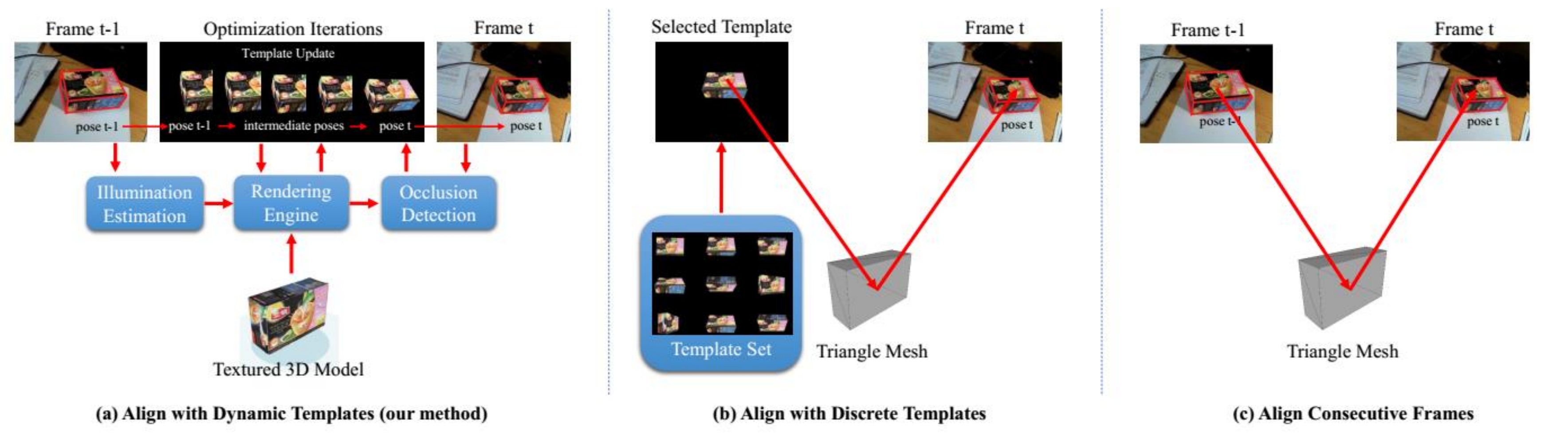
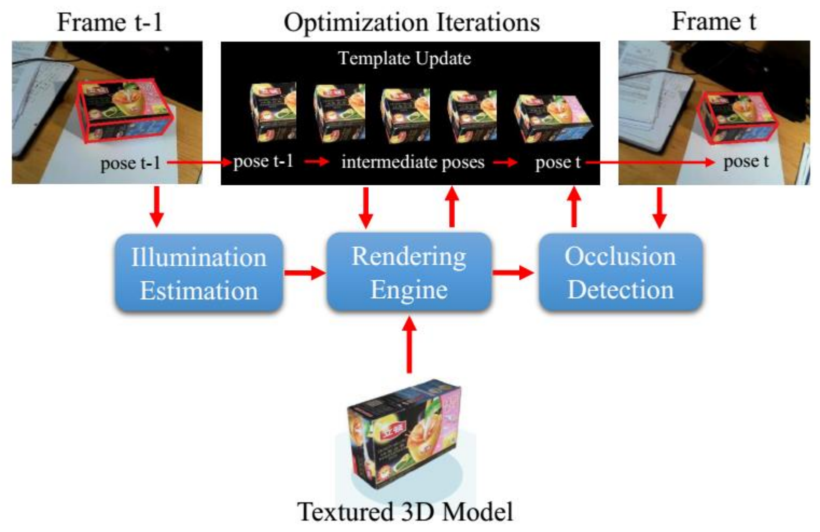


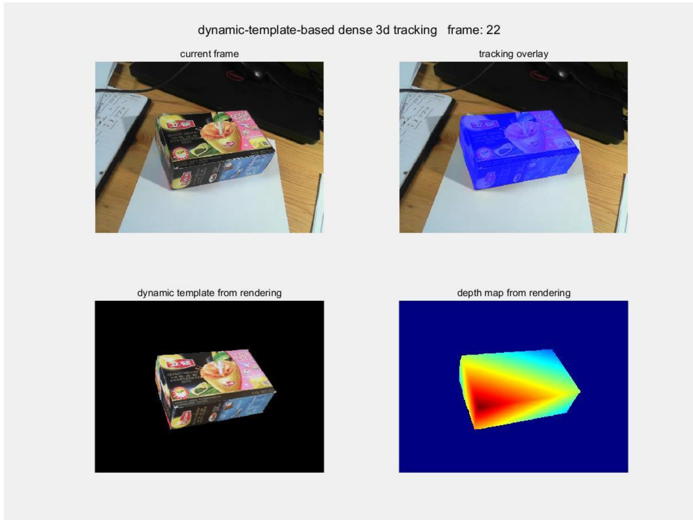
Fig. 1. Comparison of the proposed method with previous direct tracking methods. (a) We propose to align the current frame with dynamically rendered templates from a pre-built textured 3D object model. (b) Some previous works try to align the current frame with a template selected from a small set of discrete templates. (c) Other previous works align consecutive video frames over a triangle mesh model.

Main Contributions:

1. Applying dynamic textured model rendering to direct 3D object tracking.
2. Constructing a generic representation of dense features for direct image alignment.
3. Employing a simple yet efficient occlusion detection process in the tracking pipeline.

Dynamic Textured Model Rendering

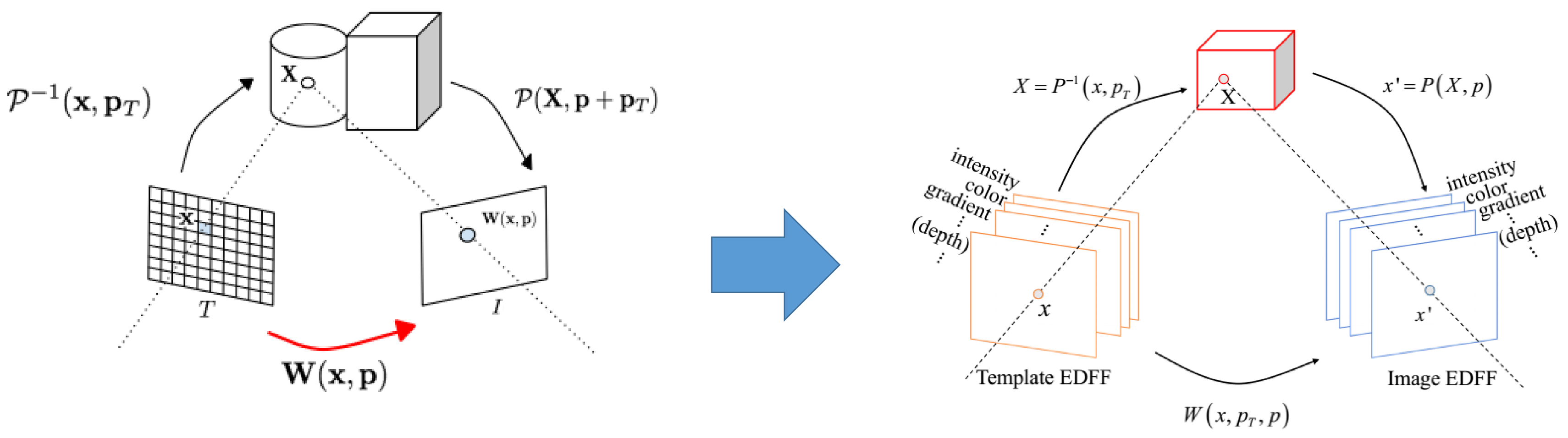




The benefits of dynamic textured model rendering:

1. A good initial template for the current frame is automatically acquired by rendering the object with the latest tracking result.
2. Template image could be dynamically updated during optimization iterations.
3. The rendered template is occlusion-free and background-free, which helps to detect occlusion and avoid cluttered background interferences.

Extended Dense Feature Fields



$$E(p) = \sum_x \|Int(I, W(x,p)) - Int(T,x)\|^2$$

$$E(p) = \sum_x \|F(I, W(x, p_T, p)) - F(T,x)\|^2$$

$$E(p) = \sum_{i=1:d} w_i \left(\sum_x \|f_i(I, W(x, p_T, p)) - f_i(T,x)\|^2 \right)$$

$$J^T \mathbf{W} J \delta p = J^T \mathbf{W} (F(I, W(x, p_T, p)) - F(T,x))$$

Illumination Estimation

$$I = R \times S(\mathbf{N}, \mathbf{L})$$

$$I(i,j) = \rho(i,j) \sum_{k=0}^{8} l_k H_k(\mathbf{n}(i,j))$$

$$E_L = \sum_{i,j} \left\| I(i,j) - \sum_{k=0}^{8} l_k H_k(\mathbf{n}(i,j)) \right\|$$

$$T^{relight}(i,j) = \rho^T(i,j) \sum_{k=0}^{8} l_k^I H_k(\mathbf{n}(i,j))$$

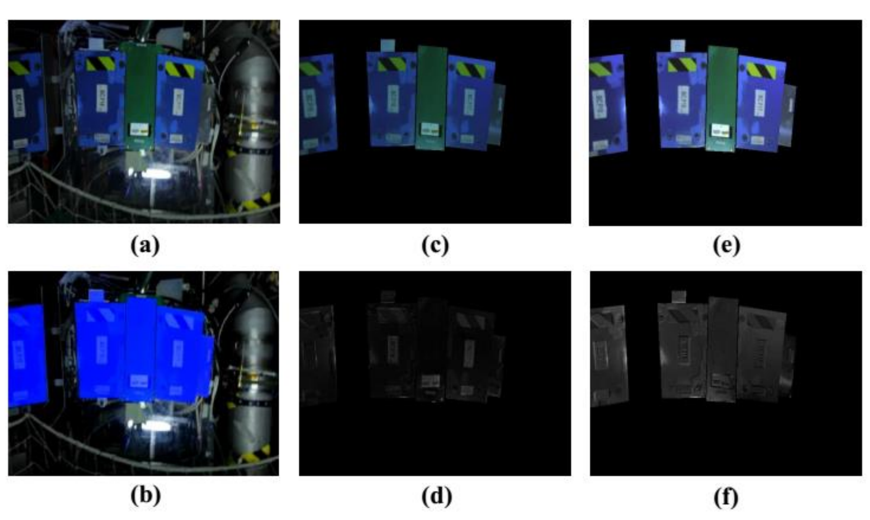


Fig. 4. Illumination Estimation. (a) A sample frame from ATLAS#2 in Dense Tracking Dataset [8]. (b) Object mask acquired from tracking result. Only pixels inside the mask are involved in illumination estimation. (c, d) Template re-lighted by the estimated illumination (of the previous frame) and the error image compared to the current frame. (e, f) Rendered template without re-lighting and the corresponding error image. After employing the estimated lighting parameters to the rendering engine, the rendered template is much more similar to the video frame.

Occlusion Detection

$$Occl = [|G^\sigma * I - G^\sigma * T| - \beta]^+ \oplus Strel$$

$$[x]^+ = \begin{cases} 1, & if\ x > 0 \\ 0, & otherwise \end{cases}$$

Fig. 5. Occlusion Detection. (a) A heavily occluded frame from Rigid Pose Dataset [11]. The tracked box is largely occluded by the teddy bear. (b) The template rendered with tracked pose from last frame. The rendered template is always clear and non-occluded, which contributes to a simple but efficient occlusion detection strategy. (c) The detected occlusion area. Pixels inside the area are discarded from pose tracking optimization.

Algorithm

Algorithm 1 3D Object Tracking Pipeline

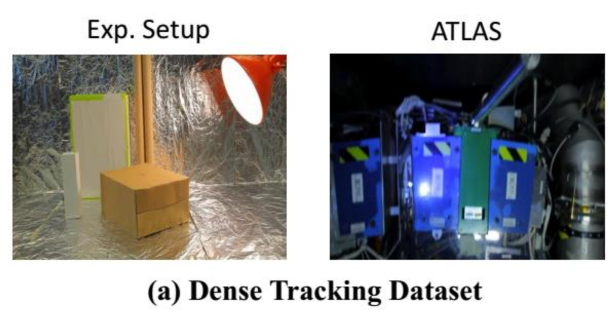
Input: Textured Model M, previous frame I_{k-1}, current frame I_k, previous pose p_{k-1}

Output: Current pose p_k

```
Render M in pose p_{k-1} to get the foreground mask of I_{k-1};
Estimate the illumination of I_{k-1} based on (8), using only pixels inside the foreground mask;
Re-render M in pose p_{k-1} with the estimated illumination to create the initial template T_0, read out the 3D coordinates of each foreground pixel from the renderer;
Calculate the template EDFF F(T_0, x) and the image EDFF F(I_k, x);
Detect the occlusion area based on (11), discard the occluded pixels;
Set p_k = p_{k-1}, p_T = p_{k-1}, T = T_0;
for a number of iterations do
    Calculate the stacked jacobian J and stacked residual Res = F(I, W(x, p_T, p_k)) - F(T, x) in (6), solve for the pose increment δp;
    Update current pose p_k = p_k + δp;
    if converged then
        Output p_k;
        Break;
    else
        Update template pose p_T = p_k;
        Update template T by rendering M in pose p_T, read out the 3D coordinates of each foreground pixel from the renderer, re-calculate the template EDFF F(T, x);
    end if
end for
```

Evaluation

Dense Tracking Dataset



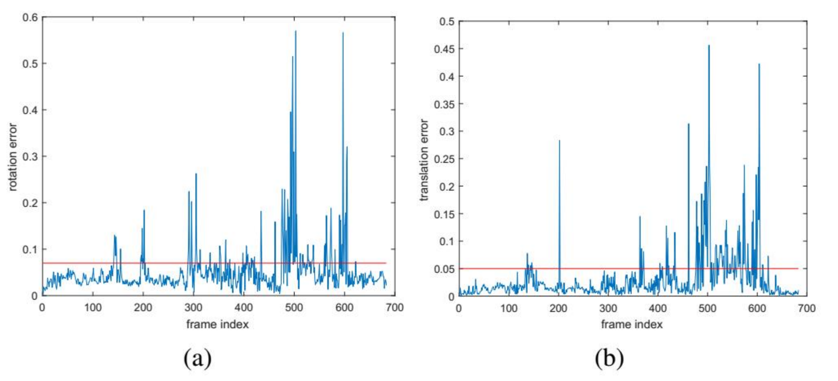
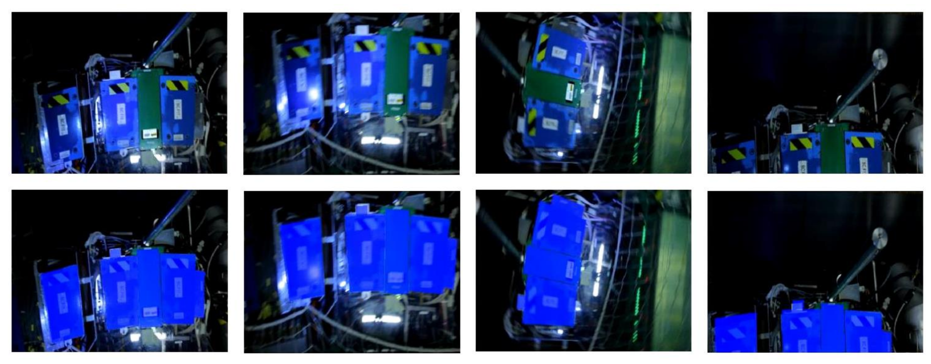


TABLE I
TRACKING SUCCESS RATE (IN %) ON DENSE TRACKING DATASET

	Exp#1	Exp#2	ATLAS#1	ATLAS#2
Intensity	42.1	22.2	88.6	22.5
Color	51.0	34.3	91.8	24.6
1^{st}-order Descriptor Fields	98.4	97.5	100	39.4
1^{st}- and 2^{nd}-order Descriptor Fields	92.8	97.8	100	33.4
Ours without illumination estimation	100	98.6	100	83.3
Ours with illumination estimation	**100**	**99.7**	**100**	**85.1**



Rigid Pose Dataset

TABLE III
TRACKING SUCCESS RATE (IN %) ON ORIGINAL AND NOISY SEQUENCES OF RIGID POSE DATASET

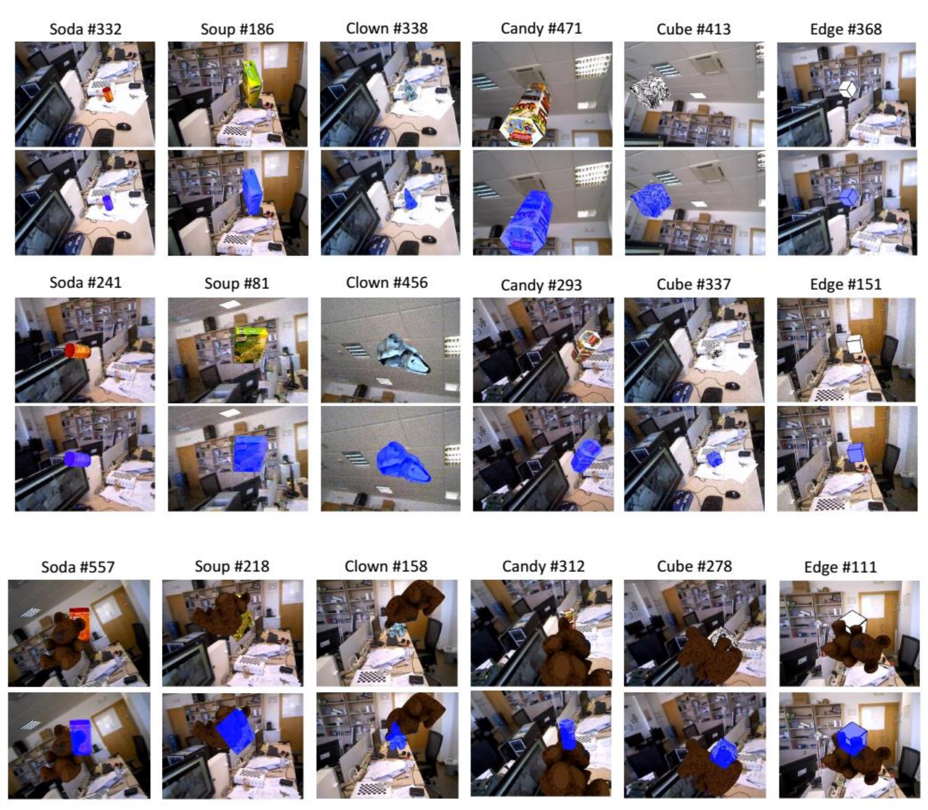
	soda		soup		clown		candy		cube		edge		average
	orig	noisy	orig	noisy	orig	noisy	orig	noisy	orig	noisy	orig	noisy	
Sparse-and-Dense	**99**	**97**	**98**	**99**	**100**	**98**	**100**	**100**	**100**	**100**	**98**	**98**	**98.9**
PWP3D	84	84	96	96	96	89	84	84	84	74	85	84	86.7
BLORT	76	65	77	66	88	82	77	76	93	94	72	91	79.8
Descriptor Fields	92	85	92	93	98	93	90	88	96	95	92	94	92.3
Consecutive	90	88	95	95	93	92	93	**93**	95	95	95	**95**	93.3
Ours	**94**	**90**	**96**	**96**	**98**	**95**	**94**	91	**98**	**96**	**95**	91	**94.5**

TABLE IV
TRACKING SUCCESS RATE (IN %) ON OCCLUDED SEQUENCES OF RIGID POSE DATASET

	soda	soup	clown	candy	cube	edge	average
Sparse-and-Dense	68	80	77	81	76	57	73.2
PWP3D	44	44	44	39	38	39	41.3
BLORT	54	63	76	64	76	68	66.8
Consecutive	66	76	68	81	71	65	71.2
Descriptor Fields without occlusion detection	50	54	48	66	53	40	51.8
Descriptor Fields with occlusion detection	74	84	81	76	80	67	77.0
Ours without occlusion detection	52	60	48	70	55	43	54.7
Ours with occlusion detection	**76**	**87**	**84**	**81**	**81**	**68**	**79.5**

TABLE V
TRACKING SUCCESS RATE (IN %) ON ORIGINAL AND NOISY SEQUENCES OF RIGID POSE DATASET (USING ONLY HALF OF THE FRAMES

	soda		soup		clown		candy		cube		edge		average
	orig	noisy	orig	noisy	orig	noisy	orig	noisy	orig	noisy	orig	noisy	
PWP3D	72	67	80	77	84	83	78	75	71	71	70	68	74.7
Descriptor Fields	71	70	73	73	82	82	80	77	82	76	86	85	78.1
Consecutive	77	74	85	84	80	80	77	75	78	77	89	82	79.8
Ours	**89**	**87**	**87**	**86**	**88**	**86**	**85**	**83**	**85**	**84**	**89**	**87**	**86.3**



Publication

1. Leisheng Zhong, Ming Lu, Li Zhang. A Direct 3D Object Tracking Method Based on Dynamic Textured Model Rendering and Extended Dense Feature Fields. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, Accepted. (IF=3.599)