Textureless Deformable Object Tracking with Invisible Markers

Xinyuan Li, Yu Guo, Yubei Tu, Yu Ji, Yanchen Liu, Jinwei Ye *Senior Member, IEEE*, and Changxi Zheng

Abstract—Tracking and reconstructing deformable objects with little texture is challenging due to the lack of features. Here we introduce "invisible markers" for accurate and robust correspondence matching and tracking. Our markers are visible only under ultraviolet (UV) light. We build a novel imaging system for capturing videos of deformed objects under their original untouched appearance (which may have little texture) and, simultaneously, with our markers. We develop an algorithm that first establishes accurate correspondences using video frames with markers, and then transfers them to the untouched views as ground-truth labels. In this way, we are able to generate high-quality labeled data for training learning-based algorithms. We contribute a large real-world dataset, DOT, for tracking deformable objects with little or no texture. Our dataset has about one million video frames of various types of deformable objects. We provide ground truth tracked correspondences in both 2D and 3D. We benchmark state-of-the-art methods on optical flow and deformable object reconstruction using our dataset, which poses great challenges. By training on DOT, their performance significantly improves, not only on our dataset, but also on other unseen data.

Index Terms—Deformable Object Tracking, Deformable Surface Reconstruction, Invisible Light Imaging

1 Introduction

Numerous applications, ranging from animation synthesis to robotic manipulation, need to track correspondences on deformable objects in order to understand their motion and deformation. Typical examples include cloth acquisition [1], [2] and gesture recognition [3], [4].

Many algorithms rely on surface texture to establish correspondences [5], [6], [7], [8], [9], [10], [11], [12], [13]. However, when an object has little or no texture—such as human skin or a piece of white paper—these methods all fall short. Some recent works train neural networks [14], [15], [16] to predict correspondences on textureless surfaces. Yet these methods are usually specific to a certain dataset. Cross-category generalization remains challenging. Furthermore, learning-based methods all face a chicken-and-egg problem: in order to obtain a dataset for training, one needs ground truth correspondences on textureless objects in the first place. A straightforward solution is to attach markers to explicitly introduce features. However, most markers would change the object's original appearance, making it hard to pair correspondences with the untouched textureless look.

In this paper, we introduce a novel type of "invisible markers" that add features to a surface without changing its appearance under normal lighting conditions. Our makers are made with *fluorescent dyes*, which are only visible under ultraviolet (UV) light, whereas invisible under normal light (in the visible spectrum). To capture the object in its original appearance, as well as with markers, we build a multi-view imaging system with UV lights: under UV lights, the markers would appear, providing rich features for accurate correspondence matching and tracking; under normal lights, the markers become invisible, allowing us to record the object's original appearance. The two types of lights are triggered in an interleaving fashion with a delay of a few milliseconds. In this way, videos with and without markers are synced.

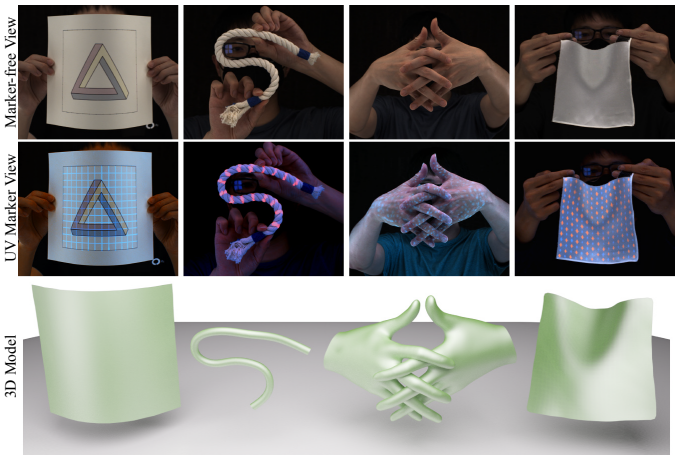


Fig. 1. Sample data of our deformable object tracking dataset (DOT). DOT has about one million video frames, featuring the deformation of various kinds of objects. For each deformation, we provide multi-view video sequences with and without markers, recovered 3D models, and ground truth surface correspondences in both 2D and 3D.

Since the two types of videos are captured from different viewpoints, we transfer the tracked correspondences from videos with marker to the ones without. Correspondences are first matched using the marker videos for 3D reconstruction (among multiple views) and tracking (across time). We then devise a template-based method for registering

- *X. Li is with Tencent America, New York, NY, 10018.*
- *Y. Guo is with George Mason University, Fairfax, VA, 22033.*
- *Y. Tu is with George Mason University, Fairfax, VA, 22033.*
- *Y. Ji is with LightThought LLC, New York, NY, 10018.*
- *Y. Liu is with Columbia University, New York, NY, 10027.*
- *J. Ye is with George Mason University, Fairfax, VA, 22033. E-mail: jinweiye@gmu.edu*
- *C. Zheng is with Columbia University, New York, NY, 10027. E-mail: cxz@cs.columbia.edu*

the tracked correspondences onto the recovered 3D models. Finally, the correspondences are projected back into marker-free videos as ground truth labels.

Using our imaging system and reconstruction approach, we collect a large dataset for deformable object tracking (DOT). Our dataset contains ∼200 deformable motions of four types of objects: rope, paper, cloth, and hands (see examples in Fig. 1). Their original appearance has various levels of textures, ranging from repetitive patterns to little or even no texture. For each motion, we provide 2D videos with and without markers from multiple viewpoints, 3D models of the deformed object, and tracked ground truth correspondences in both 2D and 3D. In total, our dataset has around one million video frames. Experimental results show DOT poses challenges to deformable surface tracking and reconstruction methods. Whereas by training on DOT, network performance significantly improved on weakly textured scenes, as being demonstrated on both our dataset and another deformable surface dataset (DeSurT [13]).

In sum, our main contributions are as follows:

- A novel type of invisible marker and an imaging system that allows simultaneous video acquisition of deformable objects with and without markers.
- A template-based algorithm for 3D reconstruction and transferring correspondences from marker view to marker-free views.
- A large dataset, DOT, for deformable object tracking with ground truth 2D and 3D correspondences.

2 Related Work

Invisible light imaging. The idea of invisible light imaging (e.g., in infrared > 750 nm or ultraviolet < 380 nm spectrum) has been widely explored in computational imaging. Many combine color images with near-infrared (NIR) images for denoising [17], [18], deblurring [19], [20], super-resolution [21], [22] and geometry estimation [23], [24], [25]. Notably, Wang *et al*. [26] uses infrared illumination to relight faces, in order to reduce the effect of uneven face color in a video conference setting. Krishnan and Fergus [17] propose the "dark flash" that uses near-infrared (NIR) and near-ultraviolet (NUV) flashlights to replace the dazzling conventional flash. In another vein, Blasinski and Farrell [27] propose using narrow-band multi-spectral flash for color balancing, and Choe *et al*. [24] derive an NIR reflectance model and use NIR images to recover fine-scale surface geometry.

Unlike widely studied NIR imaging, ultraviolet (UV) imaging has received much less attention. In our work, we use the UV fluorescent substance to create invisible markers.

UV fluorescence imaging. UV fluorescence occurs when a substance absorbs short wavelength light (such as UV light), and re-emits light at a longer wavelength in visible spectrum [28], [29]. This phenomenon has a wide range of imaging applications, including forensics [30], biomedical imaging [31], [32], and material analysis [33], [34]. In computer vision, the fluorescent reflectance has been studied for shape reconstruction [35], immersive range scanning [36], inter-reflection removal [37], multi-spectral reflectance estimation [38], and material classification [39], to name a few. Many prior works analyze the spectral response of fluorescent reflectance, leading to techniques for appearance separation [40], [41], fluorescent relighting [42], [43], and camera spectral sensitivity estimation [44].

In contrast to existing works, we use UV fluorescent markers to enrich surface features, without changing the object's appearance under normal lighting. We also design an imaging system that simultaneously captures video with and without fluorescent markers via time multiplexing.

Some prior works use UV fluorescent markers for other applications, including medical imaging [45], robotics [46], user interface design [47], and augmented reality [48]. Whereas we use the invisible markers for deformable object tracking and reconstruction. Our system is of larger scale, which is more challenging to design, build, and calibrate. In addition, we systematically study the spectral response of UV fluorescent dyes to guide our preparation of fluorescent materials. We design marker patterns for different types of objects in order to allow more robust tracking and reconstruction.

Deformable object reconstruction & tracking. Deformable objects are of great interest in computer vision as they are ubiquitous in daily life and their motions are often complex. Various sensor configurations have been explored for capturing deformable objects, including the use of single camera [5], [7], [8], multiple cameras [49], [50], [51], and color cameras in tandem with depth cameras [52], [53] and with event cameras [54]. Popular methods utilize local appearance to find feature correspondences and then match 2D image features to a 3D shape template for surface reconstruction and tracking [10], [11], [12], [55]. However, these methods are hampered when the surface has sparse or repetitive textures or has no texture at all. In these cases, the number of feature points is too scarce to establish reliable matching correspondences. When the deformable surface has some but sparse textures, dense pixel-level template matching may be helpful [7], [8], [56], [57]. Nevertheless, none of these methods can handle surfaces with no texture—for example, a piece of white paper.

In recent years, many learning-based methods have been proposed for tracking and reconstructing deformable surfaces from a single view [14], [15], [16], [58]. These methods demonstrate great success in handling challenging cases when an object has repetitive or little texture. A comprehensive dataset is critical for training the neural networks, but datasets on deformable objects are quite limited. Most existing datasets are specific to a certain kind of object (*e.g.*, [14] on clothing and [13] on paper, *etc*.). Furthermore, none of the real-captured datasets provide ground truth surface correspondences. In contrast, our DOT dataset has ∼200 deformable motions of four types of objects (*i.e.*, rope, paper, cloth, and hands). Besides 2D videos with and without markers, we also provide high-quality 3D models and ground truth surface correspondences in 2D and 3D.

3 Our Technique

In this section, we present our method that utilizes invisible markers for deformable object reconstruction and tracking. We first introduce the optical properties of our

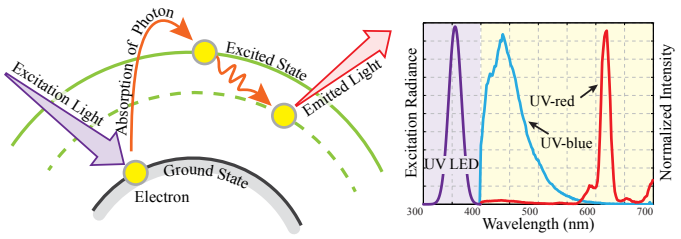


Fig. 2. **(left)** Illustration of the physics of fluorescence. **(right)** The absorption and emission spectra of the fluorescent dyes used for our invisible markers. We use two types of dyes with the peak emission in blue (UV-blue) and red (UV-red). The purple curve shows the spectral profile of our UV light.

Fig. 3. Objects covered with our fluorescent ink (UV-blue and UV-red). **(left)** Image under visible light. **(right)** Image under UV light, where the fluorescent colors become visible.

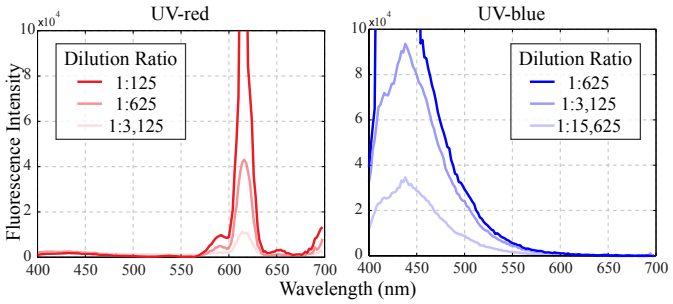


Fig. 4. The spectral response of fluorescent solutions with different concentration levels of the UV-red (left) and UV-blue (right).

markers (Sec. 3.1), and then describe our imaging system for capturing deformable motions with and without markers (Sec. 3.2). Lastly, we present our template-based algorithm for 3D reconstruction and tracking (Sec. 3.3).

3.1 Invisible Markers

We use UV fluorescent dyes to make markers invisible under normal light but visible under UV light. Fluorescent substance exhibits *Stokes Shift* [59]—an optical phenomenon wherein the material absorbs short wavelength light but re-emits light at a longer wavelength. This phenomenon is caused by the material molecules' quantum behavior: when electrons of fluorescent material are irradiated by short wavelength light (*e.g.*, UV light), they enter into an excited state after absorbing the light energy and then immediately de-excite and emit outgoing light at a longer wavelength (in the visible-light spectrum). This principle is illustrated in Fig. 2.

Specifically, we make a fluorescent dye solution and use it as the ink in a fountain pen to draw dot- or line-shaped markers on the object's surface. The resulting markers are invisible under the visible light, thereby preserving the object's original appearance. The markers emit visible light (and thus become visible) only under UV light. Moreover, the emitted light fades immediately—often within 10^{-8}sec—after the UV light is off. Therefore, by using the UV light to trigger the markers' emission and synchronizing the trigger with the camera shutters, we can capture images with and without the markers visible in a time-multiplexed manner (see details in Section 3.2).

We look for fluorescent dyes that satisfy two criteria: 1) the fluorescent emission under UV light has high contrast and strong visibility; and 2) the dyes are biologically safe and non-toxic to human skin. In our experiments, we use the MaxMax UV dyes[1]. For scenes with multiple objects (*e.g.*, hand-object interaction), we use two types of fluorescent dyes: one emits blue light when excited and the other red (they are referred to as *UV-blue* and *UV-red*). In this way, multiple objects in one image can be easily separated by color (see Fig. 3).

Fluorescence detection. Since fluorescent emissions are narrow-banded in wavelength (see Fig. 2), their hue values in fluorescent images usually have small ranges. Exploiting this fact, we can detect markers by using hue values: we convert the images into the HSV color space, and label pixels whose hue values fall into a certain fluorescent dye's emission range. This marker detection in HSV color space is robust even when the surface itself is fluorescent. This is because the narrow-banded fluorescent emission peaks at a specific spectral location, and it is very unlikely that the surface's emission peaks at the same location as our two types of markers.

Dye preparation. To prepare the fluorescent ink for drawing, we dissolve the UV-blue dye in 70% alcohol and UV-red in acetone. Both types of markers can be triggered for emission under 365 nm UV light. We chose this UV light spectrum, because it is in the range of UVA, the safest UV spectrum for human skin, abundant in natural sunlight.

Caution is needed when choosing the solution's concentration level. The higher the concentration is, the brighter the fluorescent emission under the UV light becomes. On the one hand, if the fluorescent emission is too bright, the markers will saturate image pixels, rendering the marker detection based on hue values much harder. On the other hand, if the concentration is too low, the markers appear too dim, and the detection also becomes less robust.

We choose the concentration level through systematic measurements. We test both UV-blue and UV-red dyes. For each type of dyes, we prepare the fluorescent solution of different concentrations spaced by $1/5$ dilution ratio (i.e., $1/5$, $1/25$, $1/125$, ...), covering concentration levels from $1/5$ to $1/15625$. We then measure the spectral responses of each sample using a modular multimode microplate reader (BioTek Synergy H1). Under 365 nm UV light, the measurement records the response of fluorescent emission in the range of 400 nm to 700 nm (visible light spectrum) with a 2 nm step. A subset of the measured response curves are shown in Fig. 4. Through the measurements, we find

1. https://maxmax.com/phosphorsdyesandinks

that for both types of dyes, the desirable concentration level is $1/625$, and we thereafter use this ratio in experiments. Please see the supplementary material for more details about our fluorescent dyes, including their spectral responses on different types of materials and multi-object separation example.

UV safety. Since the UVA spectrum is abundant in sunlight, short-term and low-dose exposure to UVA light is harmless to human skin and eyes. In our experiments, we strictly follow the Threshold Limit Values and Biological Exposure Indices (TLV/BEI) guidelines [60] to limit the UV illumination. Specifically, when capturing human targets in our system, we limit an imaging session to be under 10 minutes, during which the UV lights are turned on for only 36 seconds. Please see the supplementary material for more detailed discussions about UV safety.

3.2 Imaging System

We design and build a novel imaging system for capturing videos of deformable objects with and without markers, by leveraging UV fluorescence.

System configuration. Our imaging system consists of 42 global-shutter color cameras each with a 16 mm lens and 60 UV LED light units. All cameras and lights are uniformly mounted on a rhombicuboctahedral rig frame, facing inward to the frame center. Fig. 5 shows a conceptual illustration and the real physical setup of our acquisition system. Since the rhombicuboctahedral frame is a discrete approximation of a sphere, distances from the cameras and lights to the center of the frame (where the deformable object is located) are about the same (~ 75 cm), which avoids uneven light attenuation. Please see the supplementary material for more details about the specs of cameras and light sources, as well as the camera calibration procedure.

Trigger scheme. Videos of the deformable object with and without markers are captured in a time-multiplexed fashion. To this end, we group the cameras into two sets: one set is triggered when the UV lights are turned on (referred to as *UV cameras*[2]), and the other set is triggered when the UV lights are off (referred to as *reference cameras*). In this way, the UV cameras capture the object with markers, while the reference cameras capture its original untouched appearance. In practice, we use 33 UV cameras and 9 reference cameras for high-quality tracking and 3D reconstruction.

All cameras capture videos at a frame rate of 60 fps. The time interval between two consecutive frames is therefore 16 ms. Since this interval is much larger than the camera's exposure time, we can arrange the exposure period of the two sets of cameras within the 16 ms window back-to-back with no overlap. Our triggering scheme is illustrated in Fig. 5. We custom-build the FPGA control board for syncing and triggering the cameras and lights.

In practice, we use 2 ms exposure time for all cameras. The delay between the videos with and without markers is therefore 2 ms. Although the time difference is very short, we still use interpolation to reduce the amount of possible misalignment. The influence of the delay is studied in Sec. 5.1.

2. These cameras are still regular cameras sensitive to visible light. Here "UV" means that they capture images under UV lights.

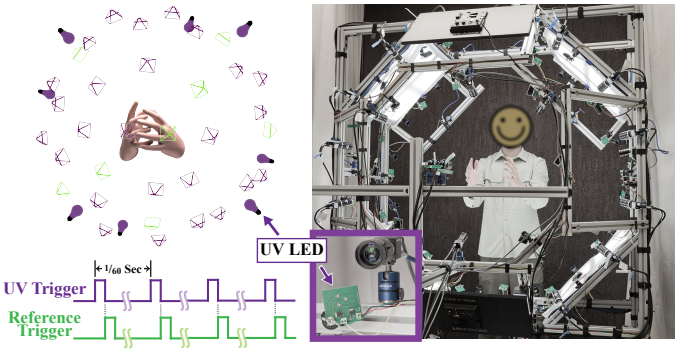


Fig. 5. **(left)** Conceptual illustration of our system with trigger scheme (green color refers to reference cameras, and purple color refers to UV cameras). **(right)** The real physical setup of our system with a zoom-in view of the UV LED unit.

3.3 Reconstruction and Tracking

Given the videos with and without markers, we first detect surface correspondences and perform template-based 3D reconstruction using the video frames that capture markers. We then apply the tracked correspondences to marker-free videos as ground-truth labels. Fig. 6 shows our pipeline.

2D marker detection & 3D reconstruction. To detect markers in UV fluorescent images (captured under UV lighting, in which markers appear), we convert the images into the HSV color space. The fluorescent reflectance typically exhibits high Saturation (S) and Value (V) values, and their Hue (H) values fall into a small range depending on the dye's emission profile (*e.g.*, $\mathrm{H} \in [0, 15]$ for UV-red and $\mathrm{H} \in [110, 125]$ for UV-blue). We therefore threshold the H value to detect marker pixels. These markers are used as features for 3D reconstruction and temporal tracking.

Any 3D photogrammetry-based reconstruction algorithm (*e.g.*, COLMAP [61], [62], Meshroom [63]) can be used on our multi-view marker images to obtain a 3D point cloud for each frame of a motion sequence. In our experiment, we first compute dense disparity maps through semi-global matching [64], and then project the depth maps to point cloud. We also perform cross-view validation to enhance the reconstruction accuracy.

Template fitting. Next, we fit each 3D point cloud to an object-dependent predefined 3D template and project the 2D feature correspondences onto the 3D model. We use different templates for different types of object (see details about the templates in Sec. 4). Here we present our fitting algorithm in general that is applicable to all data.

Consider a 3D point cloud $\mathcal{S}$ consisting of M points $\mathbf{p} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_M\}$ and a 3D shape template described by N vertices $\mathbf{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N\}$ and K faces. Our goal is to deform the shape template so that it aligns closely to the 3D point cloud. We adopt the embedded deformation graph [65] to deform the template: for every vertex i on the template shape, its deformed position is described by $\boldsymbol{v}_i + \boldsymbol{t}_i$. To ensure deformation smoothness, every vertex i also has a local region of influence. Its influence is described by a rotational matrix $\boldsymbol{R}_i \in \mathrm{SO}(3)$, which maps any point $\boldsymbol{p}$ in its local region to the position $\boldsymbol{p}'$ according to

$$\boldsymbol{p}' = \boldsymbol{R}_i(\boldsymbol{p} - \boldsymbol{v}_i) + \boldsymbol{v}_i + \boldsymbol{t}_i. \quad (1)$$

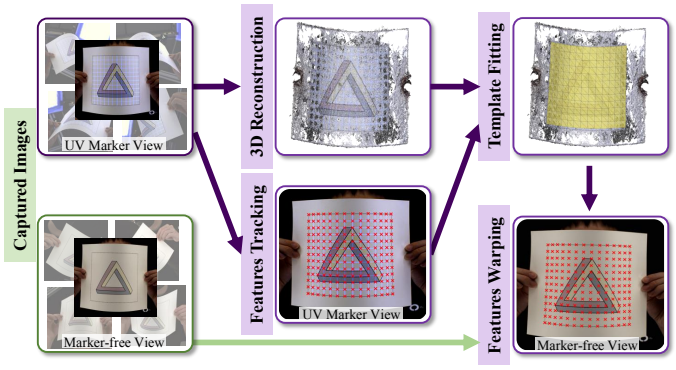


Fig. 6. Warping features from marker view (under UV lighting) to maker-free view (under normal lighting).

We determine $\boldsymbol{t}_i$ and $\boldsymbol{R}_i$ (for $i = 1..N$) by solving the following optimization problem:

$$E_{\text{total}} = E_{\text{fit}} + \lambda_{\text{m}} E_{\text{marker}} + \lambda_{\text{s}} E_{\text{smooth}}. \tag{2}$$

The first term E_{fit} measures the ℓ_2 distance between the 3D point cloud and the deformed template mesh in two ways: for each point j in the point cloud $\mathcal{S}$, its distance to the closest vertex and the closest face on the template. E_{fit} is thus a summation of two terms, namely,

$$\begin{aligned} E_{\text{fit}} = &\sum_{j=1}^{M} \underbrace{\|\boldsymbol{v}_{c(i)} + \mathbf{t}_{c(j)} - \boldsymbol{p}_j\|_2^2}_{\text{point-to-vertex distance}} + \\ &\beta \sum_{j=1}^{M} \underbrace{\|\boldsymbol{n}_{c(i)}^{\top}(\boldsymbol{v}_{c(j)} + \mathbf{t}_{c(i)} - \boldsymbol{p}_j)\|_2^2}_{\text{point-to-face distance}}, \end{aligned} \tag{3}$$

where $c(j)$ indicates the index of the deformed template vertex closet to the point $\boldsymbol{p}_j$, $\boldsymbol{n}_{c(j)}$ denotes the vertex normal, and β is a weight for balancing the two terms.

The second term E_{marker} measures the distance between the detected 2D markers and the projected marker positions from the template mesh. Consider N_f markers. Their 3D positions on the undeformed template mesh, denoted by $\boldsymbol{x}_j$ for $j = 1...N_f$, are initialized at the beginning of the capture session. $\boldsymbol{x}_j$ can be expressed using the barycentric coordinate $\boldsymbol{\alpha}_j = [\alpha_{j,1}\ \alpha_{j,2}\ \alpha_{j,3}]$ on the template triangle where it is located, that is, $\boldsymbol{x}_j = \sum_{k=1}^{3} \alpha_{j,k} \boldsymbol{v}_{j,k}$, where $\boldsymbol{v}_{j,k}$ $(k = 1, 2, 3)$ are the vertex positions of the template triangle. With these notations, E_{marker} is defined as

$$E_{\text{marker}} = \sum_{i=1}^{N_v} \sum_{j=1}^{N_f} w_{ij} \|\boldsymbol{p}_{ij} - \boldsymbol{\pi}_i \tilde{\boldsymbol{x}}_j\|_2^2, \tag{4}$$

where $\tilde{\boldsymbol{x}}_j$ is the j-th marker's 3D position on the deformed template mesh (i.e., $\boldsymbol{x}_j = \sum_{k=1}^{3} \alpha_{j,k}(\boldsymbol{v}_{j,k} + \boldsymbol{t}_{j,k})$), N_v is the number of UV camera views; w_{ij} is the confidence weight for the marker j being viewed from the i-th camera. If the marker j is occluded from the i-th camera, w_{ij} vanishes. Moreover, $\boldsymbol{p}_{ij}$ is the 2D position of a marker j (if not occluded) on the i-th camera view, and $\boldsymbol{\pi}_i$ is the projection matrix of the i-th camera.

The last term E_{smooth} regulates the smoothness of the template mesh's deformation. This is where the local influence of each vertex i (and hence $\boldsymbol{R}_i$) is involved. Following a similar term defined in [65] (called E_{reg} therein), E_{smooth} encourages the mesh deformation to be locally rigid, defined as

$$E_{\text{smooth}} = \sum_{j=1}^{N} \sum_{k \in \mathcal{N}(j)} \gamma_{jk} \|\boldsymbol{R}_j \boldsymbol{v}_{kj} - \boldsymbol{v}_{kj} + \boldsymbol{t}_j - \boldsymbol{t}_k\|_2^2, \tag{5}$$

where $\boldsymbol{v}_{kj}$ is a shorthand for $\boldsymbol{v}_{kj} = \boldsymbol{v}_k - \boldsymbol{v}_j$; $\mathcal{N}(j)$ is the neighboring vertices of vertex i (here defined as the 10-nearest neighbors of i); and γ_{jk} is a weight parameter determined by distance between $\mathbf{v}_j$ and $\mathbf{v}_k$.

When solving the optimization problem (2), we express each $\boldsymbol{R}_i$ in Lie algebra $\mathrm{SO}(3)$ and use the Levenberg-Marquardt optimizer. The optimization starts with large λ_{s} and λ_{m} values, and gradually reduces them in iterations until the optimization converges.

Feature warping. Finally, we project 3D marker features back to marker-free reference views, in order to label ground truth surface correspondences on the object's original appearance. Note that the features cannot be differently warped from marker view to marker-free view in 2D as their depths are not known. This allows us to pair deformable objects with little or no textures with their ground truth surface correspondences.

Recall that our videos with and without markers have a few millisecond delays. In cases when the motion is slow, this delay is small enough to be ignored. However, when an object moves too fast, the delay may cause a noticeable misalignment between the projected features and the reference frame. We alleviate the misalignment by linearly interpolating the 3D models and feature points of two consecutive frames to the time instant at which the reference frame is captured. This strategy, albeit simple, is effective in reducing the misalignment.

4 Deformable Object Tracking Dataset

Using our system, we collect a large dataset for deformable object tracking, which we refer to as DOT. Our dataset contains deformable motions of four types of objects: rope, paper, cloth, and hand. Original appearance of these objects has different levels of textures, ranging from repetitive texture to little or no texture (see examples in Fig. 7). We draw different fluorescent patterns on the objects, introducing rich features for correspondence matching under UV lighting. In total, we have $\sim$ 200 deformable motion sequences. Each sequence has multi-view videos with and without markers, per-frame 3D models and point clouds, and ground truth correspondences in both 2D and 3D. The total number of maker-free video frames is around one million. The details of our DOT dataset are summarized in Table 1 (see the supplementary material for video data samples of DOT).

Our dataset is versatile for network training due to its data abundancy and diversity. By pairing the marker-free videos with ground-truth correspondences, we can train networks for deformable object tracking. In addition, we provide videos from different viewpoints that can suit the need for different imaging configurations. By pairing either single-view or multi-view marker-free frames with their 3D models, our dataset can be used for training deformable object reconstruction networks. As we have an ample amount of objects with little or no textures, training on our dataset

Fig. 7. Sample scenes in our DOT dataset. Here we show marker-free images under normal lighting.

can improve networks' performance in handling these challenging cases.

Although there are several synthetic datasets on deformable objects (*e.g.*, [66]), our dataset is more advantageous because it is real-captured. It is well known that there is domain gap between rendered images and real-captured ones. Using real-captured dataset for training could achieve higher accuracy even with smaller amount of data. In the following, we provide details about each scene category.

Rope scenes. We capture ropes with different lengths (5″, 10″, and 12″) and thicknesses (1/2″, 1/4″, and 1/8″). The motions we perform include swinging, twisting, shaking, pulling, and stretching. We treat the rope as a 1D object and use connected joints as its template. We attached blue tapes at the two ends of the rope to indicate the start- and end-point of our rope object. As for invisible fluorescent markers, we use UV-red to draw dots on the rope with 0.5″ intervals. Depending on the length of the rope, we use different numbers of joints in the template (*i.e.*, 10, 16, and 18 nodes), and these joints are mapped to the invisible markers as correspondences.

Paper scenes. We use letter-sized non-fluorescent papers, and print different patterns on them to create variations in texture. These variations include rich texture (*e.g.*, texts and random dots), repetitive texture, smooth texture (*e.g.*, tie-dye patterns), weak texture (*e.g.*, sparse line drawings), and no texture (pure white paper). Note that these printed patterns are all visible and are considered as the paper's original appearance. To introduce invisible marker pattern, we draw a 13×15 line grid with UV-blue for all scenes. We use the same grid as their 3D template. The grid intersections are used as correspondences for tracking and reconstruction. We record the deformable motions by hand-twisting the paper.

Cloth scenes. We use white silk cloth of different sizes. All motion sequences in the cloth scenes are fully textureless. Same as the paper scenes, we draw a 2D line grid as invisible patterns on the cloth. The size of the grid is determined by the cloth size. As cloth is more deformable than paper, we use a rigid frame to stretch the cloth when drawing the grid pattern. Our deformable motions include swinging, shaking, blowing, and stretching. The motions are induced by hand manipulation or wind blowing. We also perform motions at different speeds.

TABLE 1
Summary of DOT per category.

Category	Rope	Paper	Cloth	Hand
# of Motion Seq.	60	20	30	90
Seq. Length (time)	10s	30s	10s	5s
Seq. Length (frame)	600	1800	600	300
# of Viewpoints	10	6	6	10
Total # of Frame	360K	216K	108K	270K
Template Type	Joint	Grid	Grid	Mesh

Hand scenes. We include a variety of common hand motions and gestures performed by a single hand or two hands. We also include interactive motions between hand(s) and objects, such as scissors, mugs, dice, and toys. For invisible markers, We use UV-blue to draw random dots on hands (see supplementary material for discussions on dye safety on human skin). For hand-object interactions, we use UV-red to fully cover the object, such that we can use the hue difference to separate hand points and object points. This largely improves the accuracy of template fitting (see the result in the supplementary material). For reconstruction and tracking, we use the MANO [67] model as a 3D template for hand. The invisible markers are registered to the template using the rest pose. In our acquisition, all hand motions start with the rest pose for feature initialization. Since we have 33 UV cameras for capturing images with markers, our 3D reconstruction and tracked correspondences are very accurate and robust, even in case of heavy occlusion (*e.g.*, crossed hands, or hand occluded by object).

5 Experiments

We first perform experiments to evaluate the performance of our imaging system and effectiveness of tracked markers (Sec. 5.1). We then benchmark state-of-the-art algorithms on tracking and reconstruction on our DOT dataset (Sec. 5.2). We also demonstrate the benefit of using DOT for network training (Sec. 5.3).

5.1 System Performance

Influence of delayed trigger. We first study the influence of delayed trigger: how much misalignment between the reference and UV views would be caused by the trigger delay, and how effective our interpolation algorithm is at reducing the misalignment. This analysis is very important as our goal is to leverage the "invisible" makers in the UV view for reconstruction and tracking under the reference view (without markers), and we want to make sure that the features detected and tracked in frames with markers are well synchronized with marker-free frames.

We perform experiments using a board paper with a grid. The grid can be seen in both reference and UV views. We use the grid corners as features for measuring the pixel shift caused by the trigger delay. Sample images of the target are shown in Fig. 8. We illustrate feature points from corresponding reference and UV views (marked in orange and

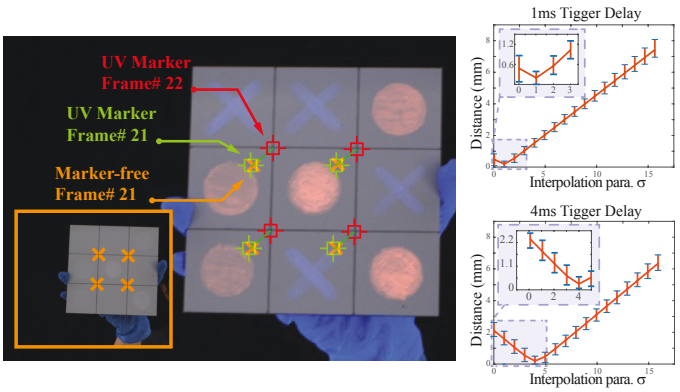


Fig. 8. Influence of trigger delay. **(left)** Sample images of our grid target. We can see apparent motion between neighboring frames (green circles in #21 vs. red circles in #22). For the same frame (#21), features between the marker-free view (orange crosses) and marker view (green circles) are slightly misaligned due to trigger delay. **(right)** Point-to-ray distance curves with respect to the interpolation parameter σ for 1 ms and 4 ms delays.

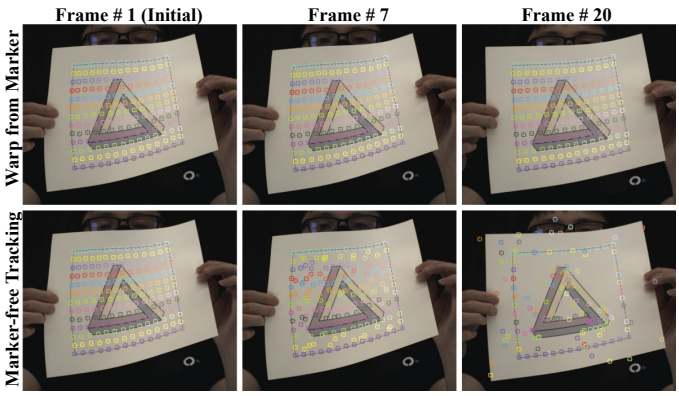


Fig. 9. Feature tracking results with vs. without using fluorescent markers (paper scene $P9$).

TABLE 2
Quantitative comparisons of flow errors on rope, cloth and paper scenes. All errors are reported in pixel unit.

Method	Flow Error (**Rope**)						
	$R14$	$R15$	$R16$	$R17$	$R18$	$R19$	Avg.
GF [68]	0.934	0.529	0.882	0.227	1.990	0.486	0.841
PWC [69]	0.764	0.525	0.779	0.834	1.941	0.703	0.929
RAFT [70]	1.172	0.887	1.052	0.984	2.360	1.176	1.272
OMNI [72]	1.448	0.871	2.138	0.701	2.971	1.166	1.549
DALF [71]	0.793	0.602	0.884	0.731	1.526	0.645	0.864
Method	Flow Error (**Cloth**)						
	$C2$	$C3$	$C4$	$C5$	$C9$	$C10$	Avg.
GF [68]	1.587	0.678	2.231	2.970	2.923	2.935	2.221
PWC [69]	0.907	0.689	1.190	1.210	1.171	1.230	1.066
RAFT [70]	1.006	0.689	1.057	1.298	1.546	1.365	1.160
OMNI [72]	1.001	0.679	1.097	1.380	1.423	1.471	1.175
DALF [71]	4.912	2.165	5.920	6.339	9.627	6.048	5.835
Method	Flow Error* (**Paper**)						
	$P1$	$P2$	$P4$	$P5$	$P7$	$P8$	Avg.
GF [68]	5.429	4.106	6.069	4.747	5.034	3.354	4.662
PWC [69]	8.215	0.564	0.394	0.402	0.362	0.484	0.441
RAFT [70]	196.2	0.625	0.408	0.405	0.374	0.435	0.449
OMNI [72]	1.360	0.954	0.852	0.842	0.860	0.885	0.879
DALF [71]	16.91	5.678	1.833	0.745	0.966	1.061	2.057

*The average errors of paper scenes are calculated without using the $P1$ errors.

green, respectively), as well as those from the consecutive UV view (marked in red). We can see that features from the two consecutive UV images have apparent shifts (whose time interval is 16 ms). The feature misalignment between the corresponding UV and reference views is slight in this case but could vary depending on the object's motion speed (*e.g.*, the faster the speed, the larger the shift).

We then quantitatively measure the amount of misalignment for different trigger delays, and evaluate the effectiveness of the following interpolation scheme. We first compute the 3D coordinates of the feature points via ray triangulation from all UV views. As the reference views are too sparse for accurate triangulation, we only trace out rays from features on the reference view. We use $\mathbf{r}^t$ to denote the rays traced out from the reference view at time t. Assume the 3D points computed by the UV views at time t and time $t+1$ are $\mathbf{v}^t$ and $\mathbf{v}^{t+1}$, respectively, we then linearly interpolate $\mathbf{v}^t$ and $\mathbf{v}^{t+1}$ to obtain an intermediate point $\bar{\mathbf{v}}^\sigma$. We calculate the point-to-ray distance from $\bar{\mathbf{v}}^\sigma$ to $\mathbf{r}^t$ and use it to measure how well the features are aligned in 3D.

We test on two delay values: 1 ms and 4 ms. We compute the point-to-ray distance with respect to various interpolation parameters σ, from 0 to 15. When $\sigma = 0$, it is equivalent as directly using $\mathbf{v}^t$ (*i.e.*, no interpolation). We plot the curves of point-to-ray distance with respect to different σ in Fig. 8. We can see that our interpolation is particularly useful when the trigger delay is large. For example, in the case of 4 ms delay, the misalignment is 2.23 mm without interpolation. Our interpolation brings down the error to 0.28 mm (when $\sigma = 4$).

Tracking results with vs. without markers. In order to show the effectiveness of our fluorescent markers on lack-of-texture surfaces, we compare the correspondence tracking results by using our markers versus without using the markers (*i.e.*, directly apply the tracking algorithm on the marker-free reference view).

Fig. 9 shows comparison results on a paper scene ($P9$). We use the tracking algorithm provided in the commercial software R3DS Wrap4D2. Features are initialized as grid-line corners. We can see that our tracked correspondences computed using the marker view are reliable and robust over time on this paper scene, which lacks texture in large regions. In contrast, if we only use the marker-free video, the algorithm will lose track of most of the feature points within 20 frames (the entire video has 1800 frames). Therefore, with our fluorescent markers, the tracking result is more accurate and robust, with the mismatch rate largely reduced.

5.2 Benchmark Experiments

We benchmark state-of-the-art methods on optical flow, hand reconstruction, and deformable object tracking and reconstruction using our DOT dataset.

Optical flow methods. We test on five state-of-the-art optical flow algorithms: classical variational optical flow as implemented in OpenCV [68] (referred to as "GF"), the two most popular learning-based optical flow networks—PWC-Net [69] and RAFT [70], deformable local feature enhanced optical flow network—DALF [71], and globally consistent motion tracking algorithm—OmniMotion [72].

We perform experiments on rope, cloth, and paper scenes. For the sake of space, here we mainly show quantita-

TABLE 3
Quantitative comparisons of warping errors on cloth and paper scenes.

Method	RMSE on warped image (**Cloth**)						
	$C2$	$C3$	$C4$	$C5$	$C9$	$C10$	Avg.
GF [68]	4.52	1.63	4.82	5.45	7.51	7.76	5.28
PWC [69]	4.75	1.67	5.21	5.33	7.60	9.38	5.66
RAFT [70]	4.98	1.67	5.20	5.51	7.56	9.20	5.69
OMNI [72]	4.89	1.65	5.52	5.60	7.71	8.93	5.72
DALF [71]	4.73	2.09	5.10	5.19	7.32	6.47	5.15
Method	RMSE on warped image (**Paper**)						
	$P1$	$P2$	$P4$	$P5$	$P7$	$P8$	Avg.
GF [68]	8.96	11.2	7.91	10.5	11.0	11.7	10.2
PWC [69]	13.8	6.89	4.34	4.85	4.59	6.38	6.89
RAFT [70]	104*	7.40	4.59	5.61	5.36	6.12	5.87
OMNI [72]	7.40	10.2	5.10	6.63	6.12	9.18	7.40
DALF [71]	9.44	14.8	4.59	5.61	4.85	7.40	7.91

*This error is excluded for calculating the average.

tive evaluation results on a subset of scenes in each category. Rope scenes all have the same thread texture, but the ropes have different lengths and thicknesses. Cloth scenes are all pure white silk cloth with various motions. Paper scenes have different variations of textures: $P1$ has no texture (*i.e.*, white paper); $P2$ has weak textures and large blank regions; $P4$, $P5$, and $P7$ has smooth textures with no sharp edges; $P8$ has a repetitive pattern (see Fig. 7 for these scenes).

For rope scene, we only use the flow vectors of their 1D node joints. For both paper and cloth scenes, we interpolate per-pixel dense flow from the optical flows at grid corners. Specifically, our dataset provides ground-truth 2D and 3D coordinates for their grid templates, from which we can compute the ground-truth 2D and 3D optical flows at grid corners. We run these algorithms on the marker-free video captured from a frontal view. We compute the mean square errors between the estimated optical flow vectors and our ground truth ones. The errors are reported in Table 2 for all three categories. For each scene, the error is averaged on the entire motion sequence. The average error in the last column is computed using all available scenes. We can see our cloth scenes and the white paper scene ($P1$) pose challenges to all methods. Since $P1$ errors are too large, we exclude them for computing the average errors.

We visualize the optical flow estimation results by using the dense optical flow to warp the deformed frames. Results on several challenging paper and cloth scenes are shown in Fig. 10 and 11, respectively. We also show the accumulated flow trajectory of 11 frames. We can see that most algorithms have large errors in textureless regions.

We further verified the accuracy of the warped images based on the grid corners. For each frame, we warp the image using the estimated flows on grid corners. Since the ground-truth warped frame image should be the next frame, we then compare the warped image to the image of the next frame and calculate the averaged root mean squared error (RMSE) for all frames in the motion sequence. Results on several cloth and paper scenes are reported in Table 3.

Hand reconstruction methods. We test on four state-of-the-art template-based hand reconstruction methods: InterNet [73], IntagHand [74], InterWild [75], and DIR [76]. All these methods take a single RGB image as input and predict the corresponding hand template using the MANO model [67]. All methods are trained on InterHand2.6M [73].

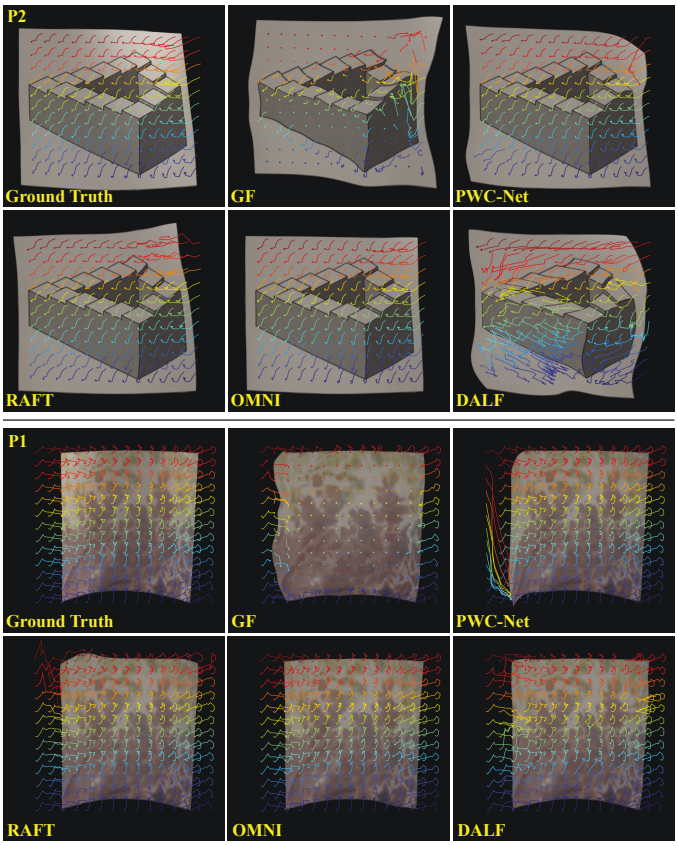


Fig. 10. Qualitative comparisons on warped images with accumulated flow trajectory (paper scene $P1$ and $P2$).

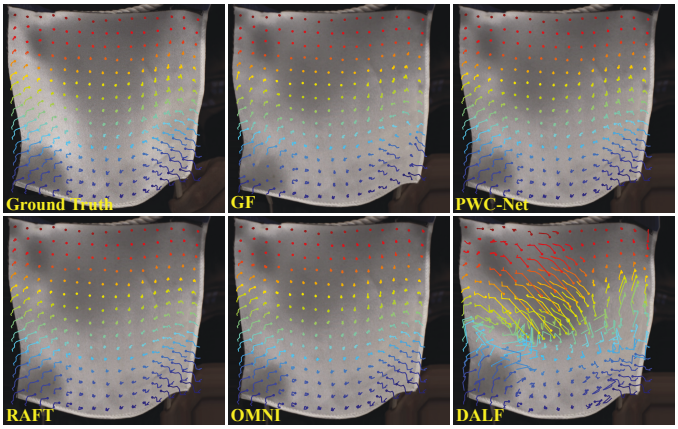


Fig. 11. Qualitative comparisons on warped images with accumulated flow trajectory (cloth scene $C4$).

We use our front-view marker-free images as input to these methods. We project the 3D reconstruction back to the original 2D image plane and compute the per-pixel error on the wrapped image. Results on six challenging hand scenes are reported in Table 4.

Deformable object tracking and reconstruction. We test on four template-based deformable object tracking methods: DDD [77], Graph-Matching [13], RoBuSfT [78], and IsMo-GAN [58]. The first three are optimization-based methods, and the last one is learning-based. These methods take in the image and template of a reference frame and output a

TABLE 4
Quantitative comparisons of hand reconstruction errors.

Method	RMSE on warped image (**Hand**)						
	$H1$	$H2$	$H3$	$H4$	$H5$	$H6$	Avg.
InterNet [73]	42.46	28.23	82.89	41.92	31.13	50.39	41.92
IntagHand [74]	34.88	27.02	64.09	28.30	30.91	54.64	34.89
InterWild [75]	24.01	18.80	33.61	14.75	14.73	18.07	17.75
DIR [76]	31.57	27.73	50.42	23.32	28.44	44.88	32.16
Method	Max error on warped image (**Hand**)						
	$H1$	$H2$	$H3$	$H4$	$H5$	$H6$	Avg.
InterNet [73]	111.32	69.42	166.00	123.41	95.51	124.27	97.01
IntagHand [74]	79.25	88.97	212.81	109.85	93.27	262.17	135.57
InterWild [75]	67.10	61.14	116.05	40.49	47.98	63.62	62.80
DIR [76]	96.39	84.66	153.23	132.78	79.25	105.44	103.16

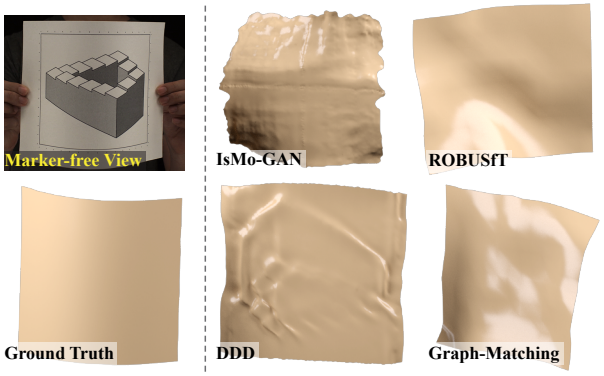


Fig. 12. Qualitative comparison on template estimation. The ground truth is reconstructed using our approach and provided in the dataset.

deformed template for a target frame. We test these methods on marker-free views, evaluating the accuracy of the estimated template by computing vertex-to-vertex distance error against our ground truth 3D model. The errors are reported in Table 5. Note that the results of IsMo-GAN cannot be evaluated in this way as their templates cannot be aligned with ours and have different scales. We visualize estimated deformed templates for one frame in $P2$ (see Fig. 12). We can see that all these methods suffer from large errors on weakly textured paper scene.

5.3 DOT Fine-tuned Network Results

Finally, we demonstrate the benefit of our DOT dataset on network training. Specifically, we test on optical flow and 3D template estimation. For optical flow, we use RAFT pre-trained on Flying Chairs [79]. We then fine tune the pre-trained RAFT on our paper scenes. We apply the fine-tuned RAFT on $P2$ in DOT (not used in training), as well as data from the DeSurT dataset [13], which provides a variety of challenging deformable surfaces with ground truth meshes. Dense optical flow results are shown in Fig. 14. Here we compare the results of fine-tuned RAFT against the pre-trained RAFT. The ground truth flow maps are interpolated from the sparse flow at grid points. We also report the average flow errors. We can see that the accuracy of fine-tuned RAFT is significantly higher on all cases.

We then combine the fine-tuned flow correspondences with RoBuSfT [78] for 3D template estimation. Since RoBuSfT uses SIFT for feature mapping by default, it fails at textureless regions. Whereas the optical flow fine-tuned on

TABLE 5
Quantitative comparisons on template reconstruction errors.

Method	Vertex-to-Vertex Distance Error					
	$P1$	$P2$	$P4$	$P5$	$P7$	Avg.
DDD [77]	25.22	15.14	28.50	24.28	23.75	23.44
Graph-Match [13]	379.77	115.32	95.35	18.48	24.16	126.62
RoBuSfT [78]	NAN	60.31	NAN	2.64	NAN	31.48

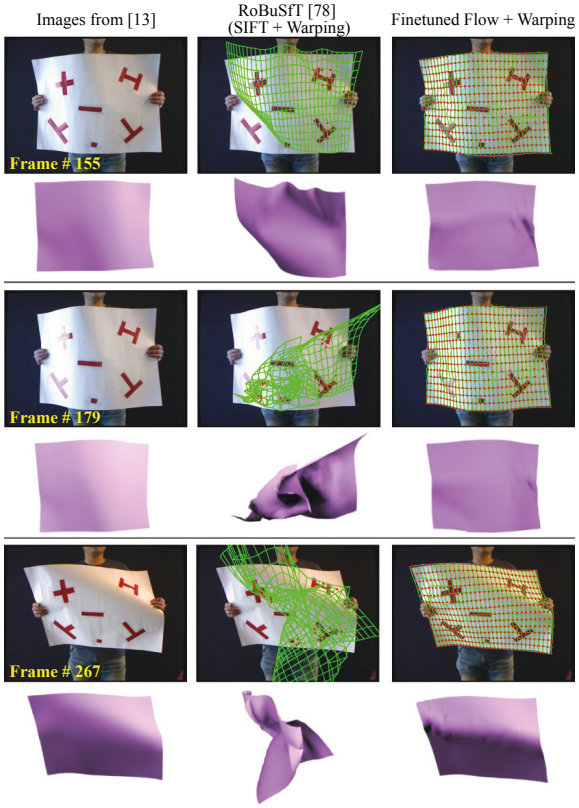


Fig. 13. Qualitative comparisons on template reconstruction with vs. without using our fine-tuned flow.

DOT is able to provide reliable correspondences regardless of textures, which allows accurate warping and template estimation. Comparison results on template reconstruction are shown in Fig. 13.

6 Conclusions

In summary, we demonstrated a solution that uses invisible fluorescent markers for tracking and reconstructing deformable object with little texture. In contrast to existing methods, we are able to simultaneous capture videos of deformable object with and without markers. Videos with markers are used for accurate 3D reconstruction and feature tracking. Tracked correspondences can be transferred to the marker-free videos as ground truth labels. We collected a large deformable motion dataset, DOT, with 200 motion sequences and 1M video frames. DOT provides diverse forms of data, including multi-view videos with and without markers, 3D models and point clouds, and ground truth correspondences in both 2D and 3D. It can be used for benchmarking, or training networks for improved accuracy and robustness on textureless scenes.

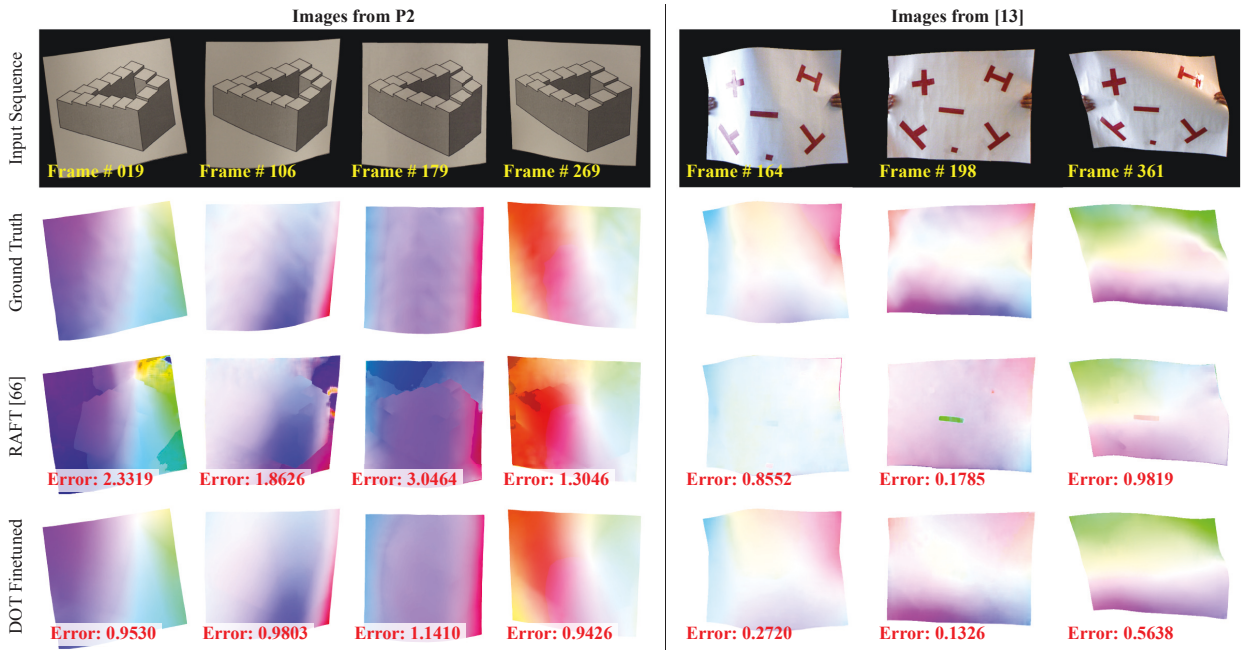


Fig. 14. Qualitative comparisons on dense optical flow (pre-trained RAFT vs. fine-tuned RAFT). **(left)** $P2$ in DOT. **(right)** Data from DeSurT [13].

One viable future direction is to increase the diversity of our data in terms of lighting conditions. We will build a portable system and capture images under various in-the-wild environment. In this way, we are able to gather images with lighting variations, which increases the data diversity.

Acknowledgments

Yu Guo, Yubei Tu, and Jinwei Ye are partially supported by NSF awards 2225948 and 2332542.

References

[1] N. Hasler, M. Asbach, B. Rosenhahn, J.-R. Ohm, H.-P. Seidel, L. Kobbelt, T. Kuhlen, T. Aach, and R. Westermann, "Physically Based Tracking of Cloth," in *Proceedings of the11th International Fall Workshop on Vision, Modeling, and Visualization (VMV)*, 2006. 1

[2] R. White, K. Crane, and D. A. Forsyth, "Capturing and Animating Occluded Cloth," *ACM Transactions on Graphics*, vol. 26, no. 3, July 2007. 1

[3] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[4] J. Wang, F. Mueller, F. Bernard, S. Sorli, O. Sotnychenko, N. Qian, M. A. Otaduy, D. Casas, and C. Theobalt, "RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video," *ACM Transactions on Graphics*, vol. 39, no. 6, November 2020. 1

[5] T. Collins and A. Bartoli, "Realtime Shape-from-Template: System and Applications," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015. 1, 2

[6] V. Gay-Bellile, A. Bartoli, and P. Sayd, "Direct Estimation of Nonrigid Registrations with Image-Based Self-Occlusion Reasoning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 87–104, 2010. 1

[7] D. T. Ngo, S. Park, A. Jorstad, A. Crivellaro, C. D. Yoo, and P. Fua, "Dense Image Registration and Deformable Surface Reconstruction in Presence of Occlusions and Minimal Texture," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1, 2

[8] Q. Liu-Yin, R. Yu, L. Agapito, A. Fitzgibbon, and C. Russell, "Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. 1, 2

[9] D. Pizarro and A. Bartoli, "Feature-Based Deformable Surface Detection with Self-Occlusion Reasoning," *International Journal of Computer Vision*, vol. 97, pp. 54–70, 01 2010. 1

[10] A. Bartoli and T. Collins, "Template-Based Isometric Deformable 3D Reconstruction with Sampling-Based Focal Length Self-Calibration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 1, 2

[11] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro, "Shape-from-Template," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2099–2118, 2015. 1, 2

[12] D. T. Ngo, J. Östlund, and P. Fua, "Template-Based Monocular 3D Shape Recovery Using Laplacian Meshes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 172–187, 2016. 1, 2

[13] T. Wang, H. Ling, C. Lang, S. Feng, and X. Hou, "Deformable Surface Tracking by Graph Matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 8, 9, 10

[14] J. Bednařík, P. Fua, and M. Salzmann, "Learning to Reconstruct Texture-Less Deformable Surfaces from a Single View," in *Proceedings of the International Conference on 3D Vision (3DV)*, September 2018. 1, 2

[15] R. Danźřek, E. Dibra, C. Öztireli, R. Ziegler, and M. Gross, "DeepGarment: 3D Garment Shape Estimation from a Single Image," *Computer Graphics Forum*, vol. 36, no. 2, pp. 269–280, May 2017. 1, 2

[16] A. Tsoli and A. A. Argyros, "Patch-Based Reconstruction of a Textureless Deformable 3D Surface from a Single RGB Image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, October 2019. 1, 2

[17] D. Krishnan and R. Fergus, "Dark Flash Photography," *ACM Transactions on Graphics*, vol. 28, no. 3, Jul 2009. 2

[18] J. Xiong, J. Wang, W. Heidrich, and S. Nayar, "Seeing in Extra Darkness Using a Deep-Red Flash," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[19] D. Sugimura, T. Mikami, H. Yamashita, and T. Hamamoto, "Enhancing Color Images of Extremely Low Light Scenes Based on RGB/NIR Images Acquisition With Different Exposure Times," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3586–3597, 2015. 2

[20] H. Yamashita, D. Sugimura, and T. Hamamoto, "RGB-NIR Imaging with Exposure Bracketing for Joint Denoising and Deblurring of Low-Light Color Images," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 2

[21] H. Tang, X. Zhang, S. Zhuo, F. Chen, K. N. Kutulakos, and L. Shen, "High Resolution Photography with an RGB-Infrared Camera," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, 2015. 2

[22] T. Honda, T. Hamamoto, and D. Sugimura, "Low-Light Color Image Super-Resolution Using RGB/NIR Sensor," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018. 2

[23] G. Choe, J. Park, Y.-W. Tai, and I. S. Kweon, "Exploiting Shading Cues in Kinect IR Images for Geometry Refinement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[24] G. Choe, S. G. Narasimhan, and I. S. Kweon, "Simultaneous Estimation of Near IR BRDF and Fine-Scale Surface Geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[25] Z. Xia, J. Lawrence, and S. Achar, "A Dark Flash Normal Camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2

[26] O. Wang, J. Davis, E. Chuang, I. Rickard, K. De Mesa, and C. Dave, "Video Relighting Using Infrared Illumination," *Computer Graphics Forum*, 2008. 2

[27] H. Blasinski and J. Farrell, "Computational Multispectral Flash," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, 2017. 2

[28] J. R. Lakowicz, *Principles of Fluorescence Spectroscopy*. Springer, 2006. 2

[29] I. Sato, T. Okabe, and Y. Sato, "Bispectral Photometric Stereo Based on Fluorescence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012. 2

[30] A. Richards and R. Leintz, "Forensic Reflected Ultraviolet Imaging," *Journal of Forensic Identification*, vol. 63, pp. 46–69, 01 2013. 2

[31] M. E. Klijn and J. Hubbuch, "Application of ultraviolet, visible, and infrared light imaging in protein-based biopharmaceutical formulation characterization and development studies," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 165, pp. 319–336, 2021. 2

[32] D. C. Gray, W. Merigan, J. I. Wolfing, B. P. Gee, J. Porter, A. Dubra, T. H. Twietmeyer, K. Ahmad, R. Tumbar, F. Reinholz, and D. R. Williams, "In vivo fluorescence imaging of primate retinal ganglion cells and retinal pigment epithelial cells," *Optics Express*, vol. 14, no. 16, pp. 7144–7158, Aug 2006. 2

[33] T. Yokoi, K. Suzuki, and K. Oba, "Ultraviolet Light Imaging Technology and Applications," in *Electron Image Tubes and Image Intensifiers II*, vol. 1449, International Society for Optics and Photonics. SPIE, 1991, pp. 30 – 39. 2

[34] G. Verri, C. Clementi, D. Comelli, S. Cather, and F. Piqueé, "Correction of Ultraviolet-Induced Fluorescence Spectra for the Examination of Polychromy," *Applied Spectroscopy*, vol. 62, pp. 1295–1302, 2008. 2

[35] T. Treibitz, Z. Murez, B. G. Mitchell, and D. J. Kriegman, "Shape from Fluorescence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 2

[36] M. B. Hullin, M. Fuchs, I. Ihrke, H.-P. Seidel, and H. P. A. Lensch, "Fluorescent Immersion Range Scanning," *ACM Transactions on Graphics*, vol. 27, no. 3, p. 1–10, August 2008. 2

[37] Y. Fu, A. Lam, Y. Matsushita, I. Sato, and Y. Sato, "Interreflection Removal Using Fluorescence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2

[38] H. Blasinski, J. Farrell, and B. Wandell, "Simultaneous Surface Reflectance and Fluorescence Spectra Estimation," *IEEE Transactions on Image Processing*, vol. 29, pp. 8791–8804, 2020. 2

[39] Y. Asano, M. Meguro, C. Wang, A. Lam, Y. Zheng, T. Okabe, and I. Sato, "Coded Illumination and Imaging for Fluorescence Based Classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018. 2

[40] M. Alterman, Y. Y. Schechner, and A. Weiss, "Multiplexed Fluorescence Unmixing," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, 2010. 2

[41] C. Zhang and I. Sato, "Image-Based Separation of Reflective and Fluorescent Components Using Illumination Variant and Invariant Color," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2866–2877, 2013. 2

[42] Y. Fu, A. Lam, I. Sato, T. Okabe, and Y. Sato, "Separating Reflective and Fluorescent Components Using High Frequency Illumination in the Spectral Domain," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[43] A. Lam and I. Sato, "Spectral modeling and relighting of reflective-fluorescent scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2

[44] S. Han, Y. Matsushita, I. Sato, T. Okabe, and Y. Sato, "Camera Spectral Sensitivity Estimation from a Single Image under Unknown Illumination by Using Fluorescence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012. 2

[45] D. Butler, A. Keim, S. Ray, and E. Azim, "Large-scale capture of hidden fluorescent labels for training generalizable markerless motion capture models," *Nature Communications*, vol. 14, September 2023. 2

[46] K. Takahashi and K. Yonekura, "Invisible Marker: Automatic annotation of segmentation masks for object manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2020. 2

[47] M. D. Dogan, R. Garcia-Martin, P. W. Haertel, J. J. O'Keefe, A. Taka, A. Aurora, R. Sanchez-Reillo, and S. Mueller, "BrightMarker: 3D printed fluorescent markers for object tracking," in *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2023. 2

[48] H. Park and J.-I. Park, "Invisible marker based augmented reality system," in *SPIE Visual Communications and Image Processing*, vol. 5960, 2005, pp. 501–508. 2

[49] D. S. Schacter, M. Donnici, E. Nuger, M. Mackay, and B. Benhabib, "A Multi-Camera Active-Vision System for Deformable-Object-Motion Capture," *Journal of Intelligent and Robotic Systems*, vol. 75, pp. 413–441, January 2014. 2

[50] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross, "Multi-Scale Capture of Facial Geometry and Motion," in *Proceedings of ACM SIGGRAPH*, 2007. 2

[51] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur, "Markerless Garment Capture," *ACM Transactions on Graphics*, vol. 27, no. 3, p. 99, 2008. 2

[52] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger, "Real-Time Non-Rigid Reconstruction Using an RGB-D Camera," *ACM Transactions on Graphics*, vol. 33, no. 4, July 2014. 2

[53] A. Božič, M. Zollhöfer, C. Theobalt, and M. Nießner, "DeepDeform: Learning Non-rigid RGB-D Reconstruction with Semi-supervised Data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[54] J. Nehvi, V. Golyanik, F. Mueller, H.-P. Seidel, M. A. Elgharib, and C. Theobalt, "Differentiable Event Stream Simulator for Non-Rigid 3D Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021. 2

[55] A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins, "A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 833–850, 2017. 2

[56] M. Salzmann, R. Urtasun, and P. Fua, "Local Deformation Models for Monocular 3D Shape Recovery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[57] A. Malti, A. Bartoli, and T. Collins, "A Pixel-Based Approach to Template-Based Monocular 3D Reconstruction of Deformable Surfaces," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011. 2

[58] S. Shimada, V. Golyanik, C. Theobalt, and D. Stricker, "IsMo-GAN: Adversarial learning for monocular non-rigid 3d reconstruction," in *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 2, 8

[59] G. G. Stokes, "On the Change of Refrangibility of Light," *Proceedings of The Royal Society of London*, vol. 6, pp. 195–200, January 1854. 3

[60] ACGIH, *2023 TLVs AND BEIs* . American Conference of Governmental Industrial Hygienists, 2023. 4

[61] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[62] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 4

[63] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. D. Lillo, and Y. Lanthony, "Alicevision Meshroom: An open-source 3D reconstruction pipeline," in *Proceedings of the 12th ACM Multimedia Systems Conference*. ACM Press, 2021. 4

[64] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005. 4

[65] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Trans. Graph.*, pp. 80–es, 2007. 4, 5

[66] T. T. B. Z. Yang Li, Hikari Takehara and M. Nießner, "4DComplete: Non-rigid Motion Estimation Beyond the Observable Surface," *IEEE International Conference on Computer Vision (ICCV)*, 2021. 6

[67] J. Romero, D. Tzionas, and M. J. Black, "Embodied Hands: Modeling and Capturing Hands and Bodies Together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, Nov. 2017. 6, 8

[68] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA)*, 2003. 7, 8

[69] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7, 8

[70] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 7, 8

[71] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, "Enhancing deformable local features by jointly learning to detect and describe keypoints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7, 8

[72] Q. Wang, Y.-Y. Chang, R. Cai, Z. Li, B. Hariharan, A. Holynski, and N. Snavely, "Tracking everything everywhere all at once," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 7, 8

[73] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *European Conference on Computer Vision (ECCV)*, 2020. 8, 9

[74] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022. 8, 9

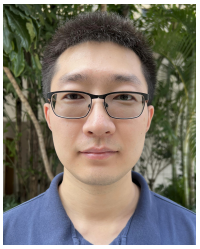
[75] G. Moon, "Bringing inputs to shared domains for 3D interacting hands recovery in the wild," in *CVPR*, 2023. 8, 9

[76] P. Ren, C. Wen, X. Zheng, Z. Xue, H. Sun, Q. Qi, J. Wang, and J. Liao, "Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 8, 9

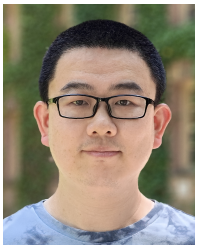
[77] R. Yu, C. Russell, N. D. F. Campbell, and L. Agapito, "Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 918–926. 8, 9

[78] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, A. Bartoli, and E. Ozgur, "Robusft: Robust real-time shape-from-template, a c++ library," 2023. 8, 9

[79] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 9

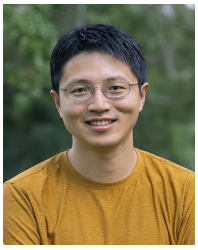


Xinyuan Li received his PhD degree in Applied Mathematics and Statistics from Stony Brook University in 2022. He is now a Senior Graphics Researcher at Tencent America. His research interests include computational photography, computer vision, and computer graphics.



Yu Guo received his PhD degree in Computer Science from the University of California, Irvine in 2021. He is now a researcher at George Mason University and a researcher at Tencent America before that. His research interests include Computer Graphics and Generative AI.

Yubei Tu is a PhD. student in the Department of Computer Science at George Mason University. He received M.Sc. in Computer Science from Illinois Institute of Technology and B.Eng. in Applied Physics from University of Electronic Science and Technology of China. His research interests include 3D reconstruction, light field imaging, and polarimetric imaging.



Yu Ji received his PhD degree in Computer & Information Science from the University of Delaware (UDel) in 2015. He received his M.Sc. in Digital Media from Nanyang Technological University (NTU) in 2010 and Bachelor degree in Electrical Engineering from Huazhong University of Science and Technology (HUST) in 2009. He is now the chief scientist at LightThought. His research interests include computational photography, computer vision, and computer graphics.

Yanchen Liu is a Ph.D. student in the Department of Electrical Engineering at Columbia University. He received M.Sc. in Electrical Engineering from Columbia University and B.Eng. in Electronic Information Science and Technology from Tsinghua University. His research interests include infrared light communication and sensing, mobile IoT and computational photography.

Jinwei Ye is an Associate Professor in the Department of Computer Science at George Mason University. She was an Assistant Professor at Louisiana State University from 2017 to 2021. She received her Ph.D. from the University of Delaware in 2014 and B.Eng. in Electrical Engineering from Huazhong University of Science and Technology in 2009. Before joining academia, she was a researcher in the US Army Research Laboratory and Canon U.S.A., Inc. Her research interests lie in computational photography, computer vision, and computer graphics. She received the NSF CAREER award in 2023. She is a senior member of IEEE.

Changxi Zheng is an Associate Professor at Columbia University. He joined the faculty of Computer Science Department after he received his Ph.D. from Cornell University in 2012. He directs the Columbia's Computer Graphics Group (C2G2) and previously also directed Tencent Americas' Pixel Lab. He works on applied computer science, with a particular focus on computer graphics and scientific computing in general.