Seeing Dynamic Scene in the Dark:  
A High-Quality Video Dataset with Mechatronic Alignment

Ruixing Wang¹,²*  Xiaogang Xu¹*  Chi-Wing Fu¹  Jiangbo Lu²  Bei Yu¹,²  Jiaya Jia¹,²  
¹ The Chinese University of Hong Kong  ² SmartMore  
{rxwang,xgxu,cwfu,byu,leojia}@cse.cuhk.edu.hk, jiangbo@smartmore.com

Abstract

Low-light video enhancement is an important task. Previous work is mostly trained on paired static images or videos. We compile a new dataset formed by our new strategy that contains high-quality spatially-aligned video pairs from dynamic scenes in low- and normal-light conditions. We built it using a mechatronic system to precisely control the dynamics during the video capture process, and further align the video pairs, both spatially and temporally, by identifying the system’s uniform motion stage. Besides the dataset, we propose an end-to-end framework, in which we design a self-supervised strategy to reduce noise, while enhancing the illumination based on the Retinex theory. Extensive experiments based on various metrics and large-scale user study demonstrate the value of our dataset and effectiveness of our method. The dataset and code are available at https://github.com/dvlab-research/SDSD.

1. Introduction

To enhance underexposed images and videos captured in low light is a longstanding task in computer vision. It is challenging since underexposed input does not have much scene structural information. Also, dark areas are typically dominated by noise with low signal-to-noise ratios (see Figure 1(a)). When enhancing such input, one may end up with amplified noise and undesirable visual artifacts in results, as shown in Figure 1(b) & (c). These issues could be exaggerated for videos taken from dynamic scenes, in which the cameras move largely. In this paper, we focus on enhancing underexposed videos taken from low-light dynamic scenes.

Many methods [34, 18, 9, 6, 25, 20, 4] have been proposed to enhance underexposed images/videos based on deep neural networks via supervised learning. Often these methods learn a mapping from images/videos taken in low-light condition to those with normal lighting. They generally do not deal with videos of dynamic scenes or severely-

*Equal Contribution.

![Figure 1: An example frame (a) from a challenging underexposed frame enhanced by a SOTA method (b), a commercial software (c), and our method (d). Our result exhibits clearer details with distinct contrast and less noise.](image-url)
undesirably amplified, leading to various visual artifacts in the enhancement results. In this work, our second goal is to develop a new solution to enhance underexposed videos, taking noise into account.

Our contribution is the following. First, we release a new dataset of 150 high-quality spatially-aligned videos that feature the same dynamic scenes in low- and normal-light conditions. To ensure the alignment and quality of the videos, we built a mechatronic alignment system, in which we assembled an electric slide rail and mounted a professional camera on it; see Figure 2. Using this system, we captured videos of nearly-identical camera motion, thereby reducing the effort needed to align the low- and normal-light videos for temporal and spatial consistency. The constructed dataset is named as SDSD dataset, standing for “Seeing Dynamic Scenes in the Dark.”

Second, we formulate an end-to-end framework for enhancing underexposed videos. We emphasize noise reduction and illumination enhancement simultaneously in our method. For noise reduction, we formulate a self-supervised strategy for learning, while for the illumination enhancement, we predict an illumination map from each input frame based on the Retinex theory [16].

Our dataset is the first high-quality paired video dataset for dynamic scenes, featuring high-resolution video pairs of the same scene and motion in low- and normal-light conditions. Trained on our new dataset, our framework works decently for enhancing underexposed videos, even in extremely low-light conditions. To evaluate and demonstrate the applicability and robustness of our new approach, we conducted comprehensive experiments to compare it with a rich set of state-of-the-art methods on our constructed dataset and SMID dataset [4]. Further, we conducted a large-scale user study with 100 participants, showing that our results are visually more pleasing and accurate than previous methods.

Figure 2: The devices in our mechatronic system. In the top row, from left to right is Canon EOS 6D Mark II, the electric machine (to drive the motion of the camera), the controller (to set the starting and ending points for motion), and an ND filter. We mount the camera and the electric machine on the electric slide rail, as shown in the bottom row.

2. Related Work

2.1. Low-light Image Enhancement and Datasets

To enhance a low-light video, one may apply an image enhancement method in a frame-by-frame manner. Histogram equalization and gamma correction are fundamental tools to increase image contrast and expand the dynamic range. Recently, Retinex-based methods [24, 8, 33, 10, 2, 35] produce impressive results enhancing low-light images.

Learning-based low-light image enhancement methods receive increasing attention in recent years [30, 31, 17, 3]. Wang et al. [23] proposed to enhance underexposed photos by learning the illumination map. Sean et al. [20] learned spatially local filters of three different types to enhance low-light images. Xu et al. [28] proposed a frequency-based decomposition and enhancement model to enhance low-images with a low-light dataset based on SID [5]. Yang et al. [32] presented a semi-supervised learning method to recover a linear band representation of an enhanced image.

Also, unsupervised learning has been explored for photo enhancement [6, 13, 9]. Guo et al. [9] trained a lightweight neural network to estimate pixel-wise and high-order curves for dynamic range adjustment of a given image. However, applying image enhancement algorithms to individual frames likely causes flickering problems.

To improve the enhancement performance, various datasets were built. Bychkovsky et al. [1] compiled the large MIT-Adobe FiveK dataset, in which the photos are paired with expert-retouched results for tone adjustment. Chen et al. [5] collected raw images of short/long exposure pairs with a U-Net to learn a raw image enhancement system. Recently, Wei et al. [27] presented a dataset containing low- and normal-light image pairs and proposed a deep Retinex-Net learned on this dataset.

2.2. Low-light Video Enhancement and Datasets

Zhang et al. [34] presented an approach for underexposed video enhancement using a perception-driven fusion. Lv et al. [18] proposed a multi-branch network to extract features up to different levels, applicable to both image and video domains. Jiang et al. [14] employed a standard CNN to learn enhancement mapping for the transformation from low-light raw camera sensor data to bright videos. However, these methods are not applicable to severe noise conditions.

Xue et al. [29] designed a flow representation tailored for specific video processing tasks. Wang et al. [25] mathematically defined the practical high sensitivity noise in digital cameras and proposed to enhance low-light videos based on the noise model using a recurrent neural network. Chen et al. [4] collected a static dataset of raw low-light videos and learned the low-light to normal-light transformation for videos. Danai et al. [22] provided a data synthesis mecha-
3.1. Capture Video Data

To control the trace of the camera, we set the starting point \( A \) and endpoint \( B \) on the electric slide rail. The camera starts capturing videos at point \( A \), then moves towards point \( B \). To capture a pair of videos, we run the slide rail by two rounds. In the first round, we capture a noise-free bright video with good contrast and vivid color. In the second round, we put the ND filter on the camera lens and increase the camera ISO to capture a low-light video with severe noise.
Alignment of videos was conducted according to the camera trace, which consists of five stages (Figure 4) — that is, static stage I, accelerating stage, uniform moving stage, decelerating stage, and static stage II.

The camera in static stages I and II locates at points A and B, respectively, and has no motion. The accelerating stage leads to the camera with speed accelerating at the beginning of the moving trace and the decelerating stage is at the ending of the moving trace. Aligning frames of the two sequences in the same position is easy in the uniform moving stage, where the camera motion is stable. Thus, we choose the frames in the uniform moving stage to construct our video dataset.

First, we find the first frame in the uniform moving stage from the normal/low-light video. Then we manually pick the aligned frame from the uniform moving stage in a frame-wise style, until finding the dis-alignment frame in the decelerating stage. Specifically, we adopt the reference objects in the top, bottom, left, right of the frames to measure the alignment where the reference objects should locate at the same position for the two aligned frames.

We collected 150 paired video sequences in total, including 80 outdoor videos and 70 indoor videos. Each video consists of 100-300 frames, and the resolution is $1920 \times 1080$. Our dataset is called SDSD, and Figure 3 shows 25% of the data in our dataset and the statistical indicators of the overall dataset. In our dataset, there are various scenes, such as cityscapes, grassland, and indoors. In Figure 5, we provide two examples for indoor/outdoor sequence under low- and normal-light conditions.

### 3.2. Align Video Data

Alignment of videos was conducted according to the camera trace, which consists of five stages (Figure 4) — that is, static stage I, accelerating stage, uniform moving stage, decelerating stage, and static stage II.

The camera in static stages I and II locates at points A and B, respectively, and has no motion. The accelerating stage leads to the camera with speed accelerating at the beginning of the moving trace and the decelerating stage is at the ending of the moving trace. Aligning frames of the two sequences in the same position is easy in the uniform moving stage, where the camera motion is stable. Thus, we choose the frames in the uniform moving stage to construct our video dataset.

First, we find the first frame in the uniform moving stage from the normal/low-light video. Then we manually pick the aligned frame from the uniform moving stage in a frame-wise style, until finding the dis-alignment frame in the decelerating stage. Specifically, we adopt the reference objects in the top, bottom, left, right of the frames to measure the alignment where the reference objects should locate at the same position for the two aligned frames.

We collected 150 paired video sequences in total, including 80 outdoor videos and 70 indoor videos. Each video consists of 100-300 frames, and the resolution is $1920 \times 1080$. Our dataset is called SDSD, and Figure 3 shows 25% of the data in our dataset and the statistical indicators of the overall dataset. In our dataset, there are various scenes, such as cityscapes, grassland, and indoors. In Figure 5, we provide two examples for indoor/outdoor sequence under low- and normal-light conditions.
level and illumination prediction (IMP).

In Figure 6, simultaneously achieving noise reduction and illumination enhancement. It consists of the modules of progressive alignment (PAL), noise estimation (SNE), and illumination prediction (IMP).

Instead of solely enhancing the illumination like traditional methods, we propose an end-to-end network as shown in Figure 6, simultaneously achieving noise reduction and illumination enhancement. This network consists of the modules of progressive alignment (PAL), noise estimation (SNE), and illumination prediction (IMP).

4.1. Progressive Alignment (PAL)

Directly conducting low-light image enhancement [23, 9, 20] to each frame causes flickering. To avoid it and take advantage of temporal information, the input of existing video enhancement methods is a sequence. Meanwhile, to produce frames without blur, existing video enhancement methods consider aligning neighboring frames into the middle one [36, 21, 15]. Such alignment can be executed at the feature level. In this section, we illustrate the process of feature extraction and our progressive strategy for alignment.

Given the input sequence under the low-light condition as \( I_{t+i}, i \in [-2, 2] \) with shape \( \mathbb{R}^{5 \times H \times W \times 3} \), we extract these frame features with three convolution layers and two down-sampling layers, to propagate the information spatially and temporally, as shown in Figures 6 and 7. The obtained features are denoted as \( F_{t+i}^L, i \in [-2, 2] \) with shape \( \mathbb{R}^{5 \times \frac{H}{4} \times \frac{W}{4} \times C} \), where \( C \) is the number of feature channels, and \( L \) is one level in the progressive alignment.

The alignment module spatially aligns features of neighboring frames to the central one, which is realized by the deformable convolution (DCN) [7] progressively, as illustrated in Figure 7. To align \( F_{t+i}^L, i \in [-2, 2] \), we extract features with different levels as \( F_{t+i}^{L+1}, i \in [-2, 2] \) and \( F_{t+i}^{L+2}, i \in [-2, 2] \) that have shape \( \mathbb{R}^{5 \times \frac{H}{4} \times \frac{W}{16} \times C} \) and \( \mathbb{R}^{5 \times \frac{H}{16} \times \frac{W}{16} \times C} \).

We first compute the offset \( \{\Delta p_k\}^{L+2}_{t+i} \) for the DCN in level \( L+2 \). The offset is learned from \( F_{t+i}^{L+2} \) and \( F_{t+i}^{L+2} \), and the aligned feature is obtained with the learned offset as

\[
\{\Delta p_k\}^{L+2}_{t+i} = f^{L+2}(F_{t+i}^{L+2} \odot F_{t+i}^{L+2}),
\]

\[
\bar{F}_{t+i}^{L+2} = g^{L+2}(DCN(F_{t+i}^{L+2}, \{\Delta p_k\}^{L+2}_{t+i})),
\]

where \( \{\Delta p_k\}^{L+2}_{t+i} \) is the learned offset for DCN at \((L+2)\)-th level, \( \odot \) denotes channel concatenation, \( f^{L+2} \) and \( g^{L+2} \) are the mapping function completed by several convolution layers, and \( DCN \) is the operation of DCN. To implement
the progressive learning, we employ the computed offset at \((L + 2)\)-th and \((L + 1)\)-th levels for offset computation at \((L + 1)\)-th and \(L\)-th levels. Further, we set the progressive learning for updating features at each level by incorporating the features from other levels. The process can be written as

\[
\{\Delta p_k\}_{b,i}^{L+1} = f^{L+1}(F_{t+i}^{L+1} \odot F_i^{L+1} \odot \{\Delta p_k\}_{b,i}^{L+1+1}),
\]

\[
\tilde{F}_{t+i}^{L+1} = g^{L+1}(DCN(F_{t+i}^{L+1}, \{\Delta p_k\}_{b,i}^{L+1+1}) \odot \{\tilde{F}_{t+i}^{L+1+1}\}),
\]

where \(\{\Delta p_k\}_{b,i}^{L+1+1}\) is the upsampling offset, \(\tilde{F}_{t+i}^{L+1}\) is the upsampled feature and \(j \in \{0, 1\}\). With the aligned features \(\tilde{F}_{t+i}\), we fuse them with the similarity between \(\tilde{F}_{t+i}\) and \(F_t\). The process to obtain the aligned feature can be denoted as \(F_t = f_a(I_{t+i}), i \in [-2, 2]\), where \(F_t\) has shape of \(\mathbb{R}^{H \times W \times C}\). Such alignment works well for videos with the smooth local motion, since the motion in the corresponding input can be simulated as the translation.

### 4.2. Self-Supervised Noise Estimation (SNE)

After obtaining the aligned feature \(F_t\), we utilize it for two purposes: noise estimation and illumination map prediction. Implementation of noise estimation is described in this section. With the input \(I_{t+i}\), we aim to predict a noise map \(N_t\) with shape as \(\mathbb{R}^{H \times W \times 3}\), and the recovered frame can be obtained as \(I_t = N_t - I_t\). The module of noise estimation is trained with the principle of “Noisier2Noise” [19] where we add the crafted noise to an input noisy/clean frame and train the network to regress the added noise. Thus, the noise estimation module can be learned in a self-supervised way.

As shown in Figure 6, to estimate noise, the aligned feature \(F_t\) forwards through a network \(f_n\) that consists of residual blocks and two layers for up-sampling. This SNE network produces the computed noise map \(N_t\) as \(N_t = f_n(F_t) = f_n(f_a(I_{t+i}), i \in [-2, 2]\). For training, we compute the average RGB value of \(I_{t+i}\) to create noise to be added to \(I_{t+i}\), so that the noise magnitude can be more relevant to image contents of \(I_{t+i}\). We compute the loss as

\[
\bar{N}_{t+i} = I_{t+i} - \mathcal{M}(I_{t+i}), i \in [-2, 2],
\]

\[
\mathcal{L}_n = \mathbb{E}(\|f_n(f_a(I_{t+i} + \bar{N}_{t+i}, i \in [-2, 2]) - \bar{N}_t\|),
\]

where \(\bar{N}_{t+i}\) is the created noise, \(\mathcal{M}(I_{t+i})\) is the average RGB value of \(I_{t+i}\), \(\mathbb{E}\) is the operation to compute the average value, and \(\mathcal{L}_n\) is the loss term for training \(f_n\).

### 4.3. Illumination Map Prediction (IMP)

According to the Retinex-based methods [23], we enhance the illumination of \(I_t\) by predicting an illumination map \(I_t\). Unlike existing Retinex-based methods, we propose to train a noise-aware network for estimating an illumination map. The illumination map should be consistent with frames and not be influenced by the noise.

As shown in Figure 6, we adopt another network \(f_i\) with the input of \(F_t\) to predict the illumination map. This IMP module also consists of residual blocks and two layers for up-sampling. We formulate this process to acquire the illumination map \(S_t\) as \(S_t = f_i(f_a(I_{t+i}, i \in [-2, 2]\))\) and the output size of the illumination map is \(\mathbb{R}^{H \times W \times 3}\). The loss term to train \(f_i\) is written as

\[
\mathcal{L}_{ic} = \mathbb{E}(\|f_i(f_a(I_{t+i}, i \in [-2, 2]) - \bar{I}_t\|),
\]

\[
\mathcal{L}_{in} = \mathbb{E}(\|f_i(f_a(I_{t+i} + \bar{N}_{t+i}, i \in [-2, 2]) - \bar{I}_t\|),
\]

where \(\bar{N}_{t+i}\) is the crafted noise defined in Eq. (3). \(\mathcal{L}_{ic}\) and \(\mathcal{L}_{in}\) are the loss terms for training \(f_i\).

### 4.4. Overall Loss Function

We denote the output of \((I_{t+i}, i \in [-2, 2]\) from our network as \((S_t, N_t) = f(I_{t+i}, i \in [-2, 2]\)), where \(f\) denotes the function implemented by our network. The final enhanced frame can be obtained as \(\bar{I}_t = \frac{I_t - N_t}{S_t}\). To this end, we add a loss function for \(\bar{I}_t\) as the constraint for \(f_a\), \(f_n\) and \(f_i\) simultaneously, which can be written as

\[
\mathcal{L}_b = \mathbb{E}(\|\bar{I}_t - \bar{I}_t\|).
\]

Moreover, to ensure the effect of enhancement with noisy input, we set another constraint for \(f_a\), \(f_n\) and \(f_i\) as

\[
(S'_t, N'_t) = f(I_{t+i} + \bar{N}_{t+i}, i \in [-2, 2]),
\]

\[
\mathcal{L}_{bn} = \mathbb{E}(\|\bar{I}_t + \bar{N}_{t+i} - N'_t - \bar{I}_t\|),
\]

where \(\bar{N}_{t+i}\) is the crafted noise defined in Eq. (3).

The overall loss function to train this framework is summarized as

\[
\mathcal{L}_a = \lambda_1 \mathcal{L}_n + \lambda_2 (\mathcal{L}_{ic} + \mathcal{L}_{in}) + \lambda_3 \mathcal{L}_b + \lambda_4 \mathcal{L}_{bn},
\]

where \(\lambda_1, \lambda_2, \lambda_3, \) and \(\lambda_4\) are weights of the loss terms. We empirically set \(\lambda_1 = 2, \lambda_2 = 0.25, \lambda_3 = 0.5, \) and \(\lambda_4 = 0.5\).

### 5. Experiments

#### 5.1. Experiment Setup

We demonstrate the superiority of our method and the impact of SDSD through experiments in this section. To illustrate the effect of our method, we retrain seven previous representative methods on the SDSD and SMID [4] datasets for comparison and provide an ablation study for our method. Besides, we conduct user study to evaluate the results of our method and the chosen baselines.

Further, we compare the performance of two models with our designed network structure that are trained on SDSD and SMID [4], respectively, and conduct the evaluation on real-world videos captured from mobile devices. For the SMID dataset, we use SMID pre-processing to process the RAW data to produce the SDR data. The comparison between
Table 2: Quantitative comparison among our method, state-of-the-art baselines, and ablation settings on our SDSD and the SMID [4] dataset. PSNR is in dB.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SDSD</th>
<th>SMID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>DeepUPE [23]</td>
<td>21.82 %</td>
<td>0.68</td>
</tr>
<tr>
<td>ZeroDCE [9]</td>
<td>20.06 %</td>
<td>0.61</td>
</tr>
<tr>
<td>DeepLPF [20]</td>
<td>22.48 %</td>
<td>0.66</td>
</tr>
<tr>
<td>DRBN [32]</td>
<td>22.31 %</td>
<td>0.65</td>
</tr>
<tr>
<td>MBLLEN [18]</td>
<td>21.79 %</td>
<td>0.65</td>
</tr>
<tr>
<td>SMID [4]</td>
<td>24.09 %</td>
<td>0.69</td>
</tr>
<tr>
<td>SMOID [14]</td>
<td>23.45 %</td>
<td>0.69</td>
</tr>
<tr>
<td>Ours w/o PAL, w/o IMP, w/o SNE</td>
<td>22.61 %</td>
<td>0.64</td>
</tr>
<tr>
<td>Ours with PAL, w/o IMP, w/o SNE</td>
<td>24.47 %</td>
<td>0.65</td>
</tr>
<tr>
<td>Ours with PAL, with IMP, w/o SNE</td>
<td>24.53 %</td>
<td>0.67</td>
</tr>
<tr>
<td>Ours</td>
<td>24.92 %</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 3: User preference comparison in the user study. “Ours” is the percentage that our result is preferred, “Other” is the percentage that some other method is preferred, “Same” is the percentage that the users have no preference.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Other</th>
<th>Same</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepUPE [23]</td>
<td>27.0%</td>
<td>8.7%</td>
<td>64.3%</td>
</tr>
<tr>
<td>ZeroDCE [9]</td>
<td>11.9%</td>
<td>4.4%</td>
<td>83.7%</td>
</tr>
<tr>
<td>DeepLPF [20]</td>
<td>19.8%</td>
<td>6.8%</td>
<td>73.4%</td>
</tr>
<tr>
<td>DRBN [32]</td>
<td>12.5%</td>
<td>19.2%</td>
<td>68.3%</td>
</tr>
<tr>
<td>MBLLEN [18]</td>
<td>6.8%</td>
<td>13.5%</td>
<td>79.7%</td>
</tr>
<tr>
<td>SMID [4]</td>
<td>13.9%</td>
<td>13.5%</td>
<td>72.6%</td>
</tr>
<tr>
<td>SMOID [4]</td>
<td>19.8%</td>
<td>25.4%</td>
<td>54.8%</td>
</tr>
</tbody>
</table>

5.3. Comparison on the SMID Dataset

To demonstrate the generalization of our method, we evaluate the effect of our method/baselines that are trained on SMID [4]. The testing sequences are in 8bit sRGB format.

In Figure 9, we provide the results of our method and baselines, which are trained on the training set of SMID [4] while evaluated on the testing set of SMID [4]. Our method restores the underexposed video frames into those with normal brightness and natural color. Also, we provide quantitative results in Table 2 for comparison. Our network achieves the best PSNR and SSIM, and performs better than the baselines with a large margin (more than 1.3dB).

5.4. User Study on the Real Testing Videos

To compare our method with the seven baselines based on human perception, we conduct user study with 100 persons using totally 12 videos, which are captured by iPhone7plus and iPhoneX with real camera motion and local subjects motion. We compute the results of different methods on these videos to conduct an AB-test. All network models are trained on the SDSD dataset.

Each participant saw two videos (called videos A and B) simultaneously, which were synthesized by different methods, and has to choose among three options: “Video A is better”, “Video B is better”, and “I cannot make a decision on which one is better”. For evaluation accuracy, we invited 100 persons to participate in our user study, and each participant was asked to complete 14 pairs of AB-test. Each AB-test was conducted between our result and one of the seven baselines — they were presented in a random left-right order. Participants made decisions according to the following five properties: suitable brightness, clear details, distinct contrast, vivid color, and well-preserved photo realism.

The results of this user study are given in Table 3, where we report the proportion that our results are preferred by participants. It proves that our method yields more appealing and natural results, as the participants often preferred our predicted videos rather than those from the baselines.
Figure 8: An underexposed video frame (a) enhanced by various methods (b)-(h). Results from baselines exhibit blurry details, noise, distorted color, weak contrast, abnormal brightness, and unnatural white balance (zoom in to see details).

Figure 9: Another underexposed video frame (a) enhanced by various methods (b)-(h) that are trained on the SMID dataset [4] (zoom in to see details).

Figure 10: A real video frame (a) captured by iPhone X enhanced by our method trained on SMID [4] (c) and our SDSD (d). (b) shows the phone’s camera setting (light blue) and its trajectory (orange) in the 3D-reconstructed scene (please zoom in to see details).

6. Conclusion

We have presented a paired high-quality video dataset built using a mechatronic system. Each video pair in our dataset is captured from an indoor/outdoor dynamic scene, containing two spatially-aligned videos taken from low- and normal-light conditions, respectively. Besides, we propose an end-to-end framework for video enhancement. Our framework achieves noise reduction and illumination enhancement simultaneously. Extensive experiments with user study are conducted, demonstrating the value of our dataset and the effectiveness of our method.

The methods trained with the SDSD dataset effectively enhance videos that are captured with a real camera trajectory, e.g., the panning and rotation motion, as shown in the user study. But they may not perfectly handle videos captured with serious camera shaking. Hence, we envision to build another dataset using a robot arm that can precisely repeat the most challenging trajectories.
References


