Enforcing Temporal Consistency in Video Depth Estimation

Siyuan Li Yue Luo Ye Zhu Xun Zhao Yu Li Ying Shan
Applied Research Center (ARC), Tencent PCG

Figure 1: Two depth estimation results on four frames and the comparisons. Existing video based methods using structure-from-motion (DeepV2D [41]) or sequential model (CLSTM [55]) has limitation in scenes they can apply and their depth estimations are not satisfactory on these two general scenes. The state-of-the-art single image depth prediction method MiDaS [32] is a general one but it generates flickering depth maps as the two sequences are processed in a per-frame manner. Our method is built on single image depth methods but with temporal consistency enforced. We can achieve temporal stable depth estimation by simple frame by frame processing while maintaining high depth quality.

Abstract

Most existing monocular depth estimation methods are trained on single images and have unsatisfactory temporal stability in video prediction. They may rely on post processing to solve this issue. A few video based depth estimation methods use reconstruction framework like structure-from-motion or sequential modeling. These methods have assumptions in the scenarios that they can apply thus limits their real applications. In this work, we present a simple approach for improving temporal consistency in video depth estimation. Specifically, we learn a prior from video data and this prior can be imposed directly into any single image monocular depth method. During testing, our method just performs end-to-end forward inference frame by frame without any sequential module or multi-frame module. In the meanwhile, we propose an evaluation metric that quantitatively measures temporal consistency of video depth predictions. It does not require labelled depth ground truths and only assesses flickering between consecutive frames. Experiments show our method can achieve improved temporal consistency in both standard benchmark and general cases without any post processing and extra computational cost. A subjective study indicates that our proposed metric is consistent with the visual perception of users, and our results with higher consistency scores are indeed preferred. These features make our method a practical video depth estimator to predict dense depth of real scenes and enable several video depth based applications.

1. Introduction

Recovering 3D information from 2D images and videos is always one of the most fundamental tasks in computer vision and has been studied for long. While direct 3D perception is difficult, depth provides an intermediate (2.5D) representation to measure the physical world. Depth perception enables application in a wide variety of scenarios, such as augmented reality (AR) [42], face recognition [31], and autonomous driving [49]. Prevailing way for depth sensing is to use specific range sensing equipment, e.g., binocular stereo cameras [18, 22, 34], time of flight (ToF) cameras [60], and structured light sensors [36]. Direct depth es-
toning from 2D images/videos is highly desired, but this
monocular depth estimation is well known for its ill-posed
nature due to scale ambiguities in the estimation.

Some early works explore different priors in solving this
problem, such as shading and texture [40], object sizes and
locations, as well as occlusion and perspective cues [47].
Some data driven methods are also proposed, which trans-
fer known depth map to an unseen scene by matching its se-
mantic features [11, 20]. With the advance of deep learning,
many convolution neural network (CNN) based method has
made remarkable achievements in this area [3, 8, 15, 32].

The aforementioned methods are mostly trained on sin-
gle image datasets. When applying it to video, it usually
causes unsatisfactory temporal stability, visually per-
ceived as flickering over time. Several video-based meth-
ods address this problem by exploiting the power of recur-
rent structure. A typical example is the use of convo-
lutional LSTM [29, 55]. Those methods do not explicitly
enforce temporal consistency, hence they rely heavily on
stable dense ground truths which are expensive to obtain.
Another type of methods adopt multi-view constraints and
reconstruct the scene from motion [44, 52, 59]. However,
such reconstruction relies on accurate matching features in
temporal space, which are commonly downgraded by dif-
ficulties including less textural area and motion blur in dy-
namic scenes.

In this paper, we introduce a simple yet effective ap-
proach to enforce temporal consistency of video depth es-
timation. A basic assumption behind it is that flickering
comes out if corresponding pixels in consecutive frames
drift a lot. By restricting and aligning the predictions under
such correspondence, the model is guided to produce depth
estimation with strong consistency under single frame in-
ference. At the same time, we define a metric that fairly
evaluate the stability of depth estimation results over time,
and it does not require labelled ground truth for processing.
By conducting experiments on public benchmarks, we show
that our method improves stability in term of our defined
temporal consistency, while does not impair original depth
estimation quality. With a model of good generalization
ability, we are easy to extent it to common real world scenes
which are lack of depth ground truth by imposing geometric
constraints and fine-tuning the network using videos only.

We summarize our contributions as follows:

• We propose an effective method to impose temporal
constraints during training. The model is then learned
to generate stable depth predictions with only single-
frame inputs. Experiments on public benchmarks in-
dicate that our method improve temporal consistency
without harming depth accuracy.

• We extend the method to dynamic video without depth
ground truths. We show that we can easily enforce
constraints and regularize the model using unlabelled
videos. Subjective study illustrates that our method
provides results with better consistency values as well
as perceptible less flickering.

2. Related Work

Mono-depth Estimation There exists extensive liter-
ature studying the recovery of depth from single image.
Early stage methods predicted the structure by exploiting
cues from shading [40], occlusion [47] or semantic la-
bels [20]. Saxena et al. [35] is the first to adopt a learning-
base method to regress depth from local feature with MRF
post-processing. With the boosting of deep learning, dif-
f erent convolutional architectures have been designed and
applied on this problem. [3, 15, 16, 21]. Besides employing
an improved network structure, several works have formul-
ated well-designed loss such as ordinal loss [4] and multi-
scale gradient loss [45] to obtain refined estimation. Xian
et al. [51] addresses the ranking loss around object edges
in order to promote sharpness.

Directly regressing single image to depth result is pos-
sible but it requires a large amount of dense labelled data.
Collecting depth maps in varying scenarios with high qual-
ity still remains as a big challenge in our community. Dif-
ferent solutions have been introduced to alleviate the de-
mand of data. One way is to manually produce depths from
source images. Ground truth are collected in the form of
annotated relative depth [1] or disparity [45, 50]. Structure-
from-motion (SfM) is also employed to extract depth from
multi-view reconstruction [17, 19]. Recent work [32] shows
that generalization capacity of the model can be substan-
tially improved by mixing diverse data source with the help
of a scale- and shift-invariant loss. Self-supervised manner
is also explored to get rid of limited depth maps. Those
methods [5, 6, 7, 53] fall into the same category that applies
a photometric loss to minimize the explicit reconstruction
error. Thus estimate depth is implicitly optimized as an in-
termediate result. Luo et al. [24] separate monocular depth
estimation into a view synthesis problem along with a stereo
matching problem, which are accessible by including extra
synthetic training data. Mahdi et al. [25] propose to refine
a pre-trained depth estimation network by merging depth
estimations in different resolution patches.

However, all above methods do not include temporal ,
which limits their performance on video sequences. It leads
to inevitable visual flickering when we directly apply the
model frame by frame.
Depth Network
Flow Network
Warping
Temporal 
Consistency 
Loss
Depth Loss
𝑋!
𝑋!”#
𝐷!”#
𝐷!
𝑓!”#
𝐷$!”#
𝑋%!”#
𝑀!
Occlusion
Estimation
𝐷!
∗

Figure 2: Our pipeline for imposing temporal consistency into the depth estimation. We use a network to directly predict depth maps on each frame independently. Beside this, we calculate the optical flow between two consecutive frames and use the flow vectors to warp one frame to align to another. After this alignment and occlusion handling, we try to minimize the depth difference loss between the two in our training. By this we can enforce the temporal consistency in our depth estimation network. During inference time, our depth estimation network can directly estimate the depth maps frame by frame which is computationally efficient.

**Depth Estimation from Videos** Multiple frames in sequential manner provide additional information for depth estimation. Considerable works [39, 41, 46, 52, 58, 59] have shown that multi-view constraints are helpful to solve the geometry on every target frames. Those methods do not rely on labelled data and do produce more stable results. However, SfM-based methods suffers severe degradation when applied to fast-changing scenes with dynamic objects as it is challenging to find corresponding relations because of motion blur and poor texture. At the same time, those approaches are computationally very extensive and not efficient in video-related tasks. Recurrent networks like LSTM [28, 55] have been used to capture temporal information and encourage consistency. It is simple to deploy such models, though their generalization performance is build upon sufficient depth data supply in varied scenarios. Luo et al. [23] combines advantages of neural network and multi-view constraints. It refines the network at test time towards geometrical consistency by aligning correspondences in 3D space. Kopf et al. [13] jointly estimate camera pose and depth alignment to remove the limitation of SfM for video, and using a geometry-aware filter to improve high-frequency details. Our method adopts a similar idea but instead restricts alignments in pixel level which forbids the error brought by 3D reconstruction.

**Temporal Consistency** Flickering occurs when single-frame based method is applied to video clips. Depth value at the same pixel is not stable across frame and it endures radical drifts which causes notable visual incoherence. Various works have been undertaken to enforce temporal consistency in the field of video-to-video synthesis [48], video enhancement [54], style-transfer [33] and semantic segmentation [26]. Though attempts have been made to evaluate stability in video segmentation [43] and video object detection [57], there is an absence of a clear definition of temporal consistency in the scope of video depth estimation. Instability is introduced in [23] where it declares instability of reliable tracks as the real discrepancy in 3D space. This is fundamentally correct but does not straightforwardly reflect the visual flickering in pixel space. In this work, we use optical flow [9] as well to develop a novel consistency metric that is closely related to 2D perception. In the meanwhile, we propose an practical method to enforce temporal consistency in video depth estimation.

### 3. Our Definition and Method

In this section, we present the definition of our temporal consistency metric, and illustrate how we can enforce temporal consistency during training and improve model capacity of producing stable depth estimation results.

#### 3.1. Metric of Temporal Consistency

A consistency-enforced model should produce a sequence of depth estimation results which do not contain notable flickering over the whole period. Variations between
two consecutive depth maps generally come up from several aspects: movement of objects in the scene, shift and rotation of camera, and unexpected frequent drifts in the same area. Considering a video of high frame rate, the change between frames is minor, and depth value between two consecutive frames should be almost identical in the corresponding pixels. Single-image based methods commonly do not impose any constraints on such variations across frames, thus flickering occurs such that depth value of pixels belonging to the same identity in 3D coordinate go through frequent and random drifts in temporal axis.

In order to measure the stability of a sequential depth results, we need to identify corresponding pixels in each pair of consecutive frames, and determine how those pixels fluctuate across the whole video. Following the previous work which focuses on evaluating consistency in semantic segmentation [43], it is easy to come up with the idea of searching corresponding pixels by optical flow.

Denote \( X_i (i = 1, \ldots, n) \) as the original video of length \( n \). Processed by a depth estimation model, no matter it uses single-frame or multi-frame structure, we can get a sequence of depth predictions \( D_i (i = 1, \ldots, n) \) of the same resolution. Considering any two consecutive frames \( X_i, X_{i+1} \) and their estimated depth maps \( D_i, D_{i+1} \), the variation in depth space for all pixels should be minor. Directly calculating the difference between \( D_i \) and \( D_{i+1} \) does not fairly reflect the flickering coming from undesirable fluctuation of depth value, since all the pixels are not aligned due to movement of objects and camera. By employing the optical flow estimator, e.g., [2, 10, 38], we compute the dense flow map from \( X_{i+1} \) to \( X_i \) as \( f_{i+1 \rightarrow i} \), such that we can use warping operation \( w(\cdot) \) to warp \( D_{i+1} \) as \( \hat{D}_{i+1} = w(D_{i+1}, f_{i+1 \rightarrow i}) \), which is now spatially aligned with \( D_i \). Similarly, we also warp the color frame \( X_{i+1} \) as \( \hat{X}_{i+1} = w(X_{i+1}, f_{i+1 \rightarrow i}) \), so that we obtain a valid mask by comparing the color difference. The mask \( M_i \) is created by thresholding color differences as \( ||X_i - \hat{X}_{i+1}|| < \delta \), and we only compare the depth fluctuation on valid pixels.

It comes to a simple idea of calculating the absolute difference between those aligned and valid depth pixels. By this means, the temporal consistency (TC) metric at frame \( i \) is formulated as:

\[
aTC_i = \frac{1}{\sum (M_i == 1)} M_i \cdot ||D_i - \hat{D}_{i+1}||. \tag{1}
\]

Two adjacent depth maps are considered consistent, if the temporal metric \( aTC \) has a small value. However, it does not take the range of depth value into consideration. During visualization, we generally normalize depth results and view them in relative version. If a model tends to produce depth results that is small in certain region after normalization, the above proposed formulation is not correct since it reflects the absolute variation that is highly correlated to the scale. Even though flickering appears radically, the video could be recognized as stable since \( aTC \) metric is small.

The accuracy threshold used in monocular depth estimation benchmark [3] inspires us to consider the relative fluctuation of depth value across frames. Ratio of change is more reasonable than absolute difference and it is more associated with visual flickering. Therefore, we modify previous metric defined in Eqn. 1 to use relative difference as:

\[
rTC_i = \frac{1}{\sum (M_i == 1)} M_i \cdot \max \left( \frac{D_i}{D_{i+1}}, \frac{\hat{D}_{i+1}}{D_i} \right) < thr. \tag{2}
\]

This metric reflects the percentage of matching pixels that go through modest variation at frame \( i \). When a video of depth maps is stability, the metric should have small value at every time \( i \). The temporal consistency of a video can be written as:

\[
TC = \frac{1}{n} \sum_{i=1}^{n} rTC_i \tag{3}
\]

which is the average ratio of intolerant variation over the entire sequence. In our experiments, we conduct a user study to demonstrate that our proposed temporal consistency metrics in video depth estimation is consistent with human perception, whose details can be found in Sec. 4.3.

### 3.2. Enforcing Temporal Consistency for Video Depth Estimation

Figure 2 shows the overview of our pipeline to enforce temporal stability in video depth estimation. Our pipeline is built on single image depth estimation and intend to impose learning temporal consistency onto the original depth network. In the training stage, we take two consecutive frames \( X_i \) and \( X_{i+1} \), and feed the two frames into the depth network to generate the corresponding depth predictions respectively. The network outputs are denoted as \( D_i \) and \( \hat{D}_{i+1} \).

**Depth loss** Following the single image depth estimation methods, we can measure the distance of the depth predictions to the ground truth depth \( D_i^* \) and \( \hat{D}_{i+1}^* \). This is just the conventional loss on depth predicting accuracy that is being minimized during training. The forward pass, loss computation, and backward pass apply to the two frames independently and we denote the loss in this part as depth loss \( L_d \):

\[
L^D_i = \sum_{t=i+1}^{n} L_{depth}(D_t, \hat{D}_t^*), \tag{4}
\]

where \( L_{depth}(\cdot) \) is the single image depth estimation loss, e.g., \( L_1 \), \( L_2 \), and \( SIloss \) [32]. In our experiments, we use \( SIloss \) as our training loss.
The First Row: The input frames from NYU. The Second Row: The generated depth estimation maps using CLSTM. The Third Row: The generated depth estimation maps using DeepV2D. The Last Row: The generated depth estimation maps using our method. In order to better visualize the video stability, we zoomed-in and concatenated the same regions from different frames and depth estimation on the last column.

Temporal consistency loss Besides the conventional depth prediction loss, we additionally impose temporal consistency loss which measures the difference between the depth estimations from the two frames. Our intention is to penalize large depth shift between two consecutive frames. As there are motions between the two frames, we need to compensate for it before measuring the distance. This is achieved by estimating the optical flow from the frame \(X_{i+1}\) to the frame \(X_i\) using an off-the-shelf optical flow estimator [38] with which we can warp the depth prediction \(D_{i+1}\) to align with \(D_i\) and denote the warping result as \(\hat{D}_{i+1}\). With the same flow vectors, we also warp the color frame \(X_{i+1}\) as \(\hat{X}_{i+1}\). After this alignment, we can impose our temporal stability loss as the \(L_1\) distance between the two depth maps as:

\[
L_i^{TC} = M_i \cdot ||D_i - \hat{D}_{i+1}||,
\]

where \(M_i\) is the occlusion mask. Unlike the \(M_i\) defined in Eqn. 1 and Eqn. 2 that uses hard thresholding, we use soft occlusion mask in Eqn. 5 as \(M_i = \exp(-\sigma \cdot (||X_i - \hat{X}_{i+1}||_2^2))\).

Overall loss The overall loss function for training depth estimation with temporal consistency is defined as:

\[
\mathcal{L} = \sum_i (\mathcal{L}_i^D + \lambda \mathcal{L}_i^{TC}),
\]

where \(\lambda\) controls the weight of the temporal consistency term in the total loss.

Training general video depth While the depth estimation loss defined in Eqn. 4 require depth ground truth, it is hard to capture large scale and diverse video depth dataset. To solve this issue, we propose to use supervision distilled from the state-of-the-art monocular depth methods. Specifically, we use the MiDaS network [32] as the teacher, which pre-trained on large variety of data from multiple datasets. MiDaS has proven to have decent generalization ability is potentially suited to our aim of general depth estimation. In this setting our pipeline described earlier still applies but we just treat the output from MiDaS network as the supervision signal \(D_i^t\).

4. Experiments

To demonstrate that our method can provide a more stable video depth prediction, we run extensive experiments on
Figure 4: Visual comparison with other state-of-the-art methods on NYU validation set. **The First Row**: The input frames from NYU. **The Second Row**: The generated depth estimation maps using BTS. **The Third Row**: The generated depth estimation maps using CLSTM. **The Last Row**: The generated depth estimation maps using our method. In order to better visualize the video stability, we zoomed-in and concatenated the same regions from different frames and depth estimation on the last column.

<table>
<thead>
<tr>
<th>Method</th>
<th>Lower is better ↓</th>
<th>Higher is better ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Speed(s)</td>
<td>RMSE</td>
</tr>
<tr>
<td>BTS[16]</td>
<td>0.0846</td>
<td>0.3271</td>
</tr>
<tr>
<td>CLSTM [56]</td>
<td>0.0829</td>
<td>0.4437</td>
</tr>
<tr>
<td>DeepV2D [41]</td>
<td>1.1619</td>
<td>0.3211</td>
</tr>
<tr>
<td>Ours</td>
<td>0.0585</td>
<td>0.3460</td>
</tr>
</tbody>
</table>

Table 1: The quantitative comparison on our NYU test set. The top scores in each category are highlighted in bold. TC: temporal consistency metrics in Eqn. 3. The resolution of input frame is 320 × 420. All the inference experiments are conducted on a NVIDIA Tesla P40 GPU.

We implement our approach based on the public deep learning library PyTorch [27]. For all experiments, our network use the same network structure as in MiDaS [32]. We initiate the network with its official parameters pre-trained on mixing datasets, and fine-tune on different datasets. Our data augmentation includes random horizontal flip and random temporal flip of video sequences.

4.1. NYU Depth v2

**Dataset** We perform experiments on NYU Depth v2 dataset [37]. The NYU dataset records 464 video sequences with both RGB and depth camera. These video sequences cover a variety of indoor scenes, including living rooms, kitchen, bathrooms. For all sequences, we preserve one image for every five images to prevent the time interval from being too short. Following the official train split, our experiment use 249 scenes for training. We separate the test images from the densely labeled pairs, which come from the rest 215 scenes. For each test image, we find its 4 adjacent frames from the corresponding video sequence to construct a test video clip. A total of 630 clips are used to construct our one standard benchmark dataset [37]. In addition, we conduct several human subjective studies to validate the effectiveness of our temporal consistency metric and our method to impose temporal consistency.
The state-of-the-art single image based method MiDaS [32] can produce frame-wise high quality depth maps but with noticeable flickering over time. After imposing the temporal consistency into the depth estimation model, our method can predict temporally more stable depth predictions.

**Benchmark Performance** Table 1 describes the quantitative comparison between our method with other learning based methods [16, 41, 56] on the NYU dataset. Among them, BTS [16] is a single frame based method, CLSTM [56] and DeepV2D [41] are multi-frames based methods. All these models are trained on the NYU dataset and released by their original author. To reduce the inference time, the iteration number of DeepV2D is set to be 1.

Except the traditional metrics used in previous arts [16, 41, 56], we additionally evaluate the stability of the generated depth videos based on our temporal consistency metric in Eqn. 3. However, our approach is far superior than them on speed and the temporal consistency metric, and obtains competitive results on other metrics.

Figure 3 and 4 show some visual comparison examples on NYU testset. CLSTM flickers on the first frame in both Figure 3 and 4. DeepV2D also flickers on the first frame at the same region in Figure 3. BTS has a small flickering in the lower half of the third frame region as Figure 4 shows. Compared with others, Our method can produce better temporal consistency depth estimation.

### 4.2. Real Scenes

**Dataset** To explore the generalizability of our approach, we extend experiments on DAVIS [30] and Videvo [14] datasets. These sequences are more challenging since they are recorded from various outdoor real scenes, and the objects recorded are not all rigid and static.

Our training set consists of 60 video sequences from DAVIS dataset and 99 sequences from Videvo dataset. The temporal consistency are evaluated on the DAVIS official testset, which contains 30 video sequences.

**Implementation Details** For DAVIS and Videvo dataset, all input images are resized to 384 \( \times \) 512 resolution. The model is trained 5 epochs using Adam optimizer with a fixed learning rate 0.001. Due to the lack of ground truth,
our model is trained in a distillation way. We use the output of the official MiDaS model as \( D^* \) shown in Figure 2.

**User Study** We have conducted a user study with 20 participants to further compare the visual stability of generated depth videos. For each video sequence in DAVIS testset, there are 4 generated depth videos (one group) produced by ours and other methods. Except one group for the tutorial, all the rest (a total of 29 groups) are displayed to 20 participants in a random order. For each group, the participants are required to pick the most stable depth video.

**Results and Analysis** Since there are no ground truth in DAVIS testset, we only evaluate the stability on the generated video results. Table 2 reports the quantitative comparison on our proposed metric and the user study results. The temporal consistency value of our method significantly exceeds that of other methods. Surprisingly, none of the results produced by CLSTM and DeepV2D are selected in our user study, which indicates that our approach can generate far more visually stable depth estimations than those two methods.

Figure 5 displays some visual comparison examples on DAVIS testset. We only compare with MiDaS since no one select any results produced by DeepV2D and CLSTM in our user study. For better visualization, we concatenate adjacent image patches from different frames of the same video and so does their corresponding depth patches.

In the top row, MiDaS shows obvious overall flickering in the third and fifth depth estimations, while our depth exhibits a better stability in all regions.

In the second row, MiDaS wrongly estimate the depth of the sky in the first four frames and shows a strong flickering across all frames. And our model separates the sky and trees in the fifth frame, while MiDaS blends them together. The second row describes that even in extreme cases where the sky presents, our approach can maintain the temporal consistency and estimated the relative depth correctly.

It is commendable that our method can generate stable depth estimations even in the case of large object motion and camera displacement as the third row demonstrates, while MiDaS is unstable in the area outside the railing.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>CLSTM</th>
<th>DeepV2D</th>
<th>MiDaS</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC ↑</td>
<td>0.4224</td>
<td>0.4245</td>
<td>0.6025</td>
<td>0.8211</td>
</tr>
<tr>
<td>User Pick % ↑</td>
<td>0</td>
<td>0</td>
<td>26.55</td>
<td>73.45</td>
</tr>
</tbody>
</table>

Table 2: The quantitative comparison and user study results on the DAVIS validation set. The best scores in each category are highlighted in bold. TC: our temporal consistency metric in Eqn. 3.

**4.3. Metric Effectiveness**

As stated before, we design a temporal consistency metric 3 to evaluate the stability of the generated depth videos. In order to prove that our metric is correlated to human perception of visual stability, we carefully design a simple yet effective user study to reveal this correlation.

We prepare 20 pairs of generated depth videos. Each pair of depth video are produced from the same video sequence by different models. Except one pair for the participants’ tutorial, all the rest video pairs are used for our user study (a total of 19 pairs). We invite 20 participants to compare and select the more stable depth video in each pair. To avoid subjective bias, the high temporal consistency score videos in each pair are randomly ordered. We control the score difference of each video pairs to be nearly uniform distributed in four intervals shown on the abscissa axis on the left of Figure 6.

The results of our user study are displayed in Figure 6. As shown in the left of Figure 6, 77.37% of results is consistent with the score generated by our temporal consistency metric, which strongly indicates that the proposed metric can sufficiently represent human perception of visual stability. In the right of Figure 6, the abscissa axis represents the metric score difference intervals between video pairs, while the bar height represents the corresponding percentage of correct answers collected in the user study. We can observe that the percentage of correct answers improves as the intervals enlarges, which indicates that the variance of our metric aligns with the human perception visual stability.

**5. Conclusion**

In this work, we introduce a simple yet effective method to improve the temporal consistency of video depth estimation under single frame inference. A temporal consistency metric, which correspond with human perception of video stability, is proposed as well. Experiments demonstrate that our approach can exhibit a more stable depth estimation and can be generalized on dynamic real world videos without corresponding depth ground truth.
References


