Video Upright Adjustment and Stabilization[1]

Jucheol Won
wjc0319@dgist.ac.kr
DGIST
Daegu, Korea

Sunghyun Cho
sodomau@postech.ac.kr
POSTECH
Pohang, Korea

Abstract

We propose a novel video upright adjustment method that can reliably correct slanted video contents. Our approach combines deep learning and Bayesian inference to estimate accurate rotation angles from video frames. We train a convolutional neural network to obtain initial estimates of the rotation angles of input video frames. The initial estimates are temporally inconsistent and inaccurate. To resolve this, we use Bayesian inference. We analyze estimation errors of the network, and derive an error model. Based on the error model, we formulate video upright adjustment as a maximum a posteriori problem where we estimate consistent rotation angles from the initial estimates. Finally, we propose a joint approach to video stabilization and upright adjustment to minimize information loss. Experimental results show that our video upright adjustment method can effectively correct slanted video contents, and our joint approach can achieve visually pleasing results from shaky and slanted videos.

1 Introduction

Smartphone cameras are always available everywhere, and action cams such as GoPro have gained huge popularity for the past several years, so now many people enjoy shooting and sharing videos of their activities and everyday lives. However, shooting a high-quality video is still challenging for casual users. Videos captured by casual users often show severely shaky and slanted contents, which not only degrade aesthetic quality but also make a video visually uncomfortable, and sometimes even cause dizziness.

To remove camera shakes from a video, video stabilization has been extensively studied [1, 8, 9, 14, 18, 19, 20, 21, 22, 23]. On the contrary, video upright adjustment, which is a problem to fix a slanted video, has not gained enough attention, even though slanted video contents are another important factor that degrades aesthetic quality. For image upright adjustment, several attempts have been made. Gallagher *et al.* [7] and Lee *et al.* [17] proposed line detection-based methods. Fischer *et al.* [5] proposed a convolutional neural network (CNN)-based approach to predict the in-plane rotation angle of an image.

However, such image upright adjustment methods cannot be directly applied to a video. Since existing methods are not 100% reliable, applying them to each video frame independently guarantees neither reliability nor temporal consistency. More importantly, these methods sometimes produce completely wrong estimates. For example, Lee *et al.*'s method may fail when edge detection fails [17]. Fischer *et al.*'s method can also fail due to various

[1] Most of this work was done when Sunghyun Cho was at DGIST.

reasons such as limited training datasets [5]. However, it is unclear how to detect such errors, especially from CNNs, as they do not rely on explicit models.

In this paper, we propose a video upright adjustment method that can reliably estimate the in-plane rotation angles of video frames and correct slanted video contents. To the best of our knowledge, our method is the first attempt to video upright adjustment. Our method is a three-step process that combines deep learning and Bayesian inference for reliable upright adjustment. The first step obtains initial estimates of angles from input video using a CNN. Initial estimates may have errors that result in an unreliable and temporally inconsistent upright adjustment result. To resolve this, we analyze errors and derive an error model. The second step estimates reliable and consistent angles from the initial estimates by solving a maximum a posteriori (MAP) problem based on the error model. Finally, the third step rotates back the input video frames by the estimated angles to straighten up slanted contents.

We also propose a simple yet effective joint solution to video stabilization and upright adjustment. For handling slanted and shaky videos, we may apply video stabilization and upright adjustment one after the other. However, such a naïve combination causes excessive loss of spatial resolution as both steps trim off frame boundaries to remove invalid pixels after warping video frames. Furthermore, it may also degrade the quality of the resulting video, as the second step uses cropped video frames from the first step that contains less information. To avoid excessive loss of spatial resolution and to improve the quality, our joint approach first estimates rotation angles without rotating video frames. Then, it estimates warping parameters for video stabilization from the uncropped input video frames reflecting the estimated rotation angles. Finally, it warps video frames for both stabilization and upright adjustment. As we use uncropped video frames for computing warping parameters, we can stabilize a video more accurately while preserving more contents.

Related Work. While there have been several studies correcting rotation of still shot images, there is no work correcting rotation of video to our knowledge. Besides [5, 7, 17], several other works have been proposed for estimating camera orientation and correcting rotated images. Coughlan and Yuille [3] introduced the Manhattan World assumption to estimate the orientation of a camera from an image of a man-made environment. Since then, many camera calibration methods [4, 15, 24, 28] have been proposed based on the Manhattan World assumption. All these approaches detect line segments, and find vanishing lines and points for estimating the orientation of a camera. As a result, they can fail for images without lines consistent with the Manhattan World assumption.

Other approaches have been proposed to utilize other image properties such as textures. Zhang *et al.* [29] estimate calibration parameters from textures in an image. Wang and Zhang [27] proposed a method that detects the orientation of an image as one of the limited set of angles $\{0°, 90°, 180°, 270°\}$ using support vector machines with hand-crafted features based on color and edge information. Joshi and Guerzhoy [11] modified the VGG-16 network to train an orientation classifier that classifies the orientation of an image into a limited set of angles. Recently, Jung *et al.* [12] proposed an upright adjustment method for 360° spherical panorama images, which is based on line detection. For 2D images, He *et al.* [10] proposed a content-preserving rotation method. Given a rotation angle, their method finds the best mesh-based transform that minimizes the loss of contents while rotating an image, but it has a limitation requiring prior information of a rotation angle.

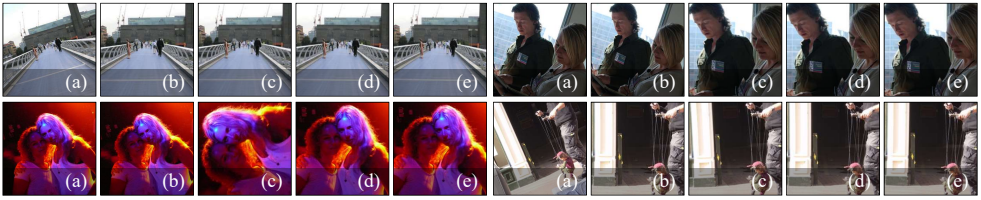


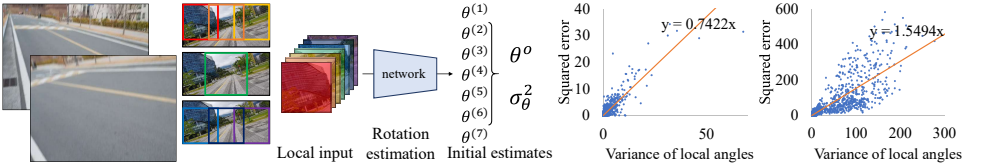
Figure 1: Image upright results. (a) Input images. (b) Photoshop upright tool. (c) Fischer *et al.* [5]. (d) Our results. (e) Ground truth.

2 Rotation Estimation Network

In this section, we describe our rotation estimation network, which will be used for initial rotation angle estimation. We assume input videos of arbitrary natural scenes. To handle a wide variety of videos, we train a CNN so that the network can estimate rotation angles even from images without straight lines, as done in [5]. A CNN based approach has several benefits over traditional methods: 1) it provides high accuracy as will be shown later, 2) it can handle images without vertical or horizontal edges, 3) it is computationally efficient as it does not involve edge detection or sophisticated optimization, and 4) it is easy to implement.

Regarding the network architecture, we modify the VGG-19 network [25], as the general performance of the VGG-19 network is superior to that of AlexNet [16], which is used by Fisher *et al.* [5]. We change the output size of the last fully-connected layer from 1,000 to 1 to produce a single regression result corresponding to the rotation angle of an input image. For our training dataset, we sampled 110K images whose contents are upright from the World Cities dataset [26]. We also sampled 500 images that look upright from the World Cities dataset for our validation set. For training the network, we randomly rotated images in our datasets from -45° to 45°, as this range can cover most videos, and training a network for a wider range degrades the overall accuracy of the network as Fischer *et al.* [5] reported. Then, the largest square-shaped region at the center of each rotated image is cropped. We then trained the network to predict the random rotation angles used for rotating the input images. Instead of training the network from scratch, we fine-tuned a pre-trained VGG-19 network. We used L1 loss, and trained our network using Adam optimizer [13] and stochastic gradient descent. The trained network achieved average error of 0.7° on our validation set. For more details, please refer to our supplementary material.

Fig. 1 shows examples of image upright adjustment using our rotation estimation network. As the network does not rely on straight lines, it successfully estimates rotation angles even for the images without any straight lines (the bottom left of Fig. 1). On the contrary, Photoshop upright tool, which is based on [17], fails to upright-adjust images on the second and third rows as it relies on explicit line detection. For comparison, we also trained AlexNet [16], which is used in [5], with our own training dataset. The trained network achieved average error of 3.32° on our validation set, which is smaller than 4.63° reported in [5]. This difference can be due to different training and validation sets. Nevertheless, its error is still larger than that of the VGG-19 based network. Some examples of AlexNet are also shown in Fig. 1(c).



(a) (b) (c) (d)

Figure 2: (a) Failure example of rotation estimation network. Top: input frame, bottom: rotated result by an angle estimated by the network. (b) Local areas for estimating rotation angles. (c) Squared error vs. angle variance computed from 1,500 images. (d) Squared error vs. angle variance computed from a video of 1,000 frames.

3 Error Analysis

Even though the average error of the rotation estimation network is low, it may still fail to estimate correct angles for various reasons (Fig. 2(a)). To measure the reliability of angles estimated by the network, we experimentally analyze errors and derive an error model. Our analysis starts with the following assumption: *If a network produces an incorrect estimate for an image, then it means that the image has either no reliable features or features that contradict each other.* In other words, wrong estimates are obtained from less reliable or contradicting features, which are likely to be different in different local areas whereas local areas should have the same in-plane rotation as the rotation is caused by a camera. Therefore, the network will produce inconsistent angle estimates for different local areas of an image, and we may predict the reliability of estimated angles by inspecting their consistency.

To prove the validity of our assumption, we experimentally analyze the relationship between the expectation of squared error and the variance of estimated angles from local regions of a video frame. For the analysis, we collected 1,500 images whose contents are upright, and randomly rotated them. For each rotated image I_i where $i \in \{1, \cdots, 1500\}$ is an image index, we cropped seven different local regions as shown in Fig. 2(b), and estimated its rotation angle for each region using the rotation estimation network. We used large local regions with large overlaps whose width and height are $5/6$ of the height of input frames to avoid the effect of perspective and lens distortion. We denote the estimate from each local region by $\theta_i^{(j)}$ where $j \in \{1, \cdots, 7\}$ is a region index. Then, we computed the mean and variance of the seven angles, which we call mean angle and angle variance, denoted by θ_i and $\sigma_{\theta,i}^2$, respectively. We defined error e_i as the difference between the mean angle θ_i and the ground truth angle θ_i^{gt}.

Finally, we plotted points $(\sigma_{\theta,i}^2, e_i^2)$, and fitted a line by the least squares method as shown in Fig. 2(c). The fitted line corresponds to the averages of squared errors with respect to different angle variances, or equivalently the variances of errors assuming that errors follow zero-mean distributions. As Fig. 2(c) shows, the fitted line has a positive slope indicating that errors increase as do angle variances. Fig. 2(d) shows another experimental result using a video of 1,000 frames. For each video frame, we manually annotated the ground truth rotation angles, and performed the same process as described above. Fig. 2(d) shows a similar trend, which proves the validity of our assumption.

Based on our analysis, we model the variance of error as a function of angle variance:

$$\sigma_{err}^2\left(\sigma_\theta^2\right) = \alpha\sigma_\theta^2 \tag{1}$$

where σ_{err}^2 and σ_θ^2 are the variance of error, and angle variance, respectively. α is a scale

factor corresponding to the slopes of the lines in Fig. 2(c) and (d). While α can be directly obtained from the lines, its absolute scale is not important in our method as will be shown later.

4 Video Upright Adjustment

Initial angle estimation. We first obtain initial angle estimates using the rotation estimation network from an input video of N frames. Specifically, for the t-th video frame, we crop seven local regions as described in Sec. 3, and estimate their rotation angles using the rotation estimation network. We compute their mean angle θ_t^o, which will be used as the initial angle estimate for the t-th frame. We also compute the angle variance $\sigma_{\theta,t}^2$ of the t-th frame.

Robust angle estimation. Given the initial angles $\theta^o = \{\theta_1^o, \cdots, \theta_N^o\}$, we estimate accurate and temporally consistent angles $\theta = \{\theta_1, \cdots, \theta_N\}$ from them for all video frames simultaneously. We formulate the problem as a maximum a posteriori (MAP) problem defined as:

$$\underset{\theta}{\operatorname{argmax}}\, p(\theta|\theta^o) = \underset{\theta}{\operatorname{argmax}}\, p(\theta^o|\theta)p(\theta) \tag{2}$$

where $p(\theta|\theta^o)$ is a posterior distribution, $p(\theta^o|\theta)$ is a likelihood, and $p(\theta)$ is a prior. Assuming that error in each initial angle estimate follows a normal distribution with zero mean and the variance $\sigma_{err}^2(\sigma_{\theta,t}^2)$, we define the likelihood $p(\theta^o|\theta)$ as:

$$p(\theta^o|\theta) = \prod_{t=1}^{N} \mathcal{N}\left(\theta_t - \theta_t^o | 0, \sigma_{err}^2(\sigma_{\theta,t}^2)\right) \tag{3}$$

where $\mathcal{N}$ denotes the normal distribution. For temporal consistency, we define $p(\theta)$ as:

$$p(\theta) = \prod_{t=1}^{N-1} \mathcal{N}\left(\rho(\theta_t, \theta_{t+1}) | 0, 2\sigma_{temp}^2\right) \tag{4}$$

where $\rho(\theta_t, \theta_{t+1})$ is a pair-wise temporal consistency measure, and σ_{temp}^2 is a parameter to control the shape of the prior. To reflect relative rotational motion between consecutive frames, we define $\rho(\theta_t, \theta_{t+1})$ as $\rho(\theta_t, \theta_{t+1}) = |(\theta_{t+1} - \theta_t) - \phi_t|^2$ where ϕ_t is the relative rotational angle between the t-th and $(t+1)$-th frames. ϕ_t can be easily estimated by feature point matching. We use SURF [2] for feature detection and description as it is invariant to rotation and scaling. We fit a similarity transform using RANSAC [6], and extract the rotation angle ϕ_t from it. The effect of Eq. (4) is examined in the supplementary material.

Optimization. By applying negative logarithm to Eq. (2), we can obtain the energy function:

$$E(\theta) = \sum_{t=1}^{N} \frac{1}{\sigma_{\theta,t}^2} |\theta_t - \theta_t^o|^2 + \lambda \sum_{t=1}^{N-1} |(\theta_{t+1} - \theta_t) - \phi_t|^2 \tag{5}$$

where $\lambda = \alpha / \sigma_{temp}^2$. This is a simple quadratic function of θ, which can be solved efficiently using the least-squares method. While λ is defined as a function of α and σ_{temp}^2, it can be set directly as well, as σ_{temp}^2 is a user parameter. In all the experiments, we used $\lambda = 4.0$.

Warping. Once θ are obtained, video upright adjustment can be done by simply rotating back input video frames and cropping their boundaries to remove invalid pixels. The spatial size of a resulting video is determined by the maximum rotation angle of its input video. While this produces satisfactory results when the maximum rotation angle is small, it may result in excessive information loss if at least one input frame is severely rotated. To avoid

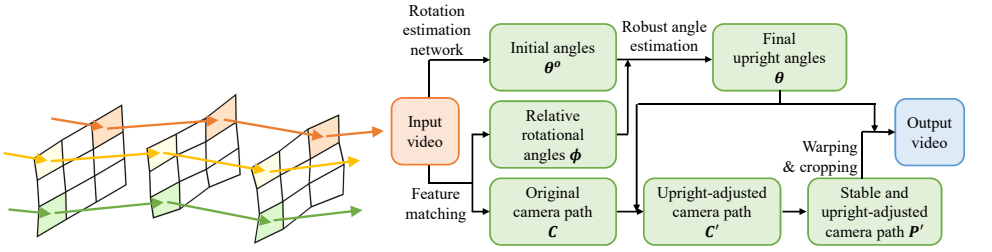


Figure 3: (a) Bundled camera path model of [20]. (b) Overview of our joint approach.

this, we truncate angles using a pre-defined upper bound τ as $\hat{\theta}_t = \text{sgn}(\theta_t)\min(|\theta_t|, \tau)$ where $\hat{\theta}_t$ is a truncated angle, and $\text{sgn}(\cdot)$ is the sign function defined as 1 for $x > 0$, 0 for $x = 0$, and -1 for $x < 0$. Input video frames are then rotated back by $\hat{\theta}$ instead of θ. τ controls the trade-off between information loss and rotation correction. A small τ preserves larger areas while a large τ straightens up more frames.

5 Joint Stabilization and Upright Adjustment

Our joint approach is built upon a state-of-the-art video stabilization method of Liu *et al.* [20]. We first briefly review Liu *et al.*'s method, and introduce our joint approach.

5.1 Bundled Camera Paths for Video Stabilization

For modeling camera paths, Liu *et al.* introduce a mesh-based motion model called *bundled camera paths*, which models a camera motion as a set of simple camera motions based on homographies (Fig. 3(a)). Specifically, the bundled camera path model splits video frames into an $M \times M$ regular grid, where $M = 16$ in their system. For each grid cell, the camera motion is modeled using homographies as done in traditional 2D stabilization methods.

For stabilizing a video, Liu *et al.* first estimate local homographies between consecutive frames using SURF [2] features. Let $F_t^{(i)}$ be a 3×3 homography matrix that aligns the i-th grid cells of the t-th and $(t+1)$-th frames. Then, the original camera path $C_t^{(i)}$ is defined as the camera motion from the first to the t-th frame in the i-th grid cell, which is computed as:

$$C_t^{(i)} = F_{t-1}^{(i)} C_{t-1}^{(i)} = F_{t-1}^{(i)} \cdots F_2^{(i)} F_1^{(i)}. \tag{6}$$

Given the original path $C = \{\cdots, C_t^{(i)}, \cdots\}$, Liu *et al.* compute a stable path $P = \{\cdots, P_t^{(i)}, \cdots\}$ by solving an energy minimization problem. Finally, for each grid cell of each frame, a warping transform $B_t^{(i)}$ is computed as $B_t^{(i)} = P_t^{(i)} \left(C_t^{(i)}\right)^{-1}$, and a stabilized video is generated by warping input video frames by the warping transforms. We refer to [20] for more details.

5.2 Joint Approach

To achieve both upright adjustment and stabilization, we may first simply perform upright adjustment and then stabilization. Our joint approach roughly follows this process, but in a more efficient and effective way. The key ideas for our joint approach are as follows: 1) The camera path of an upright-adjusted video does not need to be computed from actually

rotated frames, but can be computed by simply rotating the paths of the original video by the rotation angles for upright adjustment. 2) As upright adjustment and stabilization share feature matching, warping and cropping, we can perform them once for both tasks. Consequently, we can estimate more accurate rotation angles and camera motion as we estimate them directly from the original uncropped video. As warping and cropping are done only once, we can avoid excessive loss of spatial resolution. Also we can save a huge amount of computation time, as feature matching and warping, which are the two most time-consuming components, are performed only once.

Fig. 3(b) illustrates our joint approach. For upright adjustment, we first estimate initial angles θ^o using the rotation estimation network. In parallel, we perform feature matching and compute the relative rotation angles ϕ. We then compute the final rotation angles θ from θ^o and ϕ. For stabilization, we estimate the camera path C of the input video using the feature matching result, and convert it to the camera path C' of the upright-adjusted video.

The camera path C' of the upright-adjusted video is derived as follows. Let x_t be a point in the t-th original video frame, which is represented as a 3D vector in homogeneous coordinates. Let x'_t be its corresponding point in the t-th upright-adjusted frame. Then $x'_t = R_t x_t$, where R_t is a rotation matrix that rotates points by θ_t. As $x'_{t+1} = R_{t+1} F_t^{(i)} (R_t)^{-1} x'_t$, we can derive the camera motion $F_t^{(i)\prime}$ between the t-th and $(t+1)$-th upright-adjusted frames as:

$$F_t^{(i)\prime} = R_{t+1} F_t^{(i)} R_t^{-1}. \tag{7}$$

Then, the original camera path $C_t^{(i)\prime}$ of the i-th grid cell at the t-th upright-adjusted video frame can be computed similarly to Eq. (6):

$$C_t^{(i)\prime} = F_{t-1}^{(i)\prime} \cdots F_2^{(i)\prime} F_1^{(i)\prime} = R_t C_t^{(i)} R_1^{-1}. \tag{8}$$

Once we obtain C', we compute a new stable camera path P' for the upright-adjusted video as done in [20]. We then compute warping transforms $B_t^{(i)}$ from the input video frames to upright-adjusted and stabilized frames as a combination of rotation for upright adjustment, and warping for stabilization:

$$B_t^{(i)} = P_t^{(i)\prime} \left(C_t^{(i)\prime} \right)^{-1} R_t. \tag{9}$$

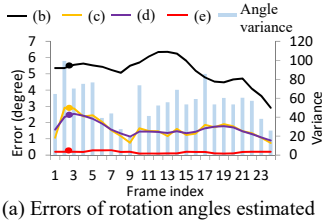
Finally, we warp the input video frames by $B_t^{(i)}$ and crop them to obtain the final result.

6 Experiments

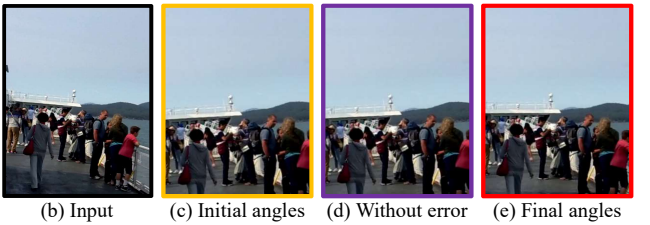
We implemented our method using PyTorch and Python. For the video stabilization method of Liu *et al.* [20], we used a third-party Matlab source code on internet as the authors' code is not available. Table 1 shows the computation time of each step of our method measured on a PC with an Intel Core i7-7700K, 16GB RAM, and a GeForce GTX 1080Ti. As Table 1 shows, our method is efficient enough for practical usage. We experimented our method using various videos that we captured ourselves or downloaded from Youtube. To measure errors reported in this section, we manually measured ground truth rotation angles of video frames. All video examples shown in this section as well as additional examples can be found in our supplementary material.

Upright Adjustment	millisec./frame	Stabilization	millisec./frame
Initial rotation estimation	5	Camera path estimation	250
Feature matching	20 - 150	Stable path computation	362
Rotation estimation	0.0063	Warping	134
Frame rotation & cropping	8		

Table 1: Computation time for each step. The input video size has 1000 frames of size 1280×720. Computation time for feature matching varies depending on video frames as the number of features detected from each frame varies. We also report the computation times of video stabilization components for comparison. The computation times of video stabilization components are different from those reported in [20], as we use a third-party Matlab code while Liu *et al.*'s original code was implemented in C++.



(a) Errors of rotation angles estimated by different methods



(b) Input (c) Initial angles (d) Without error model (e) Final angles

Figure 4: Comparison of different angle estimation. (a) Errors of rotation angles estimated by different methods. In (a), the curve labeled as (b) is the ground truth rotation angle of the input frames. The curves labeled as (c), (d), and (e) are errors of initial angles, angles estimated without using the error model, and our final angles, respectively. The images on the right correspond to the black, yellow, purple, and red points in (a), respectively.

Initial vs. final angles. To verify the effectiveness of our robust angle estimation, we compare errors of initial angles and final angles estimated by our method for a sequence of video frames in Fig. 4(a). Note that using initial angles is equivalent to directly applying Fisher *et al.*'s method [5] to video frames. Errors of the initial angles are relatively large and clearly follow the trend of angle variances. This shows that the angle variance is a good predictor of error. In contrast, our final angle errors are much smaller and show no correlation with the angle variances. Fig. 4(c) and (e) show video frames rotated by initial and final angles marked as yellow and red points in Fig. 4(a), respectively.

Error model based on angle variance. To investigate the effect of the error model based on angle variance, we conduct another experiment where we use a fixed value (16.0) instead of $\sigma^2_{\theta,t}$ in Eq. (5). Estimated angles using a fixed value show similar errors to those of the initial angles (the purple curve in Fig. 4(a)), as errors cannot be detected by a fixed value. Fig. 4(d) shows an example video frame corresponding to the purple point in Fig. 4(a).

Joint upright adjustment and stabilization. Fig. 5 shows examples of video stabilization with upright adjustment. As video stabilization does not correct slanted contents, its result still remains slanted (Fig. 5(a)). As upright adjustment trims off image boundaries to remove invalid pixels, video stabilization needs a much smaller crop window for stabilized video frames, resulting in severe loss of spatial resolution (Fig. 5(b)). On the other hand, our joint approach uses entire video frames for video stabilization while reflecting rotation angles. As a result, our joint approach minimizes loss of spatial resolution, and a resulting video has a wider field of view and more effectively stabilized contents (Fig. 5(c)).

Fig. 6 shows a comparison between joint video stabilization using our initial angles and

(a) Video stabilization (b) Video stabilization after upright adjustment (c) Joint video stabilization and upright adjustment

Figure 5: Crop windows in an input video frame. Red and blue are crop windows for video stabilization, and upright adjustment, respectively.

(a) Initial angles (b) Final angles

Figure 6: Joint upright adjustment and stabilization using our initial angles and final angles.

final angles. As video stabilization removes high-frequency jitters, we may obtain upright-adjusted and stabilized results simply using initial angles. While this approach may work for initial angles with small errors, video stabilization cannot remove large rotation errors, and consequently, can still produce slanted video frames. In contrast, our video upright adjustment can effectively detect and remove such large errors and produce satisfying results.

User study. Finally, we report a user study result. We recruited 20 participants. All the participants are graduate students majoring in either electrical engineering or computer science, but not related to computer vision or graphics. Each participant was shown 20 pairs of videos, and asked to choose a visually more pleasing video for each pair. Among the 20 pairs, 10 pairs are of original slanted videos and their corresponding upright adjustment results, and the other 10 pairs are results of stabilization and our joint approach. It was unknown to the participants which ones were ours. The user study result is shown in Table 2. For the pairs of original videos and their upright adjustment results, the participants preferred our results by 78% on average. For the pairs of stabilization and our joint approach, the participants preferred our results by 82.5% on average. This shows that our method can effectively improve the perceptual quality of videos in both scenarios. For more details of the user study, we refer the readers to the supplementary material.

7 Conclusion

In this paper, we proposed a novel video upright adjustment method. Our approach uses a CNN to estimate initial estimates of the rotation angles of input video frames so that angles can be robustly estimated even for images without obvious horizontal or vertical lines. To handle errors in initial estimates, we derived an error model on the basis of error analysis. Estimation of reliable and consistent angles is then formulated as a MAP estimation problem based on the derived error model. For higher accuracy, our approach also utilizes relative rotation between consecutive frames. Finally, we proposed a joint stabilization and upright adjustment, which can effectively correct shaky and slanted video contents. The effectiveness of our method was examined by various experiments including user study.

Video No.	1	2	3	4	5	6	7	8	9	10	Avg.
Original vs. upright-adjustment	40	90	85	90	85	70	85	65	85	85	78
Stabilization vs. joint stabilization & upright-adjustment	50	90	85	95	85	95	95	80	85	65	82.5

Table 2: User study. Numbers in the table cells are the percentages of users who prefer upright-adjusted videos.

Limitations and future work. Our method has a few limitations. Our method cannot handle videos whose all frames are rotated by more than 45°, as our network cannot estimate rotation angles from any frames reliably. Our method may also fail for videos of unnatural scenes that our network is not trained with. As our method relies on feature matching for temporal consistency, the method may fail when feature matching fails. Rotating video frames by large angles results in severe loss in spatial resolution. A small τ may preserve larger areas, but it also leads to results that are not fully corrected. Such information loss might be alleviated by adopting the content-aware rotation method proposed in [10]. We assume that input videos have well-defined upright orientations. However, some videos, e.g., videos showing only the sky or the ground, may have no such orientations. Lastly, extension to 360° videos would also be an interesting future direction.

Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7066316)

References

[1] Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. User-assisted video stabilization. *Computer Graphics Forum*, 33(4):61–70, July 2014. ISSN 0167-7055.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proc. European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[3] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by bayesian inference. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 941–947 vol.2, 1999.

[4] Patrick Denis, James H. Elder, and Francisco J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *Proc. European Conference on Computer Vision (ECCV)*, ECCV '08, pages 197–210. Springer-Verlag, 2008. ISBN 978-3-540-88685-3.

[5] P. Fischer, A. Dosovitskiy, and T. Brox. Image orientation estimation with convolutional networks. In *German Conference on Pattern Recognition (GCPR)*. Springer, 2015.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

[7] Andrew C Gallagher. Using vanishing points to correct camera rotation in images. In *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*, pages 460–467. IEEE, 2005.

[8] Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, 31(5):126:1–126:10, September 2012. ISSN 0730-0301.

[9] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 225–232. IEEE, 2011.

[10] Kaiming He, Huiwen Chang, and Jian Sun. Content-aware rotation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 553–560, 2013.

[11] U. Joshi and M. Guerzhoy. Automatic photo orientation detection with convolutional neural networks. In *Conference on Computer and Robot Vision (CRV)*, pages 103–108, 2017.

[12] Jinwoong Jung, Beomseok Kim, Joon-Young Lee, Byungmoon Kim, and Seungyong Lee. Robust upright adjustment of 360 spherical panoramas. *The Visual Computer*, 33 (6-8):737–747, 2017.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Johannes Kopf. 360° video stabilization. *ACM Transactions on Graphics (TOG)*, 35 (6):195:1–195:9, November 2016. ISSN 0730-0301.

[15] Jana Košecká and Wei Zhang. Video compass. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Proc. European Conference on Computer Vision (ECCV)*, pages 476–490, Berlin, Heidelberg, 2002. Springer. ISBN 978-3-540-47979-6.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[17] Hyunjoon Lee, Eli Shechtman, Jue Wang, and Seungyong Lee. Automatic upright adjustment of photographs with robust camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):833–844, 2014.

[18] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3D video stabilization. *ACM Transactions on Graphics (TOG)*, 28(3):44:1–44:9, July 2009. ISSN 0730-0301.

[19] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4, 2011.

[20] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78, 2013.

[21] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4209–4216. IEEE, 2014.

[22] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proc. European Conference on Computer Vision (ECCV)*, pages 800–815. Springer International Publishing, 2016.

[23] Y. Matsushita, E. Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(7):1150–1163, July 2006. ISSN 0162-8828.

[24] F. M. Mirzaei and S. I. Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2454–2461, Nov 2011.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011.

[27] Yongmei Michelle Wang and Hongjiang Zhang. Detecting image orientation based on low-level visual content. *Computer Vision and Image Understanding*, 93(3):328–346, 2004.

[28] H. Wildenauer and A. Hanbury. Robust camera self-calibration from monocular images of manhattan worlds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2831–2838, June 2012.

[29] Zhengdong Zhang, Yasuyuki Matsushita, and Yi Ma. Camera calibration with lens distortion from low-rank textures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2321–2328. IEEE, 2011.