Master's Thesis
석사 학위논문

Video Upright Adjustment and Stabilization

Jucheol Won(원 주 철 元 柱 喆)

Department of
Information & Communication Engineering

DGIST

2020

Master's Thesis
석사 학위논문

Video Upright Adjustment and Stabilization

Jucheol Won(원 주 철 元 柱 喆)

Department of
Information & Communication Engineering

DGIST

2020

Video Upright Adjustment and Stabilization

Advisor: Professor Hoon Sung Chwa
Co-advisor: Professor Sunghyun Cho

by

Jucheol Won
Department of Information & Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information & Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

11. 08. 2019

Approved by

Professor Hoon Sung Chwa (signature)
(Advisor)

Professor Sunghyun Cho (signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Video Upright Adjustment and Stabilization

Jucheol Won

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

11. 08. 2019

Head of Committee	Prof. Hoon Sung Chwa	(signature)
Committee Member	Prof. Sunghyun Cho	(signature)
Committee Member	Prof. Sang Hyun Park	(signature)

MS/IC
201722011

원 주 철. Jucheol Won. Video Upright Adjustment and Stabilization. Department of Information & Communication Engineering. 2020. 29p. Advisors Prof. Hoon Sung Chwa, Co-Advisors Prof. Sunghyun Cho

ABSTRACT

We propose a novel video upright adjustment method that can reliably correct slanted video contents that are often found in casual videos. 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 from the network are temporally inconsistent and inaccurate. To resolve this, we use Bayesian inference. We analyze estimation errors of the network, and derive an error model. We then use the error model to formulate video upright adjustment as a maximum a posteriori problem where we estimate consistent rotation angles from the initial estimates, while respecting relative rotations between consecutive frames. Finally, we propose a joint approach to video stabilization and upright adjustment, which minimizes information loss caused by separately handling stabilization and upright adjustment. Experimental results show that our video upright adjustment method can effectively correct slanted video contents, and its combination with video stabilization can achieve visually pleasing results from shaky and slanted videos.

Keywords: Upright adjustment, Video stabilization, Camera path

List of Contents

List of tables

List of figures

I. 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 as shown in Fig. 1(a), which not only degrade aesthetic quality but also make a video visually uncomfortable, and sometimes even cause dizziness.

(a) Shaky and slanted video (b) Stabilized video

Figure 1. Example of video stabilization on a shaky and slanted video.

Video stabilization, which is a problem to remove camera shakes from a video, has been extensively studied [24, 19, 20, 10, 8, 22, 23, 1, 21, 15]. Many video editing software and services like Adobe After Effects and Youtube have been providing such functionality. 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 example, Fig. 1(b) shows a video frame stabilized using Warp Stabilizer in Adobe Premier. Although the camera motion in the video is stabilized, its contents still remain slanted, making the video uncomfortable to watch.

For image upright adjustment, several attempts have been made. Gallagher [7] proposed a method for estimating the in-plane rotation angle of an image, which is based on vertical line detection and vanishing point estimation. Lee *et al*. [18] proposed a method for correcting perspective distortion in an image, which detects vanishing lines and finds a perspective transform from the vanishing lines. Fischer *et al*. [5] proposed a convolutional neural network (CNN) based method that predicts the in-plane rotation angle of an image.

However, such image upright adjustment methods cannot be directly applied to a video. Since existing image upright adjustment 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 [18]. Fischer *et al*.'s method can also fail due to various reasons such as limited training datasets and network capability [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 novel 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 designed as 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 as done in [5]. Initial estimates of angles from the network can have errors as mentioned above that result in an unreliable and temporally inconsistent upright adjustment result. To resolve this, we analyze errors from the CNN 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. For higher accuracy, we impose a temporal consistency prior that exploits relative rotation between consecutive frames. As a result, we can obtain more accurate and consistent estimates of the rotation angles.

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. Casual videos are often not only slanted but also shaky as aforementioned. We may apply video stabilization and upright adjustment one after the other to resolve both. However, this two-step approach 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.

1.1. Related work

While there have been extensive studies correcting rotation of still shot images, there is no work correcting rotation of video to our knowledge. Besides [7, 18, 5], 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 [16, 4, 25, 30] 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*. [31] estimate calibration parameters from textures in an image. Wang and Zhang [29] 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 [12] 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*. [13] proposed an upright adjustment method for 360° spherical panorama images. Their method also finds vanishing lines, and uses them to estimate a transform. For 2D images, He *et al*. [11] 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.

II. 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 [27], as the general performance of the VGG-19 network is superior to that of AlexNet [17], 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.

To generate our training dataset, we used World Cities Dataset [28], which consists of 2M images of various scenes taken in 40 cities. We randomly sampled 300K images from the dataset, and manually removed slanted images and other disturbing images such as logos and objects captured in unusual directions. We finally obtained 110K images for our training set. We assumed that all images in the training set are upright, i.e., their rotation angles are zero. The final training set consists of not only images of man-made structures, but also pictures with no obvious straight lines such as portrait pictures, and images of curved objects. The proportion of such images in our training set is about a half. We also sampled 500 images that look upright from World Cities Dataset for our validation set. Among 500 images, 100 images do not have straight horizontal or vertical lines such as boundaries of buildings.

Figure 2. Image upright adjustment results.

For training the network, we randomly rotated images in our datasets. We restrict the range of possible rotation angles to $[-45°, +45°]$, 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. The cropped region was resized to 224 × 224, which is the input size of the VGG-19 network. 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 finetuned a pre-trained VGG-19 network. Specifically, we initialized all the parameters in the network with pre-trained parameters except for the last fully-connected layer, which was randomly initialized. During the first epoch, we used Adam optimizer [14] with a learning rate of 1.0×10^{-4}. From the second epoch, we used stochastic gradient descent and set the learning rate to 2.0×10^{-3}. We decreased the learning rate by 0.1 times for every 55K iterations until it reached at 2.0×10^{-6}. We used L1 loss that measures absolute difference between estimated and ground truth angles. We trained the network for 330K iterations. The trained network achieved average error of $0.7°$ on our validation set.

Fig. 2 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 final row of Fig. 2). On the contrary, Photoshop upright tool, which is based on [18], fails to upright-adjust images on the second and third rows as it relies on explicit line detection.

For comparison, we also trained AlexNet [17], 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.

III. 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. 3). 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. 4, 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 $\mathrm{j} \in \{1, \cdots, 7\}$ is a region index.

Figure 3. Failure example of rotation estimation network. (a) Input frame. (b) Result rotated by an angle estimated by the network.

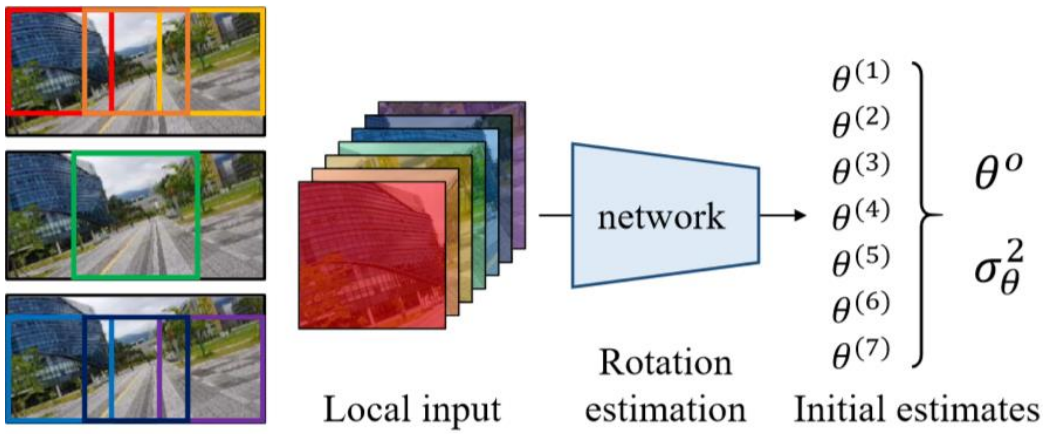


Figure 4. Local areas for estimating rotation angles.

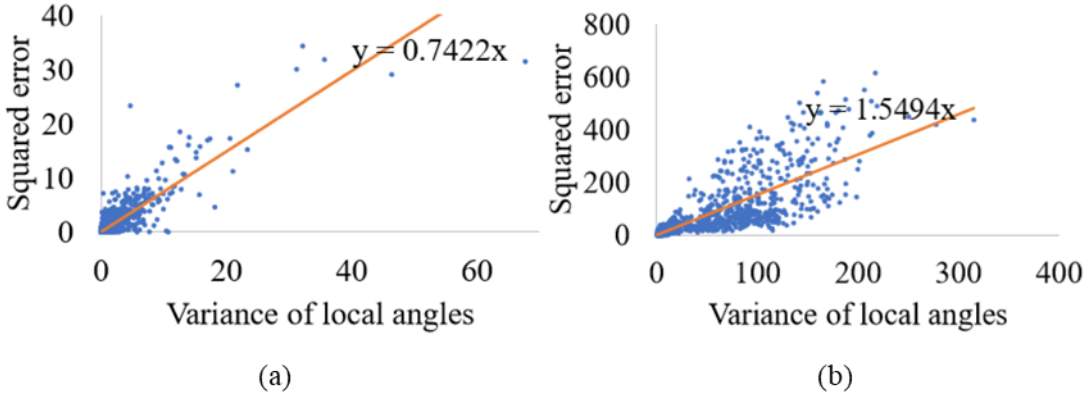


Figure 5. Squared error versus angle variance. (a) Plot from 1,500 image dataset. (b) Plot from a video of 1,000 frames.

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. 5(a). 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. 5(a) shows, the fitted line has a positive slope indicating that errors increase as do angle variances.

We also performed the same experiment with a different set of images. This time, we prepared a video that has rotational camera motion. The video has 1,000 frames, and we manually labeled the ground truth rotation angles for all video frames. Then, for each video frame, we performed the same process as the first experiment. Fig. 5(b) shows a plot of the second experiment. While the distribution of points is not the same, the general trends in both plots are similar, i.e., errors increase as do angle variances. These two experiments prove the validity of our assumption.

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

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

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. 5. While α can be directly obtained from the lines in Fig. 5, its absolute scale is not important in our method as will be shown later

Our approach can also be considered as follows. We may consider a distribution of images that are rotated by the same angle, and the network as an estimator of the angle, which is a parameter of the distribution. We may

crop different local regions from a video frame. The cropped regions are images rotated by the same angle, so they can be considered as samples from the same distribution. Then, the reliability of the estimator, or the reliability of its output can be measured by the variance of the estimator, which can be estimated by the sample variance [9].

IV. VIDEO UPRIGHT ADJUSTMENT

Our video upright adjustment method consists of three steps: initial estimation, robust angle estimation, and warping, as described in Sec. 1. We describe each step in more detail in the following.

4.1. 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.

4.2. Robust angle estimation

Given the initial angles $\theta^o = \{\theta_1^o, \theta_2^o, \cdots, \theta_N^o\}$, we estimate accurate and temporally consistent angles $\theta = \{\theta_1, \theta_2, \cdots, \theta_N\}$ 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:

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

where $\mathcal{N}$ denotes the normal distribution. To promote temporal consistency of θ, we define the prior $p(\theta)$ as:

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

where $\rho(\theta_t, \theta_{t+1})$ is a pair-wise temporal consistency measure. σ_{temp}^2 is a parameter to control the shape of the prior.

To produce temporally smooth angles, we may define $\rho(\theta_t, \theta_{t+1})$ as:

$$\rho(\theta_t, \theta_{t+1}) = |\theta_{t+1} - \theta_t|^2 \tag{5}$$

assuming that there are no or very small rotational motion between consecutive frames. However, Eq. (5) ignores rotational motion that may exist between consecutive frames. To reflect such rotational motion, we define $\rho(\theta_t, \theta_{t+1})$ as:

$$\rho(\theta_t, \theta_{t+1}) = |(\theta_{t+1} - \theta_t) - \phi_t|^2 \tag{6}$$

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 based image alignment. Specifically, given two adjacent frames, we extract feature points and match them. We use SURF [2] for feature detection and description as it is invariant to rotation and scaling. Then, we fit a similarity transform using RANSAC [6], and extract the rotation angle ϕ_t from the similarity transform.

4.3. Optimization

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

$$E(\theta) = \sum_{t=1}^{N} \frac{1}{\sigma_{\theta,t}^2} |\theta_t - \theta_t^o|^2 + \lambda \sum_{t=1}^{N-1} |(\theta_{t+1} - \theta_t) - \phi_t|^2 \tag{7}$$

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 $\lambda = 4.0$.

4.4. 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 this, we truncate angles using a pre-defined upper bound τ as:

$$\hat{\theta}_t = \mathrm{sgn}(\theta_t)\min(|\theta_t|, \tau) \tag{8}$$

where $\hat{\theta}_t$ is a truncated angle, and $\mathrm{sgn}(\cdot)$ 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.

V. JOINT UPRIGHT ADJUSTMENT AND STABILIZATION

Our joint approach combines a state-of-the-art video stabilization method of Liu *et al*. [22] with our upright adjustment method. In this section, we first briefly review Liu *et al*.'s method, and introduce our joint approach.

5.1. Bundled camera paths for video stabilization

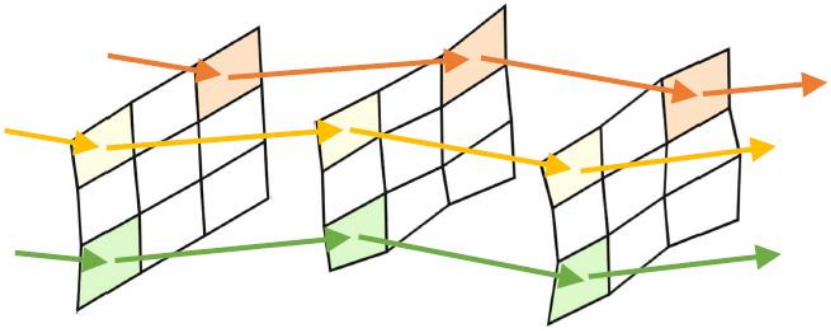


Figure 6. Bundled camera path model of [19]. Camera motion is modeled using homographies for each grid cell.

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. 6). Specifically, the bundled camera path model splits video frames into an $\mathrm{M} \times \mathrm{M}$ regular grid, where $\mathrm{M} = 16$ 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:

$$C_t^{(i)} = F_{t-1}^{(i)} C_{t-1}^{(i)} = F_{t-1}^{(i)} \cdots F_2^{(i)} F_1^{(i)} \tag{9}$$

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. For more details, we refer the readers to [22].

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. 7 shows an overview of 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

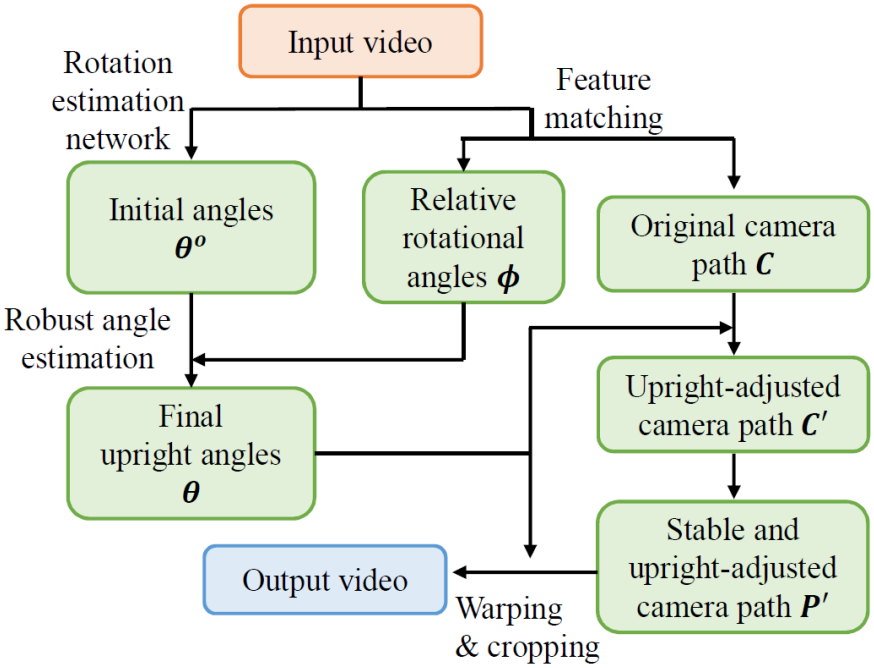


Figure 7. Overview of our joint upright adjustment and stabilization.

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:

$$F_t^{(i)\prime} = R_{t+1} F_t^{(i)} R_t^{-1} \tag{10}$$

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. (9):

$$C_t^{(i)\prime} = F_{t-1}^{(i)\prime} \cdots F_2^{(i)\prime} F_1^{(i)\prime} = R_t C_t^{(i)} R_1^{-1} \tag{11}$$

Once we obtain C', we compute a new stable camera path P' for the upright-adjusted video as done in [22]. 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:

$$B_t^{(i)} = P_t^{(i)\prime}\left(C_t^{(i)\prime}\right)^{-1} R_t \tag{12}$$

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

VI. EXPERIMENTS

We implemented the rotation estimation network and the other parts of our method using PyTorch and Python. For the video stabilization method of Liu *et al*. [22], 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, which was 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.

Step	millisec./frame
Initial rotation estimation	5
Feature matching	20 - 150
Rotation estimation	0.0063
Frame rotation & cropping	8

Components for video stabilization	
Original camera path estimation	250
Stable path computation	362
Warping	134

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 computa-

tion times of video stabilization components for comparison. The computation times of video stabilization components are different from those reported in [19], as we use a third-party Matlab code while Liu *et al*.'s original code was implemented in C++.

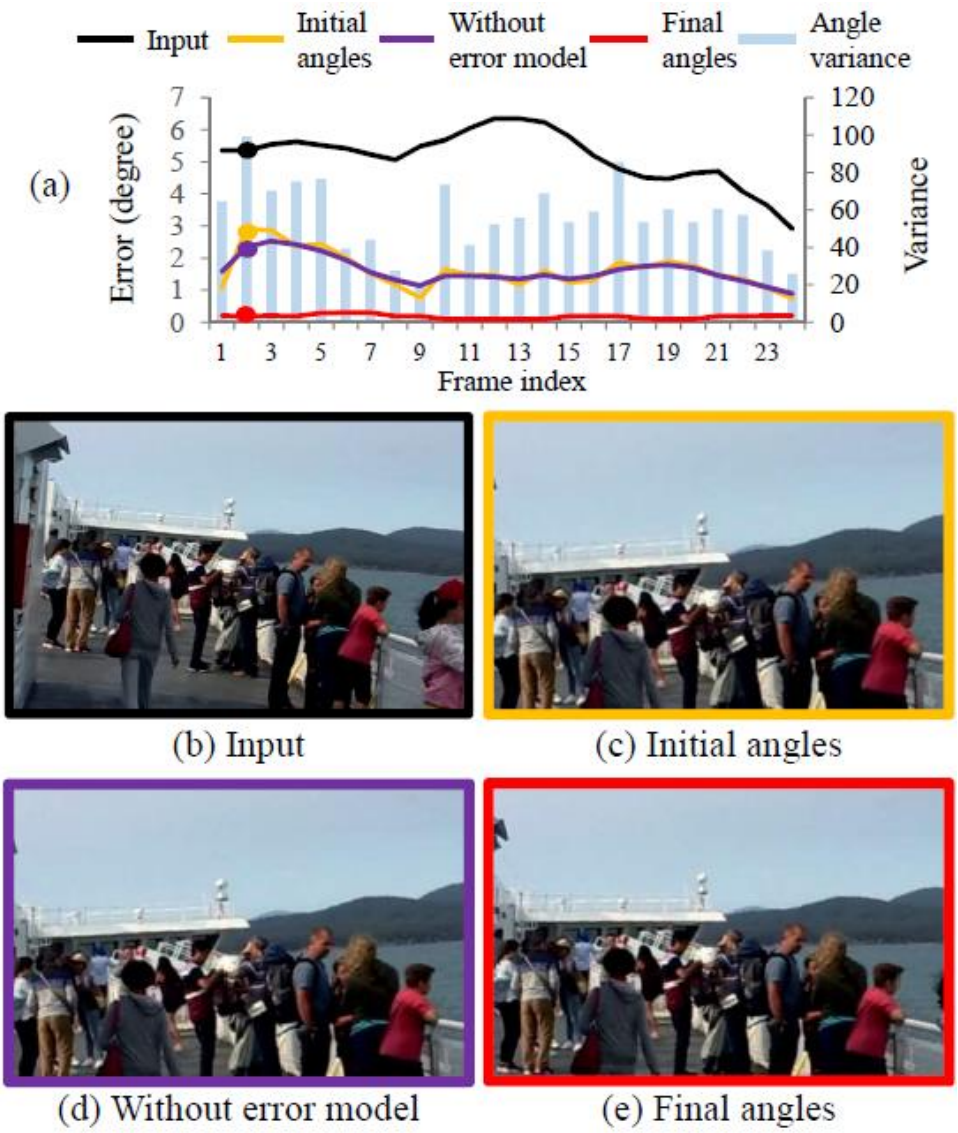


Figure 8. Comparison of different angle estimation. (a) Errors of rotation angles estimated by different methods. Images on the bottom two rows are (b) an input video frame, and its corresponding results rotated by (c) an initial angle, (d) an angle estimated without using the error model, and (e) our final angle, respectively. (b), (c), (d), and (e) 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. 8(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. 8(c) and (e) show video frames rotated by initial and final angles marked as yellow and red points in Fig. 8(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 $\sigma_{\theta,t}^2$ in Eq. (7). Estimated angles using a fixed value show similar errors to those of the initial angles (the purple curve in Fig. 8(a)), as errors cannot be detected by a fixed value. Fig. 8(d) shows an example video frame corresponding to the purple point in Fig. 8(a).

Reflecting relative rotation Fig. 9 demonstrates the effect of the temporal consistency prior that reflects relative rotation between consecutive frames. The simple prior (Eq. (5)) generates smooth but inaccurate angles as it simply smoothes the rotation angles of neighboring frames. On the contrary, the temporal consistency prior (Eq. (6)) produces an accurately upright-adjusted result.

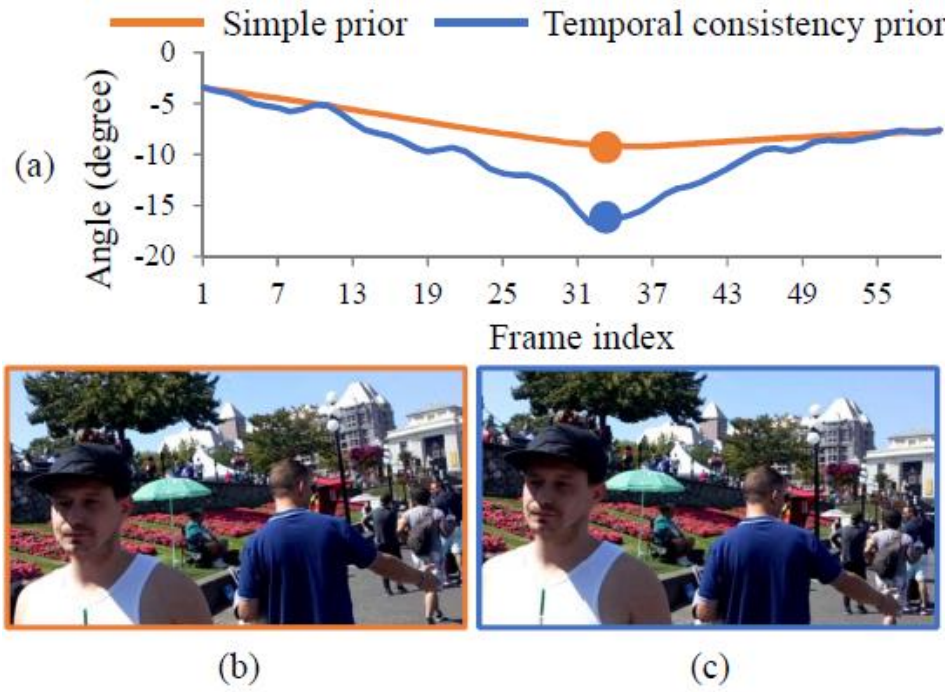
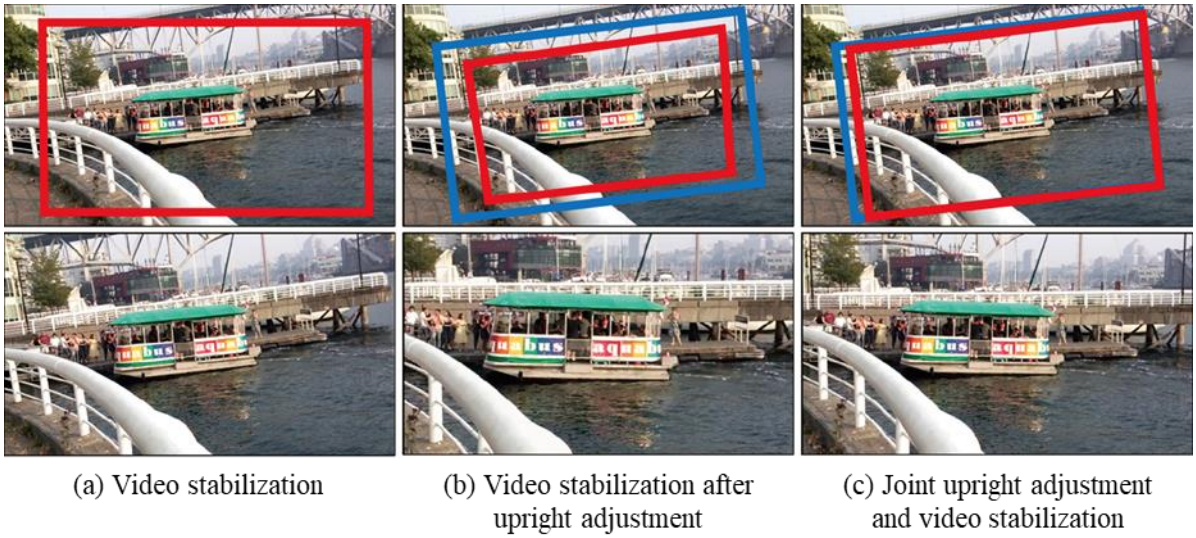


Figure 9. Comparison between a simple prior (Eq. (5)) that does not reflect relative rotation between consecutive frames and our temporal consistency prior (Eq. (6)). (a) Rotation angles estimated by different priors. (b) a result of the simple prior corresponding to the orange circle in (a). (c) our result corresponding to the blue circle in (a).

Upright adjustment for visual recognition An interesting application of our method is to pre-process video data before visual recognition tasks such as object detection, the performance of which can be largely degraded for slanted contents. To examine the effect of video upright adjustment, we randomly sampled three video clips from the validation set of the ImageNet VID validation dataset [26], and smoothly rotated them by up to 40°. We then applied our method to them so that we obtained artificially rotated video set and upright-adjusted video set. We measured the performance of a state-of-the-art video object detection method [32] on those videos. As Table 2 shows, while camera rotation severely degrades the performance of object detection, our method can successfully recover the original performance.

Original	Rotated	Upright-adjusted
0.5302	0.3448	0.5238

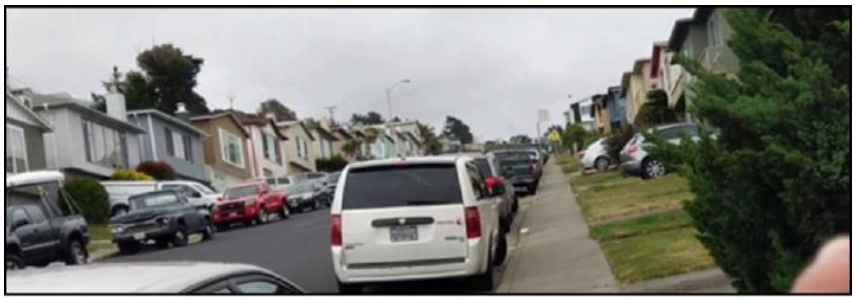
Table 2. Object detection performance. The values in table are mean average precision at Intersection-over-Union threshold of 0.5. We tested on three videos described above.



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

Figure 10. Top row: crop windows in an input video frame. Red and blue are crop windows for video stabilization, and upright adjustment, respectively. Bottom row: cropped frames.

Joint upright adjustment and stabilization Fig. 10 shows examples of video stabilization with upright adjustment. As video stabilization does not correct slanted contents, its result still remains slanted and uncomfortable to watch (Fig. 10(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. 10(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. 10(c)).



(a) Initial angles

(b) Final angles

Figure 11. Joint upright adjustment and stabilization using our initial angles and final angles.

Fig. 11 shows a comparison between joint video stabilization using our initial angles and 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 3. 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.

Video No.	1	2	3	4	5	6	7	8	9	10	Avg.
Up	40	90	85	90	85	70	85	65	85	85	78
Joint	50	90	85	95	85	95	95	80	85	65	82.5

Table 3. User study. Up: original slanted video vs. upright-adjusted video. Joint: stabilized video vs. jointly stabilized and upright-adjusted video. Numbers in table cells are the percentages of users who prefer upright-adjusted videos.

VII. CONCLUSION

In this paper, we proposed a novel video upright adjustment method, which combines deep learning and Bayesian inference to achieve high-quality results. 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 and object detection task.

Limitations and future work Our method has a few limitations that we would like to resolve in future. 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. Such information loss might be alleviated by adopting the content-aware rotation method proposed in [11]. 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.

References

[1] Bai, J. et al. User-assisted video stabilization. *Computer Graphics Forum*, 33(4):61–70, July (2014)

[2] Bay, H. et al. Surf: Speeded up robust features. In Proc. European Conference on Computer Vision (ECCV), pages 404–417 (2006)

[3] Coughlan, J. M. and Yuille, A. L. 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] Denis, P. et al. Efficient edge-based methods for estimating manhattan frames in urban imagery. In Proc. European Conference on Computer Vision (ECCV), pages 197–210. Springer-Verlag (2008)

[5] Fischer, P. et al. Image orientation estimation with convolutional networks. In German Conference on Pattern Recognition (GCPR). Springer (2015)

[6] Fischler, M. A. and Bolles, R. C. 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] Gallagher, A. C. et al. Using vanishing points to correct camera rotation in images. In Computer and Robot Vision, Proceedings. The 2nd Canadian Conference on, pages 460– 467. IEEE (2005)

[8] Goldstein, A and Fattal, R. Video stabilization using epipolar geometry. ACM Transactions on Graphics (TOG), 31(5):126:1–126:10, Sept (2012)

[9] Goodfellow, I. et al. Deep Learning. MIT Press, http://www.deeplearningbook. org. 4 (2016)

[10] Grundmann, M. et al. 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)

[11] He, K. et al. Content-aware rotation. In Proc. IEEE International Conference on Computer Vision (ICCV), pages 553–560 (2013)

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

[13] Jung, J. et al. Robust upright adjustment of 360 spherical panoramas. *The Visual Computer*, 33(6-8):737–747 (2017)

[14] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[15] Kopf, J. 360 video stabilization. *ACM Transactions on Graphics (TOG)*, 35(6):195:1–195:9, Nov (2016)

[16] Koseck, J and Zhang, W. Video compass. In A. Heyden, ´ G. Sparr, M. Nielsen, and P. Johansen, editors, Proc. European Conference on Computer Vision (ECCV), pages 476– 490, Berlin, Heidelberg (2002)

[17] Krizhevsky, A. et al. Imagenet classification with deep convolutional neural networks. NIPS, pages 1097–1105, USA (2012)

[18] Lee, H. et al. Automatic upright adjustment of photographs with robust camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):833–844 (2014)

[19] Liu, F. et al. Content-preserving warps for 3D video stabilization. *ACM Transactions on Graphics (TOG)*, 28(3):44:1–44:9, July (2009)

[20] Liu, F. et al. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4 (2011)

[21] Liu, S. et al. Meshflow: Minimum latency online video stabilization. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, Proc. European Conference on Computer Vision (ECCV), pages 800–815. Springer International Publishing (2016)

[22] Liu, S. et al. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78 (2013)

[23] Liu, S. et al. Steadyflow: Spatially smooth optical flow for video stabilization. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4209–4216. IEEE (2014)

[24] Matsushita, Y. et al. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(7):1150–1163, July (2006)

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

[26] Russakovsky, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252 (2015)

[27] Simonyan, K and Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

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

[29] Wang, Y. M. and Zhang, H. Detecting image orientation based on low-level visual content. *Computer Vision and Image Understanding*, 93(3):328–346 (2004)

[30] Wildenauer, H and Hanbury, A. 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)

[31] Zhang, Z. et al. 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)

[32] Zhu, X. et al. Flow-guided feature aggregation for video object detection. In Proc. IEEE International Conference on Computer Vision (ICCV) (2017)

요 약 문

동영상 수평 보정 및 안정화

본 논문은 일반인들이 촬영한 동영상에서 흔히 발생하는 문제인 기울어짐을 제거하여 수평이 올바른 동영상을 획득할 수 있게 하는 동영상 수평 보정(Video upright adjustment) 방법을 제안한다. 본 논문의 접근 방식은 딥 러닝(Deep learning)과 베이지안 인퍼런스(Bayesian inference)를 결합하여 동영상 프레임(Frame)에서 정확한 각도를 추정한다. 먼저 입력 동영상 프레임의 회전 각도의 초기 추정치를 얻기 위해 회선 신경망(Convolutional neural network; CNN)을 훈련시킨다. 신경망의 초기 추정치는 완전히 정확하지 않으며 시간적으로도 일관되지 않는다. 이를 해결하기 위해 베이지안 인퍼런스를 사용한다. 본 논문은 신경망의 추정 오류를 분석하고 오류 모델을 도출한다. 그런 다음 오류 모델을 사용하여 연속 프레임 간의 상대 회전 각도(Relative rotation angle)를 반영하면서 초기 추정치로부터 시간적으로 일관된 회전 각도를 추정하는 최대 사후 문제(Maximum a posteriori problem)로 동영상 수평 보정을 공식화한다. 마지막으로, 동영상 수평 보정 및 동영상 안정화(Video stabilization)에 대한 동시 접근 방법을 제안하여 수평 보정과 안정화를 별도로 수행할 때 발생하는 공간 정보 손실과 연산량을 최소화하며 안정화의 성능을 최대화한다. 실험 결과에 따르면 동영상 수평 보정으로 기울어진 동영상을 효과적으로 보정할 수 있으며 동영상 안정화 방법과 결합하여 흔들리고 기울어진 동영상으로부터 시각적으로 만족스러운 새로운 동영상을 획득할 수 있다.

핵심어: 수평 보정, 동영상 안정화, 카메라 경로