ROLLING SHUTTER CORRECTION FOR VIDEO WITH LARGE DEPTH OF FIELD

Yen-Hao Chiao, Tung-Ying Lee, Shang-Hong Lai
Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

ABSTRACT
Rolling shutter correction has attracted considerable attention in recent years. Several algorithms have been proposed to correct the distortion. Previous methods on rolling shutter correction did not consider depth variations in the scene. In this work, we overcome the limitation of the previous works that the depth of field in the scene is small. We present a correction model for rectifying the rolling shutter video based on the depth maps estimated from the rolling shutter video. In addition, we propose a two-stage optimization algorithm to estimate the temporal camera motion and the associated depth maps. Experimental results show the improvement of the proposed rolling shutter correction algorithm that takes the depth information into account.

Index Terms - Rolling shutter correction, Depth estimation, Camera motion estimation.

1. INTRODUCTION

Recently, more and more digital cameras use CMOS sensors rather than the traditional CCD sensors due to the low-cost and low-power characteristics of CMOS sensors. However, CMOS cameras use the rolling shutter architecture, which generates distortions/artifacts when the camera or objects are moving during video acquisition.

The main difference between a global shutter and a rolling shutter is the exposure scheme. For a global shutter, all pixels within a frame are exposed at the same time. In contrast, for the rolling shutter scheme, pixels are exposed sequentially row by row, from top to bottom or from bottom to top. Figure 1 illustrates the image capturing schemes for these two kinds of camera shutters. The sequential-readout structure of a rolling shutter produces undesirable distortion when there is relative motion between the camera and the scene. In a static scene, for example, if the camera is moving or panning toward the right, the vertical structure in the resulted frame becomes skewed from right to left. If the camera is moving up and down, the objects in the frame become compressed and stretched, respectively.

The geometric distortion of rolling shutter [1] can be divided into two types: skew and wobble. The skew effect occurs in situations where objects of vertical structure become slanted due to fast horizontal camera panning. The wobble effect is much more challenging than the skew effect. When the camera motion contains high-frequency vibration in many directions, objects in the video appear to be jello-like because the distortion interchanges from different directions repeatedly.

For rolling shutter video correction, we categorize the methods of rolling shutter correction into two types: translation-model based and rotation-model based methods.

Translation-model based methods assume the camera motion is composed of pure planar translation, and the scene must be planar. Liang et al. [2] derived a mathematical model of rolling shutter effect under the assumption that the scene is on the same plane, and the camera motion is simply modeled by 2D planar translation only. They estimated the low-resolution motion with the block matching method, and approximated the high-resolution motion vectors by using an interpolation method based on Bezier curve fitting. Baker et al. [3] formulated the camera motion estimation as a temporal super-resolution problem. Their optimization method accurately computes the motion even if it contains large acceleration or jitter. The corrected videos were shown to demonstrate that their algorithm can successfully remove the wobble effect.

Forssen and Ringaby [4] attempted to model the rolling shutter distortion caused by the 3D camera motion. They used the non-linear least squares method to estimate the camera ro-

This work was supported by NTHU-Novatek joint research project.
tation and apply Spherical Linear Interpolation (SLERP) to interpolate high-resolution rotation vector. They also intro-
duced another correction model that contains both translation and rotation motion. However, they failed to accurately esti-
mate both translational and rotational motion simultaneously because the cost function is not convex. Karpenko et al. [5]
used gyroscopes to obtain the temporal camera rotations for better correction.

In addition to the two main types of rolling shutter correction methods, Liu et al. [6] removed the rolling shutter effect from the aspect of video stabilization.

The flowchart of our rolling shutter correction system is shown in Figure 2. For an input video, we remove the distortion in a post-processing way. Firstly, we detect feature points in each frame, and then a set of point correspondences between every two consecutive frames are established. Sec-
ondly, we estimate the 2D camera translation and depth maps using a 2-pass alternating minimization method. With the esti-
mated camera motion and depth maps, the distorted frames can be corrected by using the model derived in section 2.

The contributions in this paper are given as follows: (1) We propose an accurate rolling shutter correction model that takes scene depth into consideration. (2) With pre-computed optical flow field, we propose a novel optimization based algorithm to estimate the depth maps and camera motion from a rolling shutter video.

2. ROLLING SHUTTER CORRECTION MODEL

Previous translational rolling shutter models [2][7] assumed the scene is of the same depth, so that the models can be sim-
plified to a global affine transform [7] or a row-based correction model [2]. However, this is a strong limitation because there are usually many objects of different depths in any video sequences. Objects of larger depths have smaller rolling shutter distortion, and objects that are closer to the camera suffer from more severe distortion. Taking the scene depth into consideration, we propose an extended rolling shutter model which overcomes the assumption of planar scene. However, like the previous works, we assume the scene is static in this work, i.e. the distortion results from only camera motion, and we ignore any moving foreground objects.

Let us consider a scene point \( X = [X, Y, Z]^T \) in the 3D space. We assume in a video sequence, the camera motion contains planar translation only, and the translation vector can be seen as a function of time \( T(t) \). Then the 2D projected im-
age point, \( X \), can be represented by the perspective projection model in a homogeneous coordinate system

\[
x(t) \sim K(X + T(t))
\]

By expanding (1), we can get

\[
x(t) = \frac{f}{Z}(X + T_x(t)), \; y(t) = \frac{f}{Z}(Y + T_y(t))
\]

We represent the exposure time of a frame as \( S \), which is the inverse of frame rate, and assume frame \( n \) is exposed from \( nS \) to \((n+1)S \). The \( i \)-th row of frame \( n \) is exposed at \((n+\alpha_i)S \), where \( \alpha = 1/R \). \( R \) is the number of image rows. Now we consider two time instants \( t_1 \) and \( t_2 \), \( nS \leq t_1, t_2 \leq (n+1)S \). The 2D projected pixel coordinates are \( (x(t_1), y(t_1)) \) and \( (x(t_2), y(t_2)) \) at \( t_1 \) and \( t_1 \), respectively. According to (2), we can compute the difference between \( (x(t_1), y(t_1)) \) and \( (x(t_2), y(t_2)) \):

\[
x(t_2) - x(t_1) = \frac{f}{Z}(T_x(t_2) - T_x(t_1))
\]

\[
y(t_2) - y(t_1) = \frac{f}{Z}(T_y(t_2) - T_y(t_1))
\]

Without loss of generality, we set \( t_2 = (n+0.5)S \), which is the exposure time of the middle row of the \( n \)-th frame. Then, for each point \( (x_i, y_i) \) that is exposed at time \((n+\alpha y_i)S \) in the rolling shutter frame \( n \), we can compute the new pixel coordinate as if it was exposed at \( t_2 \) by (3) and (4). This process warps the rolling shutter frame \( n \) into a new frame, where all pixels \( (x_i, y_i) \) are shifted to new positions: \( (x_i(t_2), y_i(t_2)) \). \( (x_i(t_2), y_i(t_2)) \) stands for the position where the corresponding 3D scene point of \((x_i, y_i)\) should have been projected if it was exposed at time \( t_2 \). Rewriting (3) and (4) to a more un-
derstandable form, we can write the proposed rolling shutter model as follows:

\[
(x_{gs}, y_{gs}) = (x_{rs}, y_{rs}) + \frac{f}{Z}(T_{x,gs} - T_{x,rs}, T_{y,gs} - T_{y,rs})
\]

where \((x_{rs}, y_{rs})\) is the input distorted rolling shutter pixel coordinate, \( f \) is the camera focal length, \( Z \) is the depth of this point, and \( T_{x,gs} \) and \( T_{x,rs} \) are the camera translation vector at time instant \((n+0.5)S \) and \((n+\alpha y_i)S \), respectively. Each pixel \((x_{rs}, y_{rs})\) is shifted to a global shutter pixel \((x_{gs}, y_{gs})\), removing the rolling shutter distortion. We can also write the translation vectors in (5) as the integration of camera velocity...
\[ v_x(t) \text{ and } v_y(t) \]
\[
x_{gs} = x_{rs} + \frac{f}{Z} \int_{(n+\alpha y_{rs})T}^{(n+0.5)T} v_x(t) dt
\]
\[
y_{gs} = y_{rs} + \frac{f}{Z} \int_{(n+\alpha y_{rs})T}^{(n+0.5)T} v_y(t) dt \tag{6}
\]
Note that \((n+0.5)S\) is just an imaginary exposure time of frame \(n\) for a global shutter. We can set this value to any time instant between \(nS\) and \((n+1)S\). Generally, we set this time instant to be the exposure time of the middle row, so that the missing pixels are more uniformly distributed.

3. MOTION AND DEPTH ESTIMATION

Determining the motion from a video has been an important problem in computer vision. From the proposed rolling shutter model in (6), to correct a distorted pixel \((x_{rs}, y_{rs})\), we must know both the instant camera velocity at its exposure time \((n+\alpha y_{rs})S\), and the velocity at the global shutter exposure time \((n+0.5)T\). Baker et al. [3] showed that the camera motion estimation can be posed as a temporal super-resolution problem. The high-resolution motion vector can be determined from a set of low-resolution constraints. By extending this work, we develop a two-stage motion and depth estimation method, which will be described subsequently.

3.1. Point Correspondences

For accurate motion estimation, plenty of point correspondences from consecutive frames are needed. In our experiment, we use the optical flow field to uniformly sample 30x20 points to use in the motion estimation process.

3.2. Two-Stage Motion and Depth Estimation

We propose a two-stage motion and depth estimation algorithm. The optical flow fields estimated from every two consecutive frames are used in our method. Figure 3 depicts the flow of our algorithm, which takes an alternating minimization strategy to refine the motion and depth estimation.

3.2.1. Motion Estimation

Recall equations (5) and (6) in section 2, which represent the difference between two corresponding points that are projected from the same 3D point at two different time instants \(t_1\) and \(t_2\). We assume there is a pair of matching points in two consecutive frames \(F_n\) and \(F_{n+1}\). This point is at \((x_1, y_1)\) in \(F_n\), and \((x_2, y_2)\) in \(F_{n+1}\). \((x_1, y_1)\) is exposed at time \((n + \alpha y_{1})S\), and \((x_2, y_2)\) is exposed at time \((n + 1 + \alpha y_{2})S\). Representing the camera translation as the integration of camera velocity, we can get
\[
x_2 - x_1 = \frac{f}{Z} \int_{(n+\alpha y_{1})S}^{(n+1+\alpha y_{2})S} v_x(t) dt \tag{7}
\]
where \(f\) is the camera focal length, \(Z\) is the depth of this point, and \(v_x(t)\) and \(v_y(t)\) is the camera velocity in the x and y directions, respectively. Eq. (7) and (8) serve as the low-resolution constraints, and we treat \(Z\) as a known value.

To solve for \(v_x(t)\) and \(v_y(t)\), we have to discretize the continuous motion function in time domain to \(\vec{v} = [v_x, v_y]^T\). For example, we uniformly take 20-30 temporal samples during \(S\), the exposure period for a frame, and assume the motion is constant within a temporal sample. The number of temporal samples determines the resolution of the estimated motion vector.

Second, we would formulate the integration of velocities as a matrix-vector multiplication form. We construct a matrix \(A\), called integration matrix, in which \(A(i,:)\vec{v}\) represent the integration of velocity between the \(i\)th pair of matching points. According to the row number of the two matching points \(y_1\) and \(y_2\), we can compute the temporal samples where they lie in. The elements of \(A(i,:)\) between the two samples are set to 1, and the head and tail samples \(A(i, sample_{y_1})\) and \(A(i, sample_{y_2})\) are determined by the number of rows they cover in the sample, which is between 0 and 1. Third, we can formulate the discretized version of (7) and (8) into a minimization problem: \(\min_{\vec{x}} \| A\vec{x} - \vec{b}\|\) \(\vec{x}\) is the motion vector, and \(\vec{b} = [b_x, b_y]^T\) is a vector where \(b_x(i) = \frac{Z(i)}{f} (x_2^i - x_1^i)\) and \(b_y(i) = \frac{Z(i)}{f} (y_2^i - y_1^i)\) representing the \(i\)th matching pair. As described by Baker et al. the camera motion should be a smooth curve. We add an L2-norm regularization term on the gradient of the motion vector to our objective function so that is can solved fast by the linear least squares method. The resulted objective function takes the following form:
\[
J_x = \frac{1}{2} \| A\vec{x} - \vec{b}\|^2 + \frac{1}{2} \lambda_v \| G\vec{x}\|^2 \tag{9}
\]
where \(G\) is a discrete differentiation matrix that computes the gradient of \(\vec{x}\). By taking the derivative of eq. (9), we can get a closed form solution for \(\vec{x}\), which can be easily computed using the least squares method.
3.2.2. Depth Estimation

We estimate the depths of the feature points also from eq. (7) and (8), but treat the motion vector as known, and the depth $Z$ is the unknown vector to be estimated. Actually, eq. (7) and (8) already give $Z(i)$ a closed-form solution:

$$Z(i) = \frac{f}{x_2^i - x_1^i} A(i, :) v_x^i = \frac{f}{y_2^i - y_1^i} A(i, :) v_y^i$$

(10)

We can still use the least squares method to solve for $Z$ for higher stability:

$$J_z = \frac{1}{2} \left\| \begin{pmatrix} \text{diag}(x_2^i - x_1^i) \\ \text{diag}(y_2^i - y_1^i) \end{pmatrix} Z - f A \vec{v} \right\|^2$$

(11)

The following step-by-step procedure details our two-stage algorithm for camera motion and depth estimation:

Stage 1:
1. Initialize the depths of all feature points to a constant.
2. Solve for $\vec{v}_1^i$ using $Z_0$ with eq. (9).
3. Solve for $Z_1$ using $\vec{v}_1^i$ with eq. (11).

Stage 2:
1. Refine the motion vector $\vec{v}_1^i$ by using the updated depth map with eq. (9).
2. Update the depths of all pixels by using eq. (10).

The core of the proposed algorithm is that we use a limited-number of feature points to compute and refine the camera motion, and the depth maps can be faithfully recovered from the estimated camera motion.

4. EXPERIMENTAL RESULT

To demonstrate the effectiveness of our method, we experiment on the synthetic dataset provided by [4] for quantitative evaluation. We choose the synthetic video for the pure translational motion [4] for our experiment.

Figure 4 depicts the experimental results on some frames in the synthetic video. From the estimated depth maps shown in the third row, we can see the depth maps are mostly correct, but there are problems in the areas with small or thin objects, such as the utility poles in this case. This is mainly due to the smoothness assumption imposed in the optical flow computation. Besides, the incorrect optical flow in less textured regions such as the sky and ground also affects the estimated depths. By comparing the residue images for the rectification with and without using the depth information, it is obvious that the proposed algorithm which incorporates the depth information provides superior results. In addition, we compare their correction accuracies by using the same evaluation scheme proposed in [4], and the results are shown in Figure 5. It is obvious that the proposed algorithm with using the depth information provides better correction accuracy.

5. CONCLUSION

In this paper, we present a new rolling shutter correction model that rectifies the distorted video according to the camera motion and depth maps estimated from the video. Previous rolling shutter correction methods were developed under the assumption of planar scene, which may produce unsatisfactory results for scenes with large depth variations. We demonstrate the effectiveness of the proposed rolling shutter correction algorithm through experiments.

The estimated depth maps by using our algorithm may be inaccurate due to errors in the optical flow estimation. How to obtain accurate optical flow as well as depth map estimation from rolling shutter videos is a direction of future research.
6. REFERENCES


