Computer Vision

Spring 2018

Unrolling the Shutter

Submitted By:

Jyotish P
20161217

Anjan Kumar
201501238

Contents

1 Introduction

Row-wise exposure delay present in some cameras cause distortions (rotations and translations) in images called Rolling Shutter distortions. Most existing methods use a series of frames or scene specific information to correct a Rolling Shutter (RS) distorted image. But our method requires only a single image to undistort it. This method employs CNNs to predict row wise exposure delays and undistort the image geometrically. We also experimented on an end to end network that predicts the flow of pixels from distorted image to undistorted image.

2 What is Rolling Shutter?

2.1 Theory

Rolling shutter is a method of image capture in which a still picture (in a still camera) or each frame of a video (in a video camera) is captured not by taking a snapshot of the entire scene at a single instant of time but rather by scanning across the scene rapidly, either vertically or horizontally. Hence not all parts of the image are captured simultaneously at the same instant.

2.2 Distortions Caused

Since not all parts of the image are captured simultaneously, if the camera is moving, same pixel can be captured twice and one pixel might not be captured at all. This causes unwanted distortions.

2.2.1 Mathematical Model

In RS cameras, the final image formed is a result of the integration over different intervals of the camera trajectory. Let the exposure intervals for two successive scanlines is offset by the line delay t_d. Let the final rolling shutter image for a particular

row be I_r, where r is the row number. The final image can be written as $I_{RS} = [I_1, I_2, ... I_R]$

Intensity at point p_i on a row r can be given by

$$I_r(p_i) = \frac{1}{t_e} \int_{r.t_d}^{r.t_d+t_e} I_r(w(T(-tf_v), o_{i,t}))\ dt$$

where t_e is the time of exposure, f_v is a vector of linear and angular velocities, T is the SE(3) transformation, w is the warping function, $o_{i,\,t}$ is the 3D point corresponding to p_i.

2.2.2 Intuitive Examples for Distortion

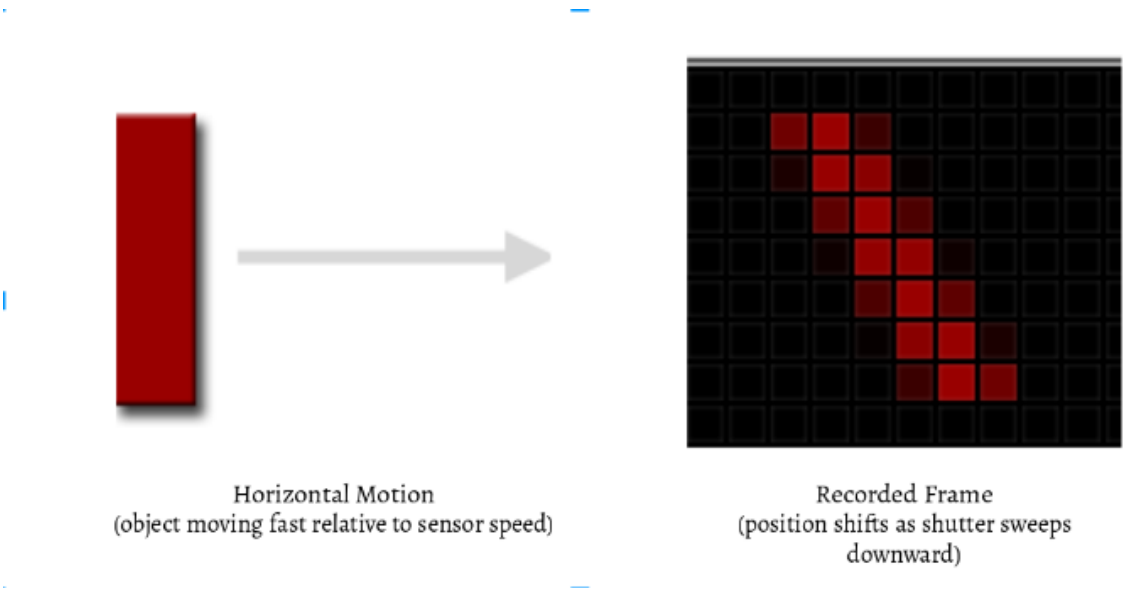


Figure 1: Rolling Shutter Effect

Figure 2: Example of Rolling Shutter Effect

3 Undistortion Goals

RS correction is essentially local image warping undoing the geometric distortion. No new information need to be learnt. Hence theoretically, it is sufficient to learn the trajectory or motion information of the camera. Thus both methods we try in this project try to predict the warping rather than the image.

3.1 Visually Disturbing Causes

Below figure illustrates the local distortions produced by different types of RS motion. We can classify the distortions into roughly four types.

I Vertical curvature [I]

II Vertical stretch/shrinking [II]

III Horizontal curvature [III]

IV Vertical scale change [IV]

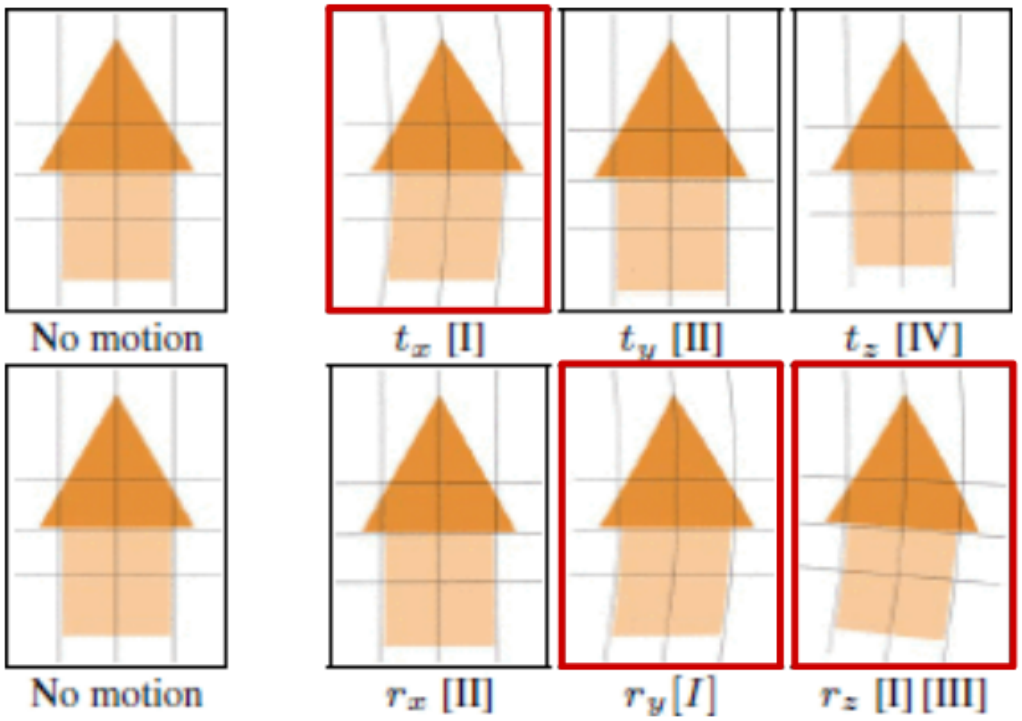


Figure 3: Distortions caused due to different translations and rotations. Red outlined boxes are most visually disturbing and hence we target to correct these.

3.2 Trajectory Modeling

We observe that humans are reactive to only [I], [II], [III] distortions. Distortions in [IV] are negligible and can be safely ignored. We also approximate the effect due to r_y as that caused by t_x. Therefore, we consider only the motions t_x and r_z in our model and ignore the others.

4 Methods to Undistort Rolling Shutter Effect

4.1 Geometric Methods

Very few works concentrate on correcting RS effect in a single image

a. Correcting rolling-shutter distortion of CMOS sensors using facial feature detection

b. From Bows to Arrows: Rolling Shutter Rectification of Urban Scenes

c. Rolling Shutter Motion De-blurring

In the absence of Motion blur, a and b correct faces and urban scenes using respective features. In this work the author tries to correct both using a common representation. c corrects the scene using blind de-blurring. It estimates the image trajectory and uses Gauss Newton optimization to solve for the trajectory. One drawback of it is that it can't handle in plane rotations which are very common.

4.2 Drawbacks of Geometric Methods

Traditional methods need different features for each type of image.

1 Facial features to un-distort images focusing human faces.

2 Curves to un-distort urban images.

These methods are tailored for specific image classes and are thus heavily dependent on the extraction of their respective scene-specific features.

4.3 Advantage with Neural Networks

By using spatial convolution filters to learn features, we can make one model that can learn the features which are not scene specific.

5 Undistorting Rolling Shutter using CNNs

5.1 Pipeline

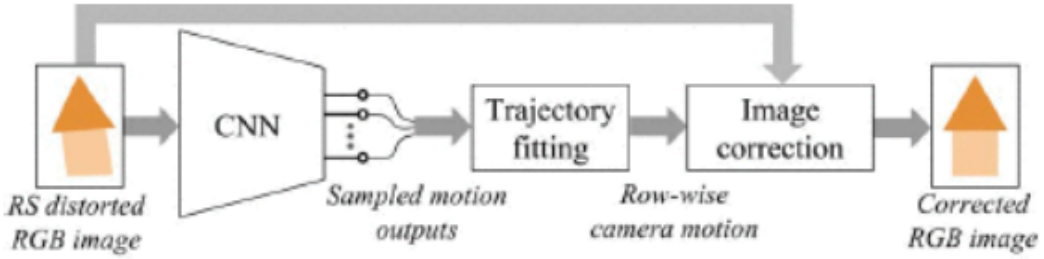


Figure 4: Pipeline of our proposed model

There are mainly three parts in this method,

1 Neural network for camera motion estimation.

2 Trajectory fitting to get row-wise motion.

3 Image correction using the estimated camera motion.

The input to our method is a single RGB RS-distorted image, and the output is the corresponding corrected image.

5.2 Dataset

5.2.1 Generation

We use random third-degree polynomials as synthetic camera trajectories to generate training and testing data. Each trajectory is a set of two 256-length vectors (for row-wise motion), one each for translation and rotation.

5.2.2 Chessboard Class

We take different horizontally and vertically translated versions of a 16-square chessboard image as our basic data. We then apply synthetic RS motions over these images to populate our full training set.

5.2.3 Urban scene class

We build the clean urban scene data by combining the building images in Sun, Oxford and Zurich datasets. Each clean image is distorted with 150 random camera trajectories giving us approximately 300,000 labeled images. We also randomly flip the original images left-right before applying motion distortion.

5.2.4 Face class

We use face images from the Labeled Faces in the Wild (LFW) face dataset. The training set consists of faces of 5000 persons at different poses with 50 motions applied on each face, thereby making the size of training data as 250,000. We choose 200 faces (different from that of training) for testing having different camera motions applied on each of them.

5.3 Vanilla CNN

5.3.1 Architecture

It uses standard convolutional and pooling layers, in which square kernels extract and combine local information from the RS image to deduce the camera motion. Out of the seven layers, the first four convolutional layers consist of square filters, the outputs of which are passed on to ReLU units followed by max-pooling over 2 $\times$ 2 non-overlapping cells. The last three are fully connected layers; the first of the three uses Tanh, the second HardTanh.

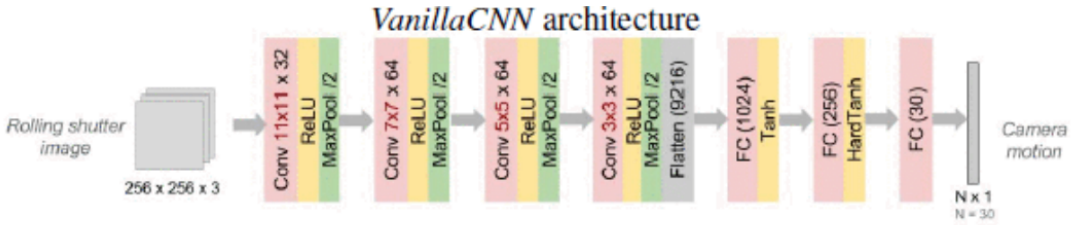


Figure 5: Vanilla CNN Architecture

5.4 Row-Column CNN

We observe the following,

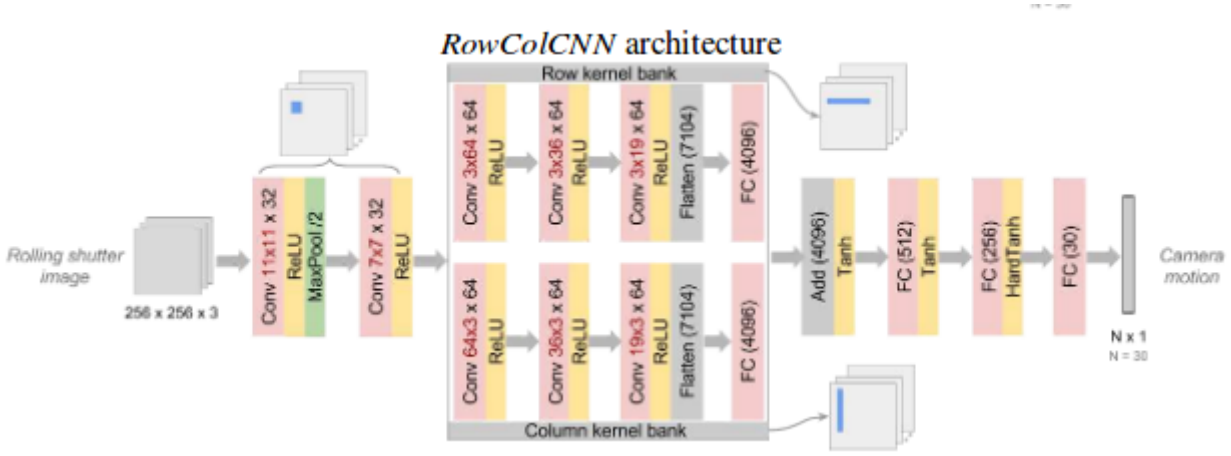
1 Temporal motion information is present along image columns.

2 Information from image rows helps to reinforce row- wise motion constancy.

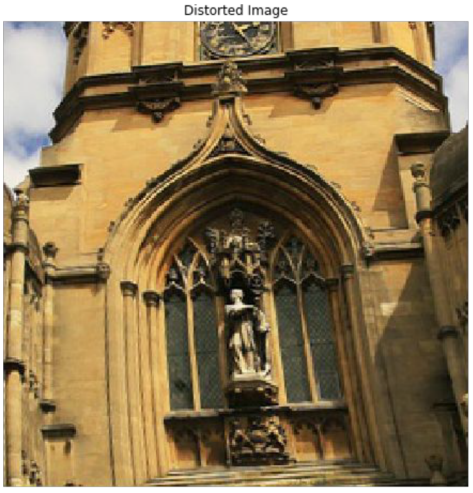
3 Rotation can be better estimated if information from image rows are extracted earlier since it affects left and right areas of an image row differently.

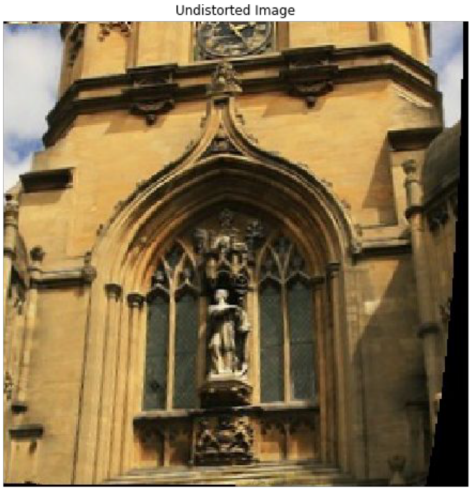
Hence, we branch out VanillaCNN after feature extraction from the initial square convolutional layers into two banks. The column kernel bank employs filters whose effective support spans longer along the column, while the row kernel bank employs row-oriented filters. Both these banks extract locally oriented information and combine them in their own fully connected layers.

5.4.1 Architecture



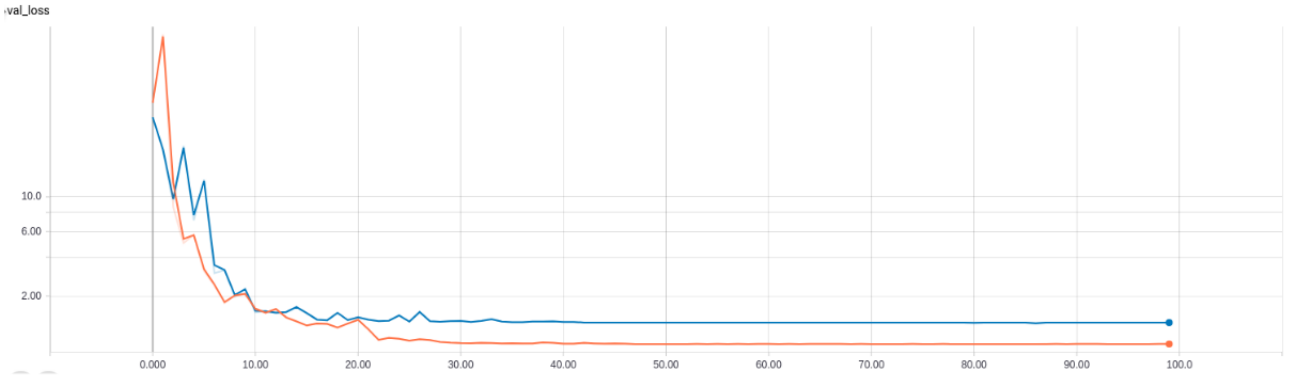
5.4.2 Undistortion Results





5.4.3 Advantage over Vanilla CNN

Validation Loss for both architectures. Orange line represents Row-Column CNN and Blue line represents Vanilla CNN.



5.5 Trajectory fitting

CNN in previous step outputs two K length vectors that denote transition and rotation. These two vectors are then fit using a third-degree polynomial to obtain the estimated motion values for each row.

5.6 Image correction

Once we estimate the camera motion trajectory, we compute 256 transformations (one for each row of the image). We start with a target image with all zeros and apply row-wise transformation for every pixel in the target image to corresponding image pixel in the source image. The pixel values from source image are obtained using Bi-linear interpolation.

6 Appearance Flow - Extension

This approach is based on the fact that pixels in GS image and RS image are highly correlated. We try to predict the flow of the pixels from GS image to RS image.

6.1 Theory

We predict dense appearance Flow fields for an RS image that specifies how to reconstruct a GS image from the said RS image. Specifically for each pixel i in the GS image, the appearance flow vector f(i) specifies the coordinate at the RS image where pixel value is sampled to reconstruct to reconstruct pixel i.

6.2 Possible benefits

1 It alleviates the perceptual blurriness in images generated by CNN trained with L2 loss. By constraining the CNN to only utilize pixels available in the input image, we are able to avoid the undesirable local minimum obtained by predicting the mean colors around texture/edge boundaries that lead to blurriness in the resulting image.

2 The color identity of the instance is preserved by construction since the synthesized view is reconstructed using only pixels from the same instance.

3 The appearance flow field enables intuitive interpretation of the network output since we can visualize exactly how the target view is constructed with the input pixels

6.3 Components

1 Input RS image encoder – extracts relevant features about the scenes like urban scenes and facial features.

2 Trajectory transformation encoder : maps the specified handshake trajectory to a higher-dimensional hidden representation.

3 Synthesis decoder : Outputs appearance flow field.

6.4 Experimentation with Architecture

The eight equidistant points that we are looking for can be obtained by rotating the point about the Z-axis.

6.5 Experimentation with Hyper-Parameters

The eight equidistant points that we are looking for can be obtained by rotating the point about the Z-axis.

6.6 Observations

The eight equidistant points that we are looking for can be obtained by rotating the point about the Z-axis.