RECTIFYING ROLLING SHUTTER VIDEO FROM HAND-HELD DEVICES

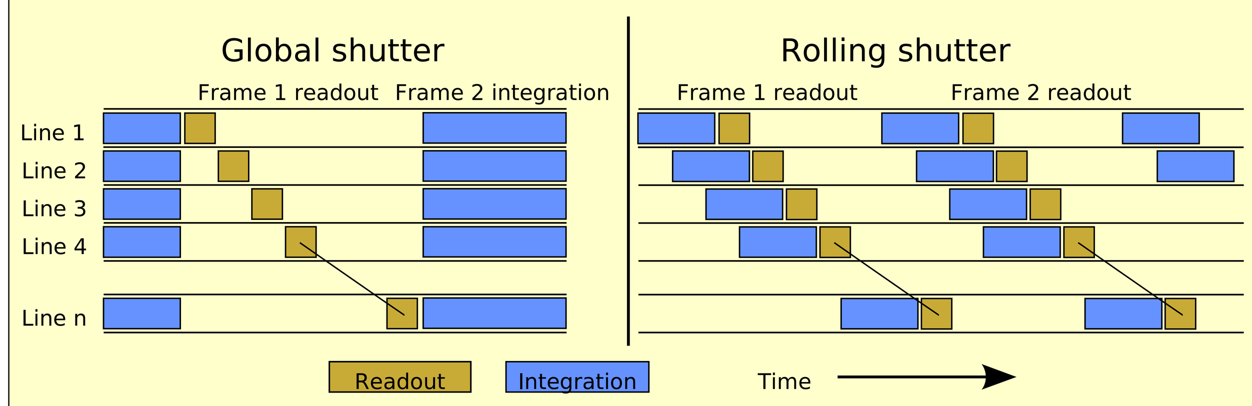
Ringaby E., Forssén P.
{ringaby,perfo}@isy.liu.se, Computer Vision Laboratory, Linköping University

Abstract

Most hand-held video capturing devices sold today (e.g. cell-phones, camcorders, compact cameras, and music players) use CMOS sensors that are prone to rolling-shutter (RS) artefacts. We introduce a method for rectifying video sequences from RS cameras. In contrast to previous RS rectification attempts we model distortions as being caused by the 3D motion of the camera. The camera motion is parametrised as a continuous curve, with knots at the last row of each frame. Curve parameters are solved for using non-linear least squares over inter-frame correspondences obtained from a KLT tracker. We have generated synthetic RS sequences with associated ground-truth to allow controlled evaluation. Using these sequences, we demonstrate that our algorithm improves over to two previously published methods, see [1]. The RS data-set is available on the web to allow comparison with other methods.

[1] P. Forssén, E. Ringaby, Rectifying rolling shutter video from hand-held devices. In CVPR'10, San Francisco, USA, June 2010.

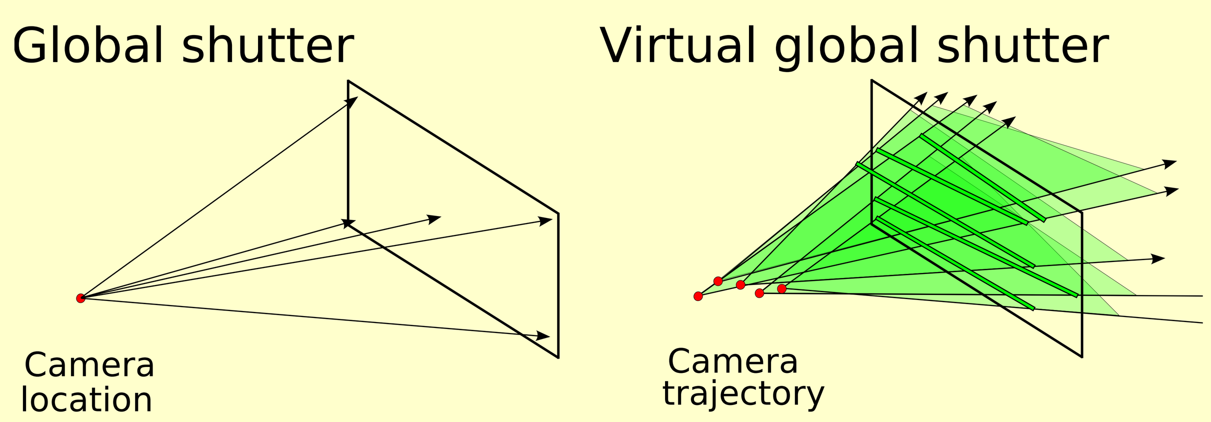
Global vs Rolling shutter



In an RS camera, detector rows are read and reset sequentially and each row is exposed during a slightly different time window. The more conventional CCD sensors use a global shutter (GS), where all pixels are reset simultaneously, and collect light during the same time interval.

Since pixels are acquired at different points in time, motion of either camera or target will cause geometrical distortions in the acquired images.

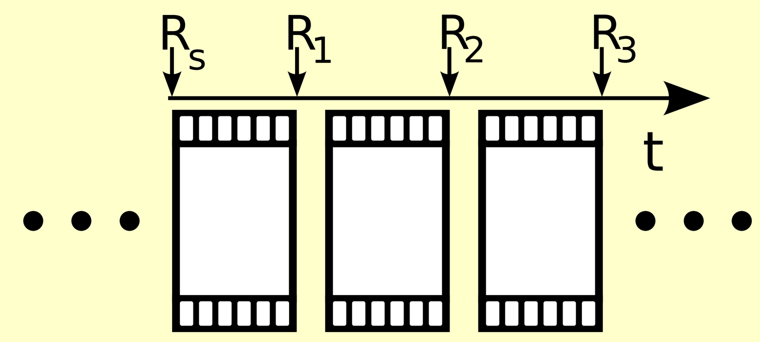
The algorithm creates a virtual global shutter image by rectifying each image row separately.



Algorithm overview

The algorithm finds the dominant camera motion through image measurements and rectifies the video frames:

- Find point correspondences between frames with KLT and outlier-rejection
- Estimate camera motion
- Transform rows/coordinates to a common coordinate system
- Interpolate image with new coordinates



$$\mathbf{x} = \mathbf{K}\mathbf{R}(t_1)\mathbf{X} \quad \mathbf{y} = \mathbf{K}\mathbf{R}(t_2)\mathbf{X}$$
$$\mathbf{x} = \mathbf{K}\mathbf{R}(t_1)\mathbf{R}^T(t_2)\mathbf{K}^{-1}\mathbf{y}$$

X – point in world coordinate system
x – image coordinate in first image
y – image coordinate in second image
K – intrinsic camera matrix
R(t_i) – interpolated rotation from key-rotations $\mathbf{R}_i$

Each rows' coordinate system can be transformed to a reference row $\mathbf{R}_0$ which gives the forward mapping used for rectification.

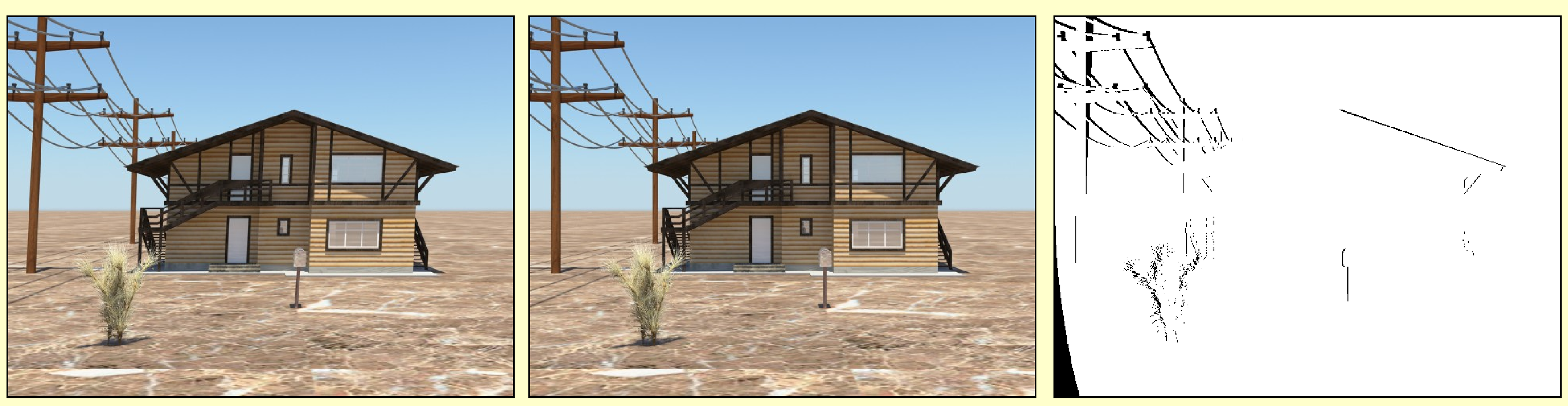
$$\mathbf{x}' = \mathbf{K}\mathbf{R}_0^T\mathbf{R}(\mathbf{x})\mathbf{K}^{-1}\mathbf{x}$$

Synthetic data-set

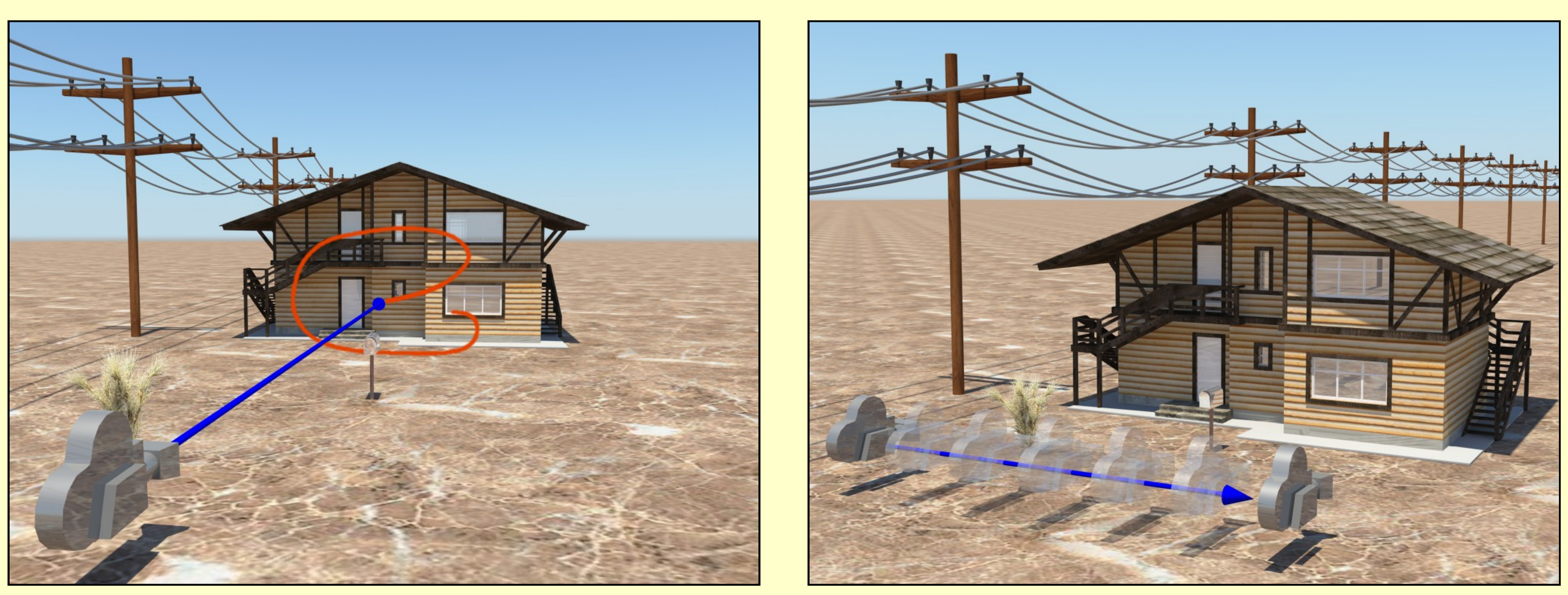
The synthetic data-set has been created with Autodesk Maya in order to do a controlled evaluation of RS-algorithms.

Each synthetic RS frame is created from a GS sequence with one frame for each RS image row. One row in each GS image is used, starting at the top row and sequentially down to the bottom row.

For each frame in the ground truth sequences, we have created masks that indicate pixels that can be reconstructed from the corresponding RS frame and we also provide the ground truth camera trajectory.



Rolling shutter image | Global shutter ground truth | Mask



Examples of camera motions

The data-set and video examples are available at:
http://www.cvl.isy.liu.se/research/rs-dataset/dataset

Optimisation

The camera motion is estimated by iterative minimisation of the symmetric transfer error J.

$$J = \sum_{k=1}^{K} d(\mathbf{x}_k, \mathbf{H}\mathbf{y}_k)^2 + d(\mathbf{y}_k, \mathbf{H}^{-1}\mathbf{x}_k)^2$$

$$\mathbf{H} = \mathbf{K}\mathbf{R}(\mathbf{t}_{1,k})\mathbf{R}^T(\mathbf{t}_{2,k})\mathbf{K}^{-1}$$

$$d(\mathbf{x}, \mathbf{y})^2 = (x_1/x_3 - y_1/y_3)^2 + (x_2/x_3 - y_2/y_3)^2$$

See [1] for a combined rotation/translation model.

Rotations are interpolated with SLERP

$$\mathbf{R}(w) = \mathbf{R}_1(\mathbf{R}_1^T\mathbf{R}_2)^w \qquad w \in [0, 1]$$
$$\mathbf{R}_k = \text{expm}(\mathbf{n}_k)$$

where $\mathbf{n}_k$ is a minimal representation of a rotation with ||**n**|| as the rotation angle around the axis **n** / ||**n**||.

iPhone 3GS | HTC Desire | SonyEricsson W890i

Top: Frames from input videos, bottom: Rectified frames