**Correct Rolling Shutter Distortions From A Single Image**

Most mobile phone (CMOS sensor) cameras employ row-wise light acquisition.

Camera motion even during short exposure causes local geometric distortions known as the **rolling shutter effect**.

**Each image row is associated with a camera pose**

- Top row exposure
- Bottom row exposure
- Camera motion

**Geometry-Based Approach**

A desirable motion trajectory transforms curves to lines on inverse warping

- Line cost
  \[ \sum_{ij} \sum \text{dist}(x_i, y_j) \]
- Angle cost
  \[ \sum \theta_i^2 + \sum \theta_j^2 \]
- Length cost
  \[ \sum d_i^2 \]

- Skew and rotation ambiguities
- Stretch and shrink ambiguities

**Estimate polynomial coefficients** of motion that result in minimum line cost subject to desirability constraints

\[ \hat{\alpha}_{ij} = \arg \min_{\alpha_{ij}} \left\{ \text{Line cost} \right\} \]
subject to

\[ \begin{align*}
\text{Angle cost} & \leq \epsilon_1 \\
\text{Length cost} & \leq \epsilon_2 
\end{align*} \]

**Intermediate results over iterations**

- Distorted
- Corrected

**Learning-Based Approach**

Use a CNN to map the distorted image space to the motion space

- CNN input: Rolling shutter image (256x256 RGB)
- CNN output: Motion values (15 tx samples, 15 rz samples corresponding to equally spaced rows)

Fit a polynomial trajectory to get row-wise camera poses

Correct the distorted image using inverse warping

**Challenges**

- Lack of multiple images to exploit correspondences
- What image features to extract to decode camera trajectory?

**Polynomial motion model**

For \( i \in \{x, y, z\} \):

\[ r_i(y) = \alpha_{i0} + \sum_{j=1}^n \alpha_{ij} \left( \frac{y-1}{M} \right)^j \]

- \( y \) : row index \( \in [1, M] \)
- \( M \) : number of rows
- \( \alpha_{ij} \) : \( j \)-th coefficient for the \( i \)-th axis motion
