Computer Vision Assignment 3

Maitreyee Mulawkar 13D170011 Anand Bhoraskar 130050025
Lokit Kumar Paras 130050047

April 10, 2017

Honor Code : We have not referred or shared our solutions with any other team.

1

Figure 1: Translation shaky video correction using least squares

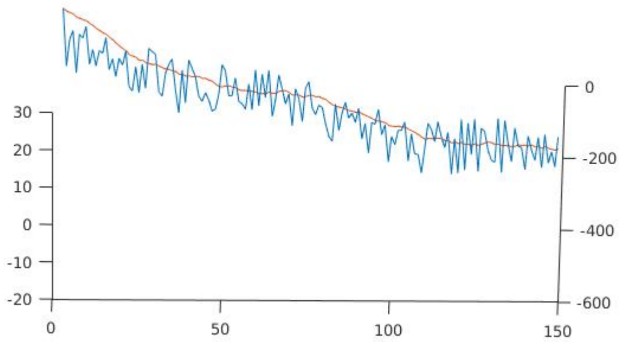


Figure 2: Translation shaky video correction using ransac

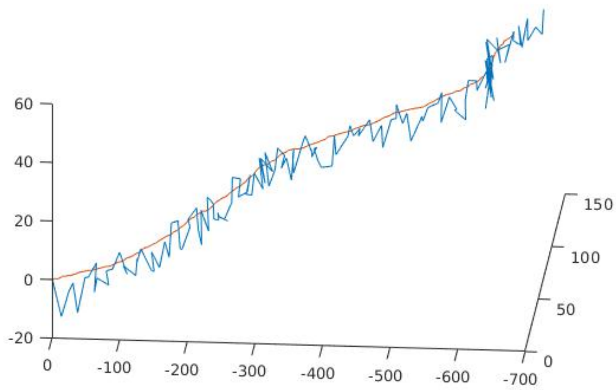


Figure 3: Rigid shaky video correction using ransac - translation

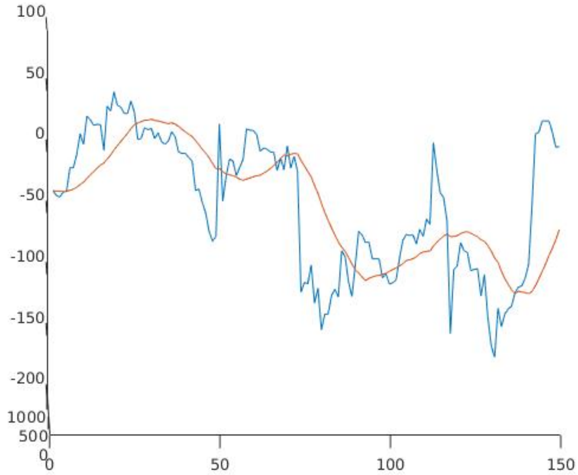


Figure 4: Rigid shaky video correction using ransac - rotation

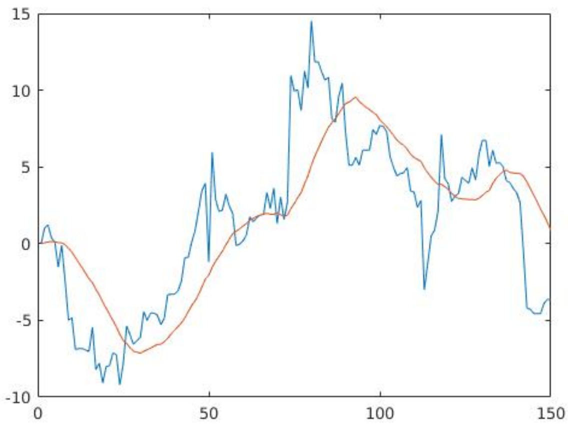


Figure 5: Rigid shaky video correction using ls - translation

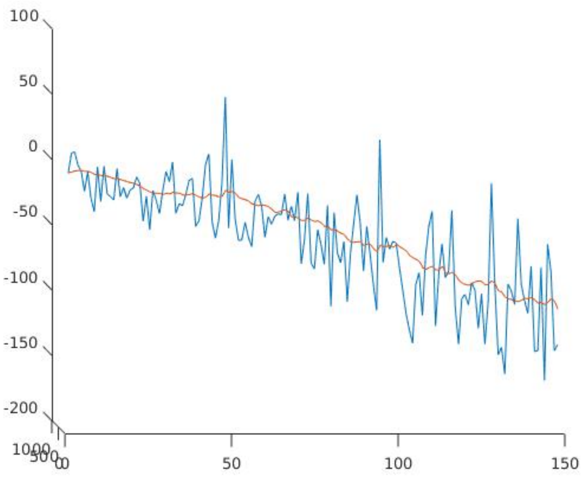
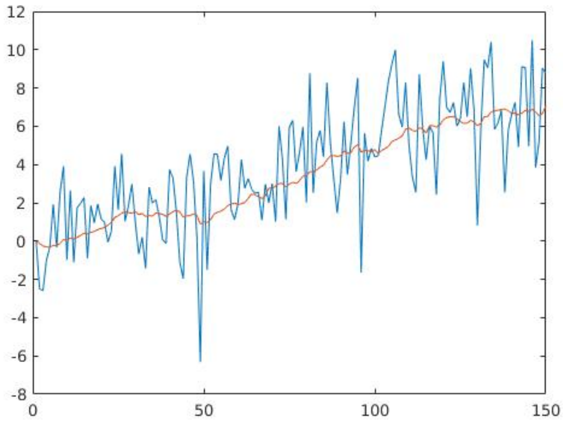


Figure 6: Rigid shaky video correction using ls - rotation



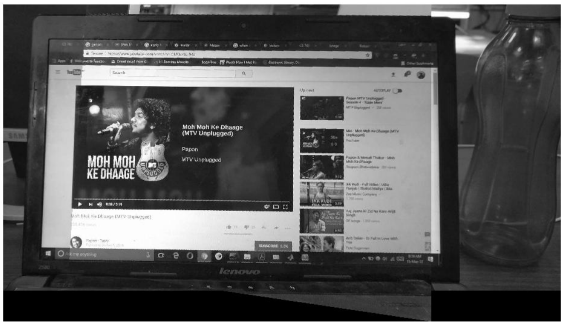
2

Mosaics:

Figure 7: The magnificent hotel taj(new)

Iterations: 1000
Threshold: 0.1

Figure 8: Laptop mosaic



Iterations: 1000
Threshold: 0.1

Normalization before computing homography matrix typically improves the conditioning of the homography matrix because the points would become more evenly distributed.

3

$$J = \sum_{i=1}^{n}\sum_{j=1}^{n}(I_{xij}u_{ij} + I_{yij}v_{ij} + I_{tij})^2 + \lambda((u_{ij+1} - u_{ij})^2 + (u_{i+1j} - u_{ij})^2 + (v_{ij+1} - v_{ij})^2 + (v_{i+1j} - v_{ij})^2)$$

Taking derivatives:

$$\frac{\partial J}{\partial u_{kl}} = \frac{\partial}{\partial u_{kl}}(I_{xkl}u_{kl} + I_{ykl}v_{kl} + I_{tkl})^2 + \lambda((u_{kl+1} - u_{kl})^2 + (u_{k+1l} - u_{kl})^2 + (u_{kl} - u_{kl-1})^2 + (u_{kl} - u_{k-1l})^2)$$

$$\frac{\partial J}{\partial u_{kl}} = 0 = 2I_{xkl}(I_{xkl}u_{kl}+I_{ykl}v_{kl}+I_{tkl})+\lambda(-2(u_{kl+1}-u_{kl})-2(u_{k+1l}-u_{kl})+2(u_{kl}-u_{k-1l})+2(u_{kl}-u_{kl-1}))$$

$$I_{xkl}^2u_{kl} + I_{xkl}I_{ykl}v_{kl} + I_{xkl}I_{tkl} + 4\lambda(u_{kl} - \frac{u_{k+1l} + u_{kl+1} + u_{k-1l} + u_{kl-1}}{4}) = 0$$

Taking $\frac{u_{k+1l}+u_{kl+1}+u_{k-1l}+u_{kl-1}}{4} = \bar{u_{kl}}$,

$$(I_{xkl}^2 + 4\lambda)u_{kl} + I_{xkl}I_{ykl}v_{kl} + I_{xkl}I_{tkl} - 4\lambda\bar{u_{kl}} = 0$$

$$\boxed{(I_{xkl}^2 + 4\lambda)u_{kl} + I_{xkl}I_{ykl}v_{kl} = 4\lambda\bar{u_{kl}} - I_{xkl}I_{tkl}} \tag{1}$$

$$\frac{\partial J}{\partial v_{kl}} = \frac{\partial}{\partial v_{kl}}(I_{xkl}u_{kl} + I_{ykl}v_{kl} + I_{tkl})^2 + \lambda((v_{kl+1} - v_{kl})^2 + (v_{k+1l} - v_{kl})^2 + (v_{kl} - v_{kl-1})^2 + (v_{kl} - v_{k-1l})^2)$$

$$\frac{\partial J}{\partial v_{kl}} = 0 = 2I_{ykl}(I_{xkl}u_{kl}+I_{ykl}v_{kl}+I_{tkl})+\lambda(-2(v_{kl+1}-v_{kl})-2(v_{k+1l}-v_{kl})+2(v_{kl}-v_{k-1l})+2(v_{kl}-v_{kl-1}))$$

$$I_{ykl}^2v_{kl} + I_{xkl}I_{ykl}u_{kl} + I_{ykl}I_{tkl} + 4\lambda(v_{kl} - \frac{v_{k+1l} + v_{kl+1} + v_{k-1l} + v_{kl-1}}{4}) = 0$$

Taking $\frac{v_{k+1l}+v_{kl+1}+v_{k-1l}+v_{kl-1}}{4} = \bar{v_{kl}}$,

$$I_{xkl}I_{ykl}u_{kl} + (I_{ykl}^2 + 4\lambda)v_{kl} + I_{ykl}I_{tkl} - 4\lambda\bar{v_{kl}} = 0$$

$$\boxed{I_{xkl}I_{ykl}u_{kl} + (I_{ykl}^2 + 4\lambda)v_{kl} = 4\lambda\bar{v_{kl}} - I_{ykl}I_{tkl}} \tag{2}$$

Applying cramer's rule which states if:
$a_1x + b_1y = c_1$
$a_2x + b_2y = c_2$
$x = \frac{c_1b_2-b_1c_2}{a_1b_2-b_1a_2}$
$y = \frac{a_1c_2-c_1a_2}{a_1b_2-b_1a_2}$

Hence:

$$u_{kl} = \frac{(4\lambda\bar{u_{kl}} - I_{xkl}I_{tkl})(I_{ykl}^2 + 4\lambda) - (I_{xkl}I_{ykl})(4\lambda\bar{v_{kl}} - I_{ykl}I_{tkl})}{(I_{xkl}^2 + 4\lambda)(I_{ykl}^2 + 4\lambda) - (I_{xkl}I_{ykl})(I_{xkl}I_{ykl})}$$

$$u_{kl} = \frac{4\lambda\bar{u_{kl}}(I_x^2 + I_y^2 + 4\lambda) - 4\lambda I_x(I_x\bar{u_{kl}} + I_y\bar{v_{kl}} + I_t)}{4\lambda(I_x^2 + I_y^2 + 4\lambda)}$$

$$\boxed{u_{kl} = \bar{u_{kl}} - \frac{I_x(I_x\bar{u_{kl}}+I_y\bar{v_{kl}}+I_t)}{(I_x^2+I_y^2+4\lambda)}} \tag{3}$$

$$v_{kl} = \frac{(I_{xkl}^2 + 4\lambda)(4\lambda\bar{v_{kl}} - I_{ykl}I_{tkl}) - (4\lambda\bar{u_{kl}} - I_{xkl}I_{tkl})(I_{xkl}I_{ykl})}{(I_{xkl}^2 + 4\lambda)(I_{ykl}^2 + 4\lambda) - (I_{xkl}I_{ykl})(I_{xkl}I_{ykl})}$$

$$v_{kl} = \frac{4\lambda\bar{v_{kl}}(I_x^2 + I_y^2 + 4\lambda) - 4\lambda I_y(I_x\bar{u_{kl}} + I_y\bar{v_{kl}} + I_t)}{4\lambda(I_x^2 + I_y^2 + 4\lambda)}$$

$$\boxed{v_{kl} = \bar{v_{kl}} - \frac{I_y(I_x\bar{u_{kl}}+I_y\bar{v_{kl}}+I_t)}{(I_x^2+I_y^2+4\lambda)}} \tag{4}$$

We rearrange the equations for the jacobi method as:

$$u_{kl} - (1 - \frac{I_x^2}{I_x^2 + I_y^2 + 4\lambda})\bar{u_{kl}} + \frac{I_xI_y}{I_x^2 + I_y^2 + 4\lambda}\bar{v_{kl}} = -\frac{I_xI_t}{I_x^2 + I_y^2 + 4\lambda}$$

$$v_{kl} + (\frac{I_xI_y}{I_x^2 + I_y^2 + 4\lambda})\bar{u_{kl}} - (1 - \frac{I_y^2}{I_x^2 + I_y^2 + 4\lambda})\bar{v_{kl}} = -\frac{I_yI_t}{I_x^2 + I_y^2 + 4\lambda}$$

Hence, these equations in the form Ax = b where A is a 2MN x 2MN matrix, and x and b are 2MN x 1 vectors. Vector x will contain all the unknowns, i.e. u and v values at a pixel (k,l) and its neighbors. Vector b will contain the (known) terms on the RHS of these equations.

Now, for Jacobi's method:

$$x_i^{t+1} = \frac{b_i - \sum_{j\neq i} A_{kl}x_j^t}{A_{ii}}$$

or:

$$x_i^{t+1} = \frac{b_i + A_{ii}x_i^t - \sum A_{kl}x_j^t}{A_{ii}}$$

Now consider the equation for u_{kl},
Coefficient for $u_{kl} = 1$. The other coefficients are for neighboring cells. Hence:

$$\sum A_{kl} x_j^t = LHS$$

$$\sum A_{kl} x_j^t = u_{kl} - (1 - \frac{I_x^2}{I_x^2 + I_y^2 + 4\lambda})\bar{u_{kl}} + \frac{I_x I_y}{I_x^2 + I_y^2 + 4\lambda}\bar{v_{kl}}$$

$$A_{ii} = 1$$

$$A_{ii} x_i^t = u_{kl}^t$$

$$b_i = -\frac{I_x I_t}{I_x^2 + I_y^2 + 4\lambda}$$

$$u_{kl}^{t+1} = \frac{-\frac{I_x I_t}{I_x^2+I_y^2+4\lambda} + u_{kl}^t - (u_{kl}^t - (1 - \frac{I_x^2}{I_x^2+I_y^2+4\lambda})\bar{u_{kl}}^t)}{1}$$

$$\boxed{u_{kl}^{t+1} = \bar{u_{kl}}^t - \frac{I_x(I_x\bar{u_{kl}}^t + I_y\bar{v_{kl}}^t + I_t)}{(I_x^2+I_y^2+4\lambda)}} \tag{5}$$

Similarly:

$$\boxed{v_{kl}^{t+1} = \bar{v_{kl}}^t - \frac{I_y(I_x\bar{u_{kl}}^t + I_y\bar{v_{kl}}^t + I_t)}{(I_x^2+I_y^2+4\lambda)}} \tag{6}$$

4

We will have to use robust methods such as LMedS instead of least squares. For Horn and Shunk method, or we can set the error term to be:

$$J = \sum |I_x u + I_y v + I_t| + |u_x + u_y + v_x + v_y| \tag{7}$$

To solve this problem, we will have to use gradient descent type methods.

For Lucas-Kanade method, we can similarly use LMeds. To estimate optimal u,v in a region more robustly (towards outliers), we can also change our error function to:

$$J(u, v) = \sum_{i=1}^{N^2} |I_{xi} u + I_{yi} v + I_{ti}|$$

We cannot get a closed form solution for this, hence we will need to use gradient descent.