Rigorous model for quasi-epipolar images generation

The subsequent is the rigorous model implemented for the estimation of a plane rotation starting from a dataset of Tie Points generated,which returns images with null y parallax error and whatever disparity value on the x coordinates for the single TP

$$\begin{cases} x' = cos\theta(x + D) + sin\theta(y) \\ y' = -sin\theta(x + D) + cos\theta(y) + Ty' \end{cases} \tag{1}$$

The above is clearly a non linear model that should be linearized as follows in order to solve it with a least squares approach.

Under the hypothesis of small rotation angle ($\theta \simeq 0$), the following expressions are obtained

$$\begin{cases} x' = (x + D) + \theta(y) \\ y' = -\theta(x + D) + y + Ty' \end{cases} \tag{2}$$

By linearizing with respect to the disparity D in the point $\tilde{D} = Dmed$ the equations became

$$\begin{cases} x' = x + (\tilde{D} + \delta D) + \theta(y) \\ y' = -\theta x - \theta\tilde{D} - \theta\delta D + y + Ty' \end{cases} \tag{3}$$

As $\theta\delta D$ is a second-order term, it is possible to semplify and order the other terms as follows

$$\begin{cases} x' = (x + \tilde{D}) + \delta D + \theta(y) \\ y' = y - \theta(x + \tilde{D}) + Ty' \end{cases} \tag{4}$$

The model can be wrote in a matrix form considering a set of N TPs

$$\underline{y} = \begin{vmatrix} x'_1 \\ y'_1 \\ \vdots \\ \vdots \\ x'_n \\ y'_n \end{vmatrix} \qquad A = \begin{vmatrix} y_1 & 0 & 1 & ... & 0 \\ -(x_1 + \tilde{D}_1) & 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_n & 0 & 0 & ... & 1 \\ -(x_n + \tilde{D}_n) & 1 & 0 & ... & 0 \end{vmatrix} \qquad \underline{x} = \begin{vmatrix} \theta \\ Ty' \\ \delta D_1 \\ \vdots \\ \delta D_n \end{vmatrix} \qquad b = \begin{vmatrix} x_1 + \tilde{D}_1 \\ y_1 \\ \vdots \\ \vdots \\ x_n + \tilde{D}_n \\ y_n \end{vmatrix}$$

This is what is implemented in the ossim-opencv app for quasi-epipolar images generation.

With reference to the above expressions:

- x'_ii is the x-coordinates after the transformation
- y'_i is the y-coordinates after the transformation
- x_i is the x-coordinates before the transformation
- y_i is the y-coordinates before the transformation
- θ is the rotation angle
- Ty' is the translation in y
- D_i is the disparity value in each point
- $\tilde{D}_i$ the approximated value of D
- δD_i the D derivative