Class-Specific Intra-Frame Tracking for Fast Detectors

Fatih Porikli* Oncel Tuzel†

*Mitsubishi Electric Research Laboratories, USA, †Rutgers University, USA

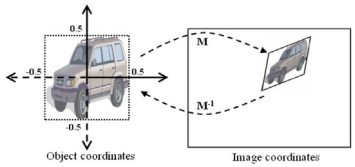
Don't do multi-scale, multi-rotation, multi-warp, affine, etc. object search. Too expensive! Do Intra-frame tracking. We do it with regression. TD rate 5% → 96.7%.

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{pmatrix} \qquad \mathbf{M}_t = \mathbf{M}_{t-1}.\Delta\mathbf{M}_t \qquad \Delta\mathbf{M}_t = f(\mathbf{o}_t(\mathbf{M}_{t-1}^{-1}))$$

$$\exp(\mathbf{m}) = \mathbf{M}$$

$$\rho(\mathbf{M}_1, \mathbf{M}_2) = \|\log(\mathbf{M}_1^{-1}\mathbf{M}_2)\| \approx \|\mathbf{m}_2 - \mathbf{m}_1\|$$

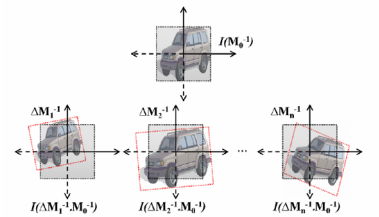
$$f(\mathbf{o}) = \exp(g(\mathbf{o})) \qquad g(\mathbf{o}) = \mathbf{o}^T\boldsymbol{\Omega}$$



$$J_g = \sum_{i=1}^{n} \rho^2\left[f(\mathbf{o}_0^i), \Delta\mathbf{M}_i\right] \qquad \Delta\mathbf{M}_t = f(\mathbf{o}_t(\mathbf{M}_{t-1}^{-1}))$$

$$\left\{\mathbf{o}_0^i, \Delta\mathbf{M}_i\right\}_{i=1\ldots n}$$

$$J_a = \sum_{i=1}^{n} \left\|\log\left(f(\mathbf{o}_0^i)\right) - \log\left(\Delta\mathbf{M}_i\right)\right\|^2$$

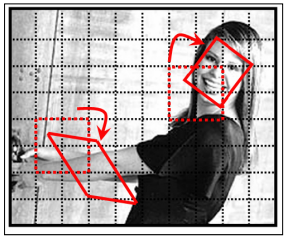


$$\mathbf{X} = \begin{pmatrix} [\mathbf{o}_0^1]^T \\ \vdots \\ [\mathbf{o}_0^n]^T \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} [\log(\Delta\mathbf{M}_1)]^T \\ \vdots \\ [\log(\Delta\mathbf{M}_n)]^T \end{pmatrix}$$

$$J_a = tr[(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})^T(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})]$$

Input: Location of target at time $t-1$ is $\mathbf{M}_{t-1}$ and the current observation is I_t, maximum iteration number is K.

- $k = 1$ and $\mathbf{M}_t = \mathbf{M}_{t-1}$
- Repeat
  - $\Delta\mathbf{M}_t = f(\mathbf{o}_t(\mathbf{M}_t^{-1}))$
  - $\mathbf{M}_t = \mathbf{M}_t.\Delta\mathbf{M}_t$
  - $k = k + 1$
- Until $\Delta\mathbf{M}_t = \mathbf{I}$ or $k = K$



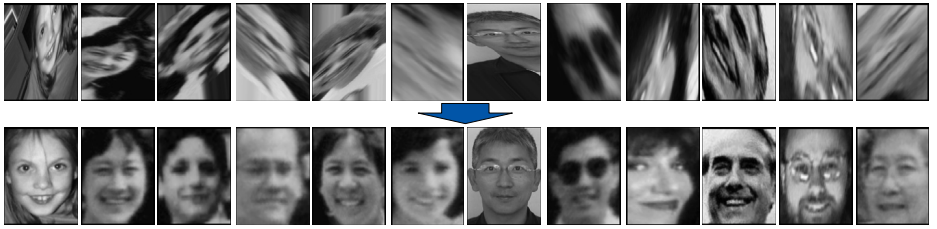
$$J_r = tr[(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})^T(\mathbf{X}\boldsymbol{\Omega} - \mathbf{Y})] + \lambda\|\boldsymbol{\Omega}\|^2 \qquad \boldsymbol{\Omega} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

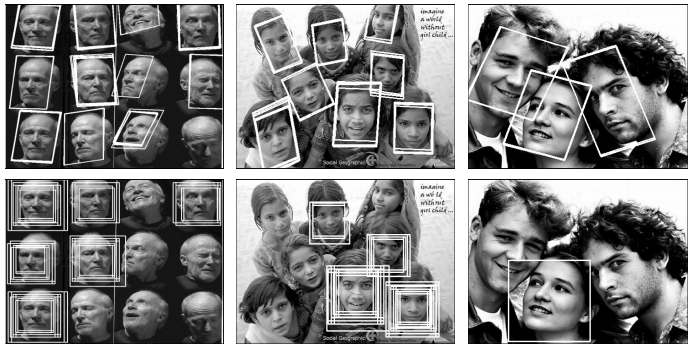
•803 face images from CMU, MIT and MERL datasets.

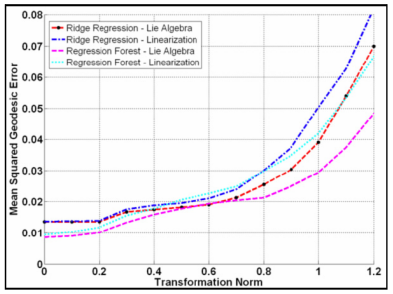
•The dataset is divided into 503 images for training and 300 for testing.

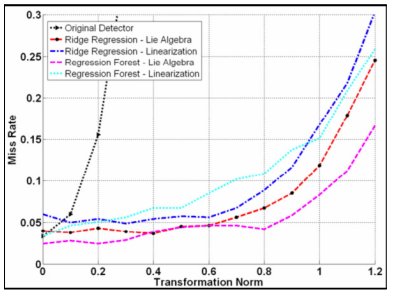
•The training set consists of 25150 samples which are generated by applying 50 transformations having a random norm between 0 to 1.0 to each face image

fatih@merl.com









•For an image of size 320x240, the VJ detector evaluates 58367 locations for translation and scale search, whereas the proposed method evaluates face detector at only 642 locations searched on the affine space.

•Tracker iterations are performed which requires 12840 warping operations.

•Since the warps at each iteration can be performed in parallel we implemented the warping in GPU using NVIDIA GeForce 8800 GTX graphics card and CUDA SDK.

•The search for an image of size 320x240 takes 0.85 and 2.4 seconds with the linear and the regression forest models respectively. On average, Lie algebra based parametrization have 50% less miss rate for regression forest and 24% less for ridge regression, compared to linearization.