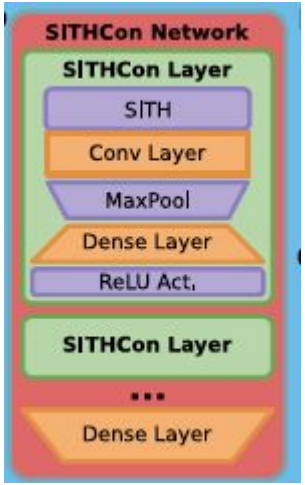
Deep Log-polar networks

Comparison to SITHCon

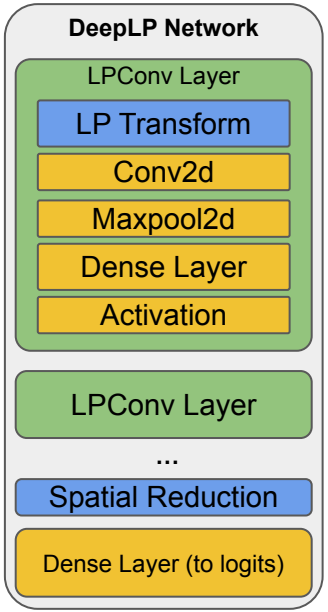
SITHCon

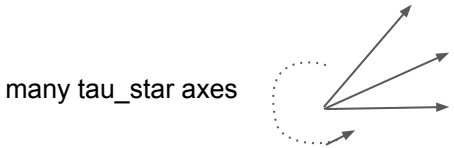
- Principal operation is log-compression of time series to create f(t, i)
  - i indexes tau_star
- Learned 1D convolution over tau_star, then maxpooling



DeepLP

- Principal operation is log-compression of space across rays from a set of angles θ to create f(x, y, i, j)
  - i indexes tau_star
  - j indexes theta
- Learned 2D convolution over tau_star+theta, then maxpooling.





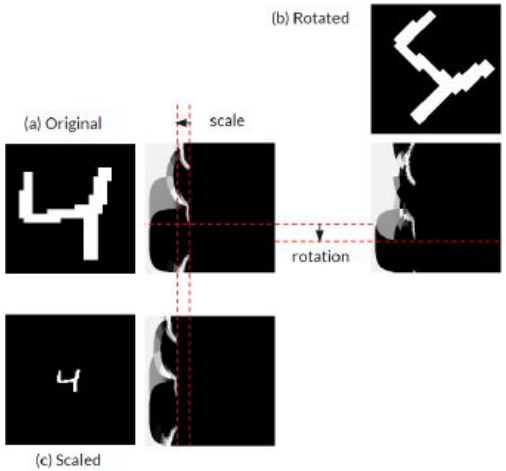
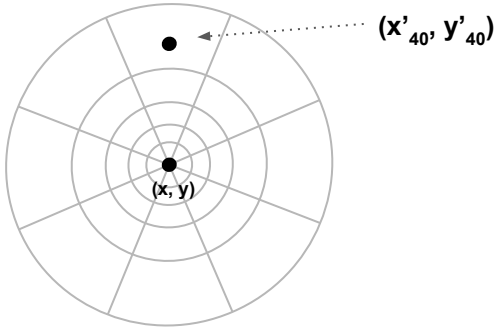


Fig. 2. Rotation and Scaling transformations to a Euclidean image can be read as horizontal and vertical shift respectively, after a log-polar transformation. The log-polar transformation translates rotation and scale in Euclidean images into vertical and horizontal translations (respectively) in the log-polar model.

(figure from Remmelzwaal et al.)

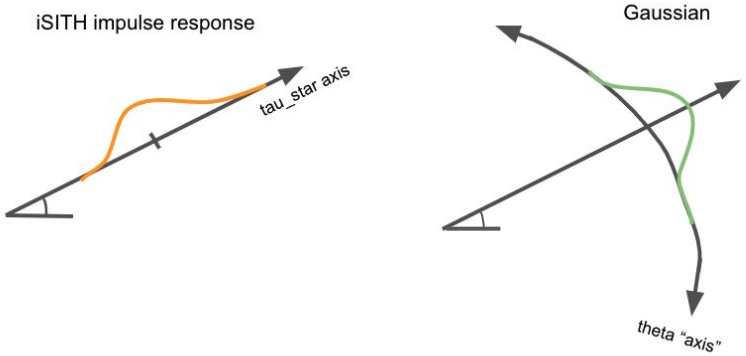
Computing Log-polar transform



Each $(\tau^\star, \theta)$ pair has a corresponding Cartesian coordinate (x'_{ij}, y'_{ij}):

$$x'_{ij} = \tau_i^\star \cos\theta_j = (1+c)^i \tau_{\min} \cos\theta_j$$

$$y'_{ij} = (1+c)^i \tau_{\min} \sin\theta_j$$



Get $f(x, y, i, j)$ by:

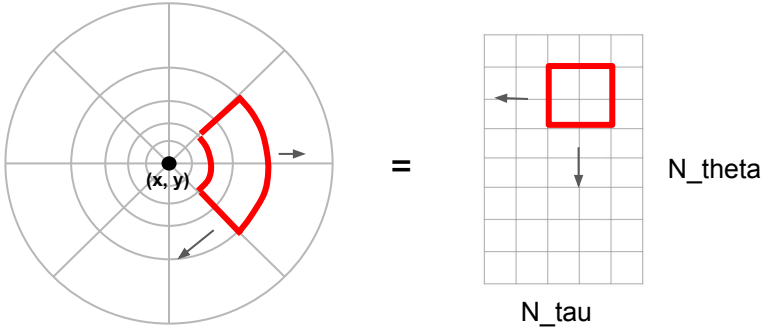
- Taking the inner product with a window: iSITH in $\tau^\star$ direction, a Gaussian in the orthogonal direction, with peak at (x'_{ij}, y'_{ij})
- Could also take the average of all pixels in the "pie slice" centered at (x'_{ij}, y'_{ij}), or use bilinear interpolation to get value at (x'_{ij}, y'_{ij})

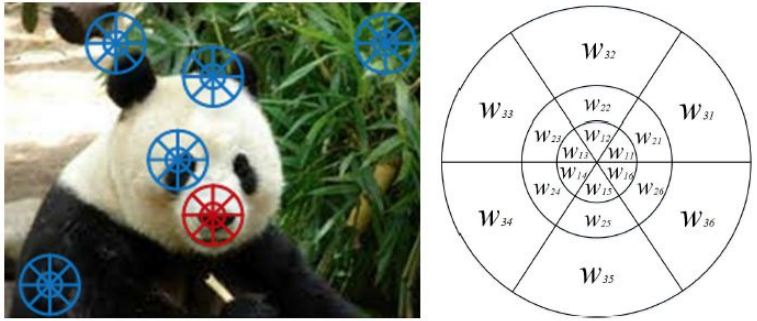
LPConv Layer

- Log-polar coordinate transform is computed at every point in the image, resulting in a tensor of size
  H x W x C_1 x N_tau x N_theta

- After conv and pooling, the size is
  H x W x C_2 x 1 x 1

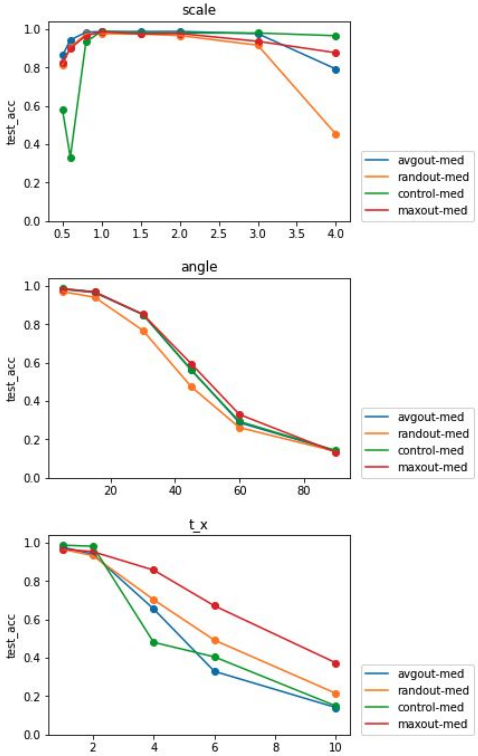
Computation of log-polar feature map at a single pixel (x, y)





(figure from Su et al., a similar approach but without convolution or max-pooling)

Results on MNIST and RotMNIST

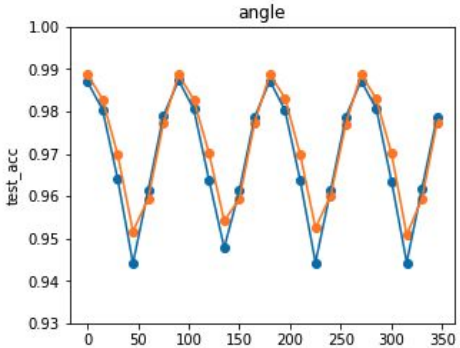


Train small, models (13.5K params) on MNIST, test on rot/scaled/translated images. Comparison of different ways to reduce spatial dimension in last layer (control = choose center pixel). Full max-pooling over only tau dimension.

Note: The rotation-invariant tests are with small ntau and tau_max (to reduce on training time). Scale invariance suffers, though.

Using two models, with 148k and 81k parameters, respectively:

- Training and testing on both RotMNIST gives **98.5%** accuracy for the large model and **97.3%** for the medium model
- Training on only MNIST, then testing on RotMNIST gives **97.8%** and **97.4%** accuracy



rotation invariance out-of-the-box from training on non-augmented MNIST