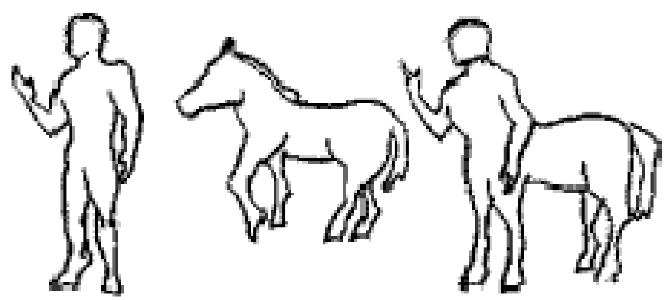
Recognition of Occluded or Distorted Faces Using Partial Similarity



Frank James

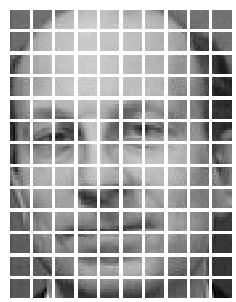
Funded by the NSF REU program
at Utah State University

Overview

- The proposed face recognition algorithm consists of four steps
  1. Partition images into sub-blocks
  2. Calculate local pair wise similarity matrix
  3. Compute the PD measure for a probe image
  4. Identify the image

Partition Images into Sub-blocks

- Partitioned into K (= dim_a/dim_b) non-overlapping sub-blocks
  - dim_a – dimension of whole image
  - dim_b – dimension of sub block
    - 4 × 4 pixels is ideal



- Each sub-block represented by LFV
  - 16 gray scale values (0-255)

Calculate Local Pair Wise Similarity Matrix *D*

- Each element of *D* is the local pair wise distance between a sub block of the probe image ***q*** and the corresponding sub block in another image $\boldsymbol{x}_i$

$$D(q,\ T) = \begin{bmatrix} d(\boldsymbol{q}^1, \boldsymbol{x}_1^1) & \dots & d(\boldsymbol{q}^1, \boldsymbol{x}_N^1) \\ d(\boldsymbol{q}^2, \boldsymbol{x}_1^2) & \dots & d(\boldsymbol{q}^2, \boldsymbol{x}_N^2) \\ \vdots & \vdots & \vdots \\ d(\boldsymbol{q}^K, \boldsymbol{x}_1^K) & \dots & d(\boldsymbol{q}^K, \boldsymbol{x}_N^K) \end{bmatrix}$$

- K = number of blocks
- N = number of images
- Each column represents distances between probe and the *i*th image.

Computing the PD Measure

- PD (partial distance): Threshold divides the set of pair wise distances into two subsets:
  - Similar, $S = \{ k \mid d_{ki} \leq \tau , k = 1,\ldots,K\}$
  - Dissimilar, $F = \{ k \mid d_{ki} > \tau , k = 1,\ldots,K\}$

- Global distance between two face images:
  - PD: $d_{PD}(\boldsymbol{q}, \boldsymbol{x}_i, \beta) = \beta \sum_{k \epsilon S} d_{ki} + (1-\beta) \sum_{k \epsilon F} d_{ki}$
    - Sum of similar and dissimilar blocks.
    - $\beta \in [0,1]$ and balances weight of *S* and *F* blocks.

$$\beta = \min\left(1, \frac{|S|}{|F|}\right)$$

Computing the PD Measure

- cPD (continuous partial distance):
  - Rather than relying on a similarity threshold τ, cPD uses the *golden proportion* to determine which blocks are similar.
- Golden proportion – "whole is to the larger part as the larger part is to the smaller part"
  - Number of blocks to compare is α = (0.618 x total)
- Sort the columns of matrix *D* in increasing order. Smaller distances near top. Sum first α blocks.

$$d_{cPD}(q, \boldsymbol{x}_i) = \sum_{k \le \alpha} d_{ki}$$

Identifying a Probe Image

- Identifying ***q*** is simple.
- Closest match is the image with the smallest partial distance to ***q***.

$$\text{label}(\boldsymbol{q}) = \min_{i=1\ldots N}(d_{PD}(\boldsymbol{q}, \boldsymbol{x}_i, \beta))$$

- d_{PD} is calculated by using one of the previous two formulas.

How to set threshold for PD?

- As seen before, cPD uses α.

- PD is more complicated.
  - Uses *max margin* criterion.
  - Class-dependent thresholds are optimized and learned from “training sets”.

How to set threshold for PD?

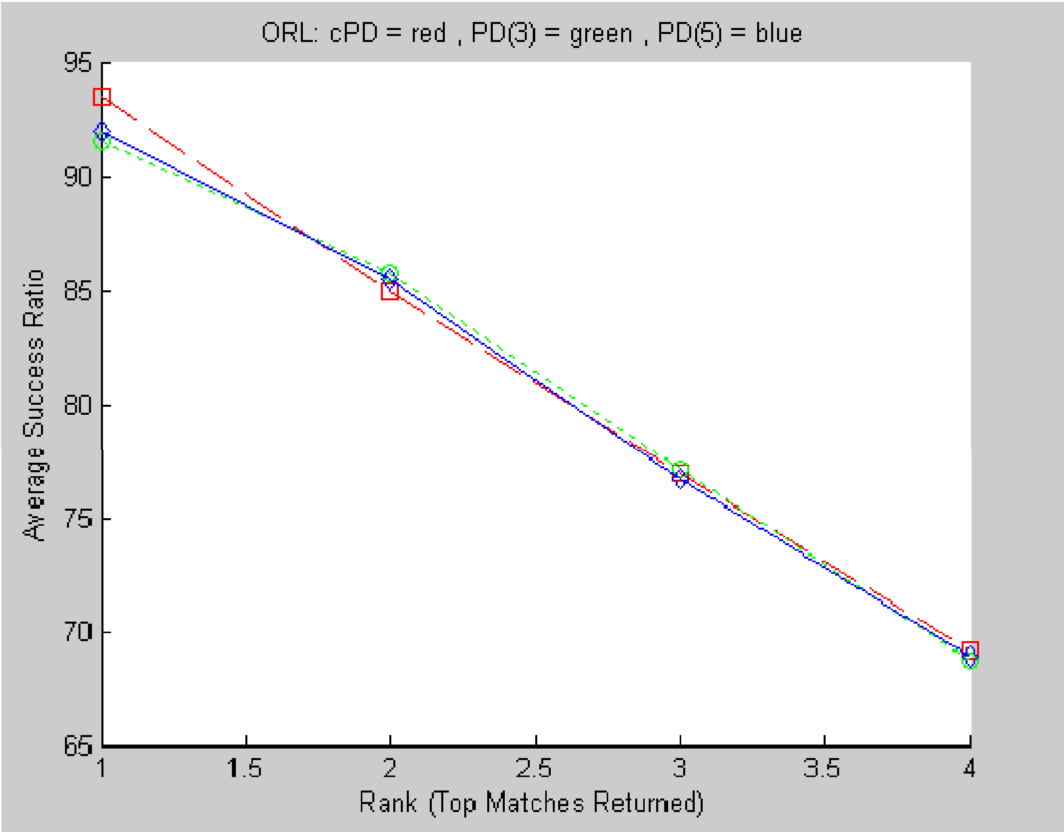
$$\bar{m}_c(X_c, \tau_c) = \frac{1}{|H_c|} \sum_{i \in H_c} \{1(FOO(\boldsymbol{x}_i, \tau_c) = y_i) \times [\min_{j \in \bar{H}_c} d_{\mathrm{PD}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \tau_c) - \min_{j \in H_{c\, j \neq i}} d_{\mathrm{PD}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \tau_c)]\}$$

- If a training image is successfully identified under some trial threshold, the margin is measured as the difference between the distance of the closest match from another class and the distance of the closest match within the class (except for the image itself).
- Compute the average margin value for each class.
- Final class-dependent threshold will be the threshold that gives the largest margin value.

Experimental Results

- Testing the affects of using larger training sets in maximum margin threshold learning.
  - First training set uses 3 images per person.
  - Second training set uses 5 images per person.
  - Each of these training sets learned thresholds are tested on the same test set of 5 images per person.
- Which training set should produce better results?

Experimental Results

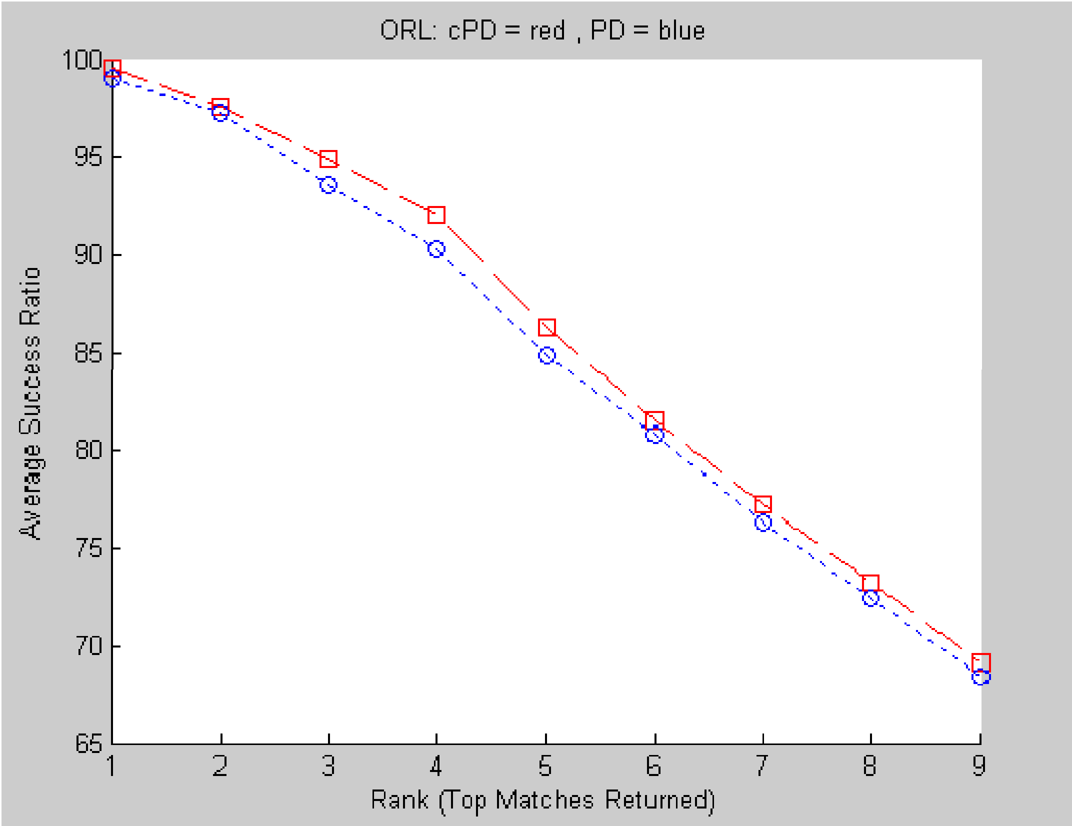


- Remember, cPD does not use learned thresholds. Shown here just for comparison.

Experimental Results

- We’ll try again with a training set of 10 images per person…

Experimental Results



Experimental Results

Rank	cPD(5)	cPD(10)	PD(3)	PD(5)	PD(10)
1	93.5%	99.5%	91.5%	92%	99%
2	85%	97.6%	85.8%	85.5%	97.3%
3	77%	94.9%	77.2%	76.7%	93.6%
4	69.1%	92.1%	68.8%	68.9%	90.4%
5	X	86.4%	X	X	84.9%
6	X	81.5%	X	X	80.8%
7	X	77.3%	X	X	76.3%
8	X	73.3%	X	X	72.4%
9	X	69.2%	X	X	68.4%

Experimental Results

- “That’s great, but I thought this system was supposed to test occluded images? The ORL database contains hardly *any* occlusions!”
- …let’s create some occlusions.

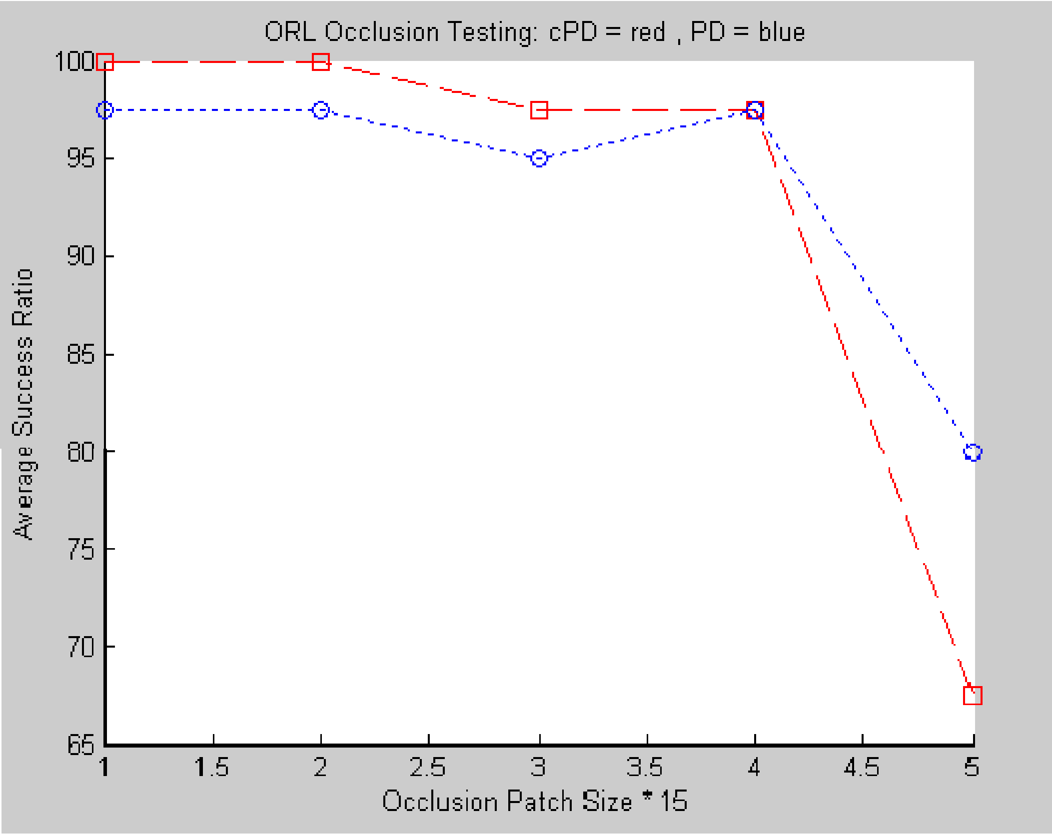
Experimental Results

- One image per person is removed from the database for occlusion probing.
- The leftover 360 images are used as the test set.
- Occlusions of various sizes are randomly placed on the probes.
    - Black patches ranging from (15 x 15) to (75 x 75) pixels are randomly placed on the probe images.
    - NOTE: Largest patch size occludes 55% of the face image.

Experimental Results

- So, how do cPD and PD hold up against occlusions?

Experimental Results



- PD successfully recognizes 80% of probe images whose faces are occluded over 55%.

  Not too shabby!

Future Work

- Combine with a face detection algorithm to achieve a truly robust system.

- Increase run time efficiency so system can be adapted to real time video.
  - Law enforcement and security systems
  - Applications in robotics

Conclusions

- The *continuous partial distance* (cPD) and the *partial distance* (PD) methods held up well against the ORL database.
- The proposed system is robust against face images that are occluded and distorted by varying facial expressions.
- Also tolerates images of angled faces.

Questions?