IMPACT OF POSE AND GLASSES ON FACE DETECTION USING THE RED EYE EFFECT

Yednekachew Asfaw, Bryan Chen and Andy Adler
University Of Ottawa

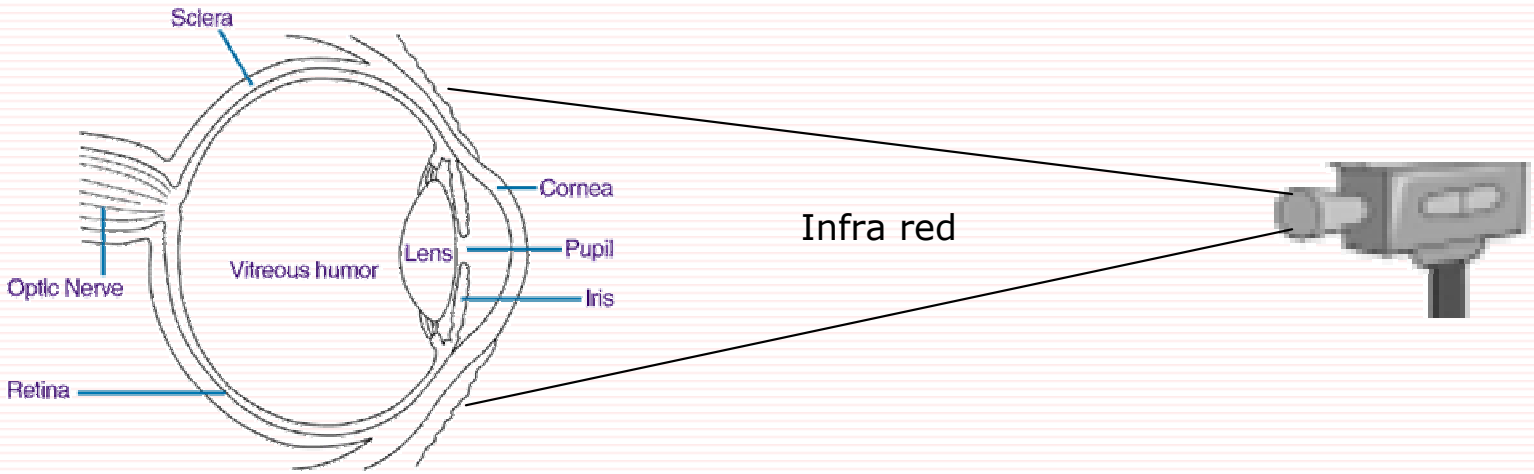
March 2003

Face Detection

- Applications for face detection
  - Surveillance
  - Input to face recognition systems
  - Human computer interface
- Current systems typically rely on processing visible image, and aren’t 100% reliable

Red Eye effect for face detection

- Red eye effect is caused by reflection off the retina due to coaxial illumination



Research Objectives

- Research face detection using red-eye data
- Evaluate and classify impact of variables
  - Fixed distance from camera
  - Pose
  - Glasses
  - Eye color and skin tone

Image Analysis Techniques

- Goal: develop face detection algorithm to account for variables
- Face detection algorithm stages
  - Normalization
  - Localization
  - Blurring
  - Validation

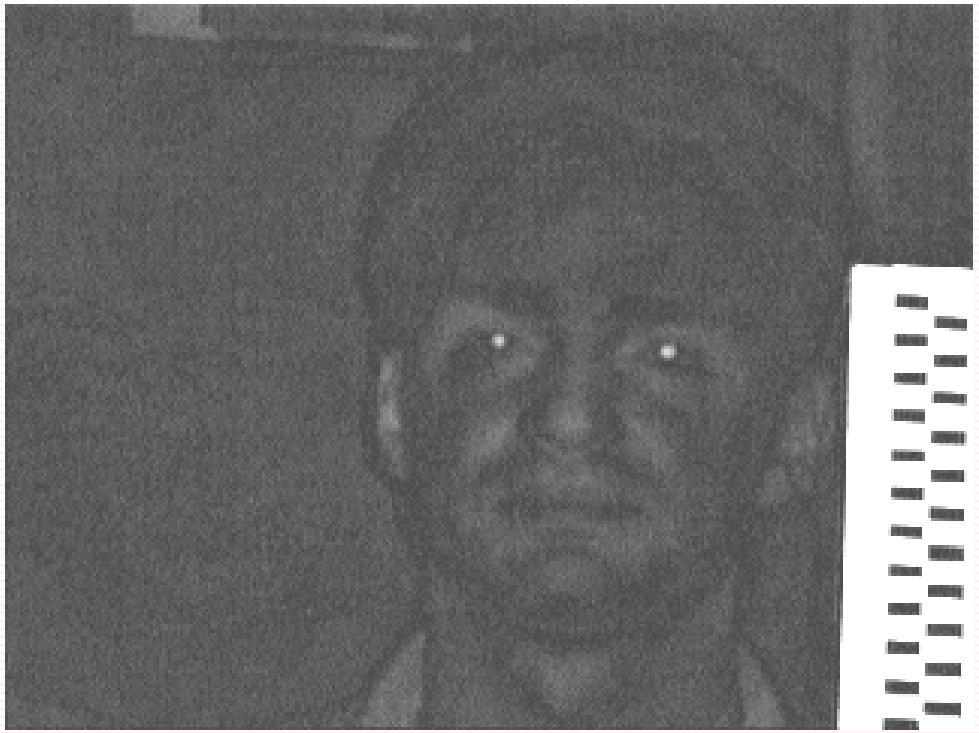
1.Normalization

- Purpose:
  - Normalize overall intensity of the image
- Technique:
  - Morphological Opening from Matlab image processing toolbox
  - Creates a uniform background to image
  - Subtracting background to original produces an image with a normalized intensity

1.Normalization cont’

- Example:

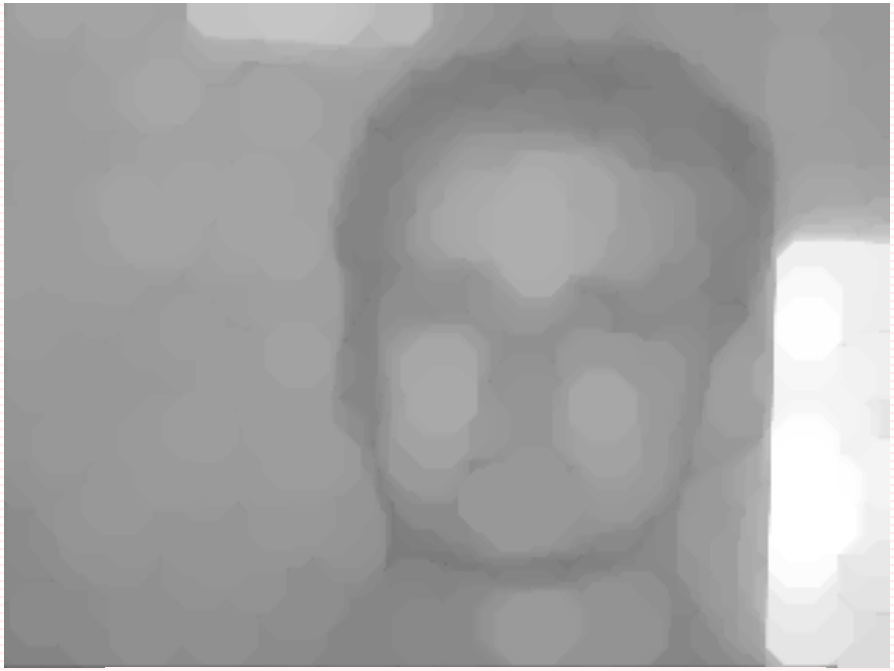
1. Given the original



Original Image

1.Normalization cont’

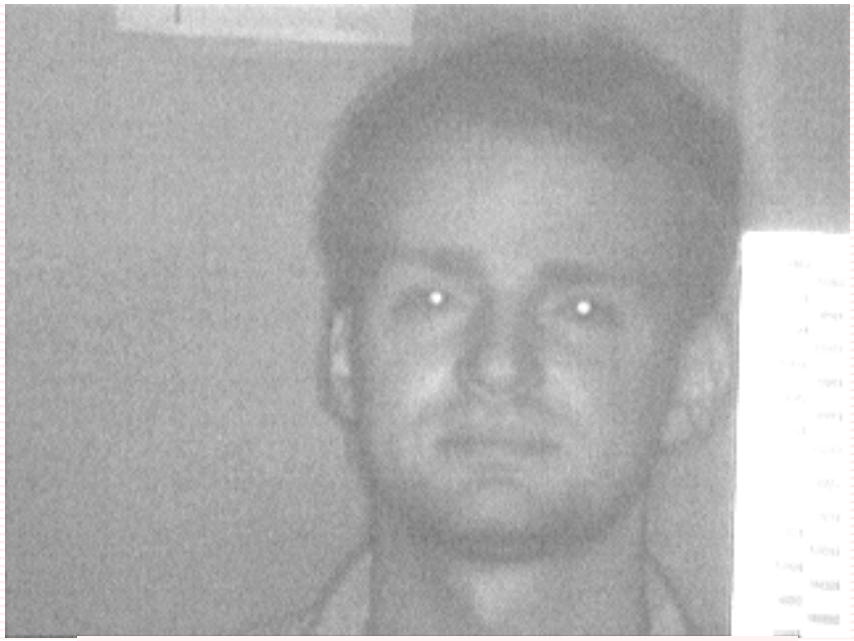
2. Generate a uniform background



Background

1.Normalization cont’

3. Subtract background from original



Original-Background

2.Localization

- Purpose
  - Isolate Possible candidates for pupils
- Technique
  - Based on overall intensity of image
  - Calculate the mean intensity and standard deviation of the image
  - Threshold image based on mean and standard deviation

    Threshold value = mean + stand dev/3

2.Localization cont’

- Example

1. Change Image to B&W by applying Dynamic threshold

Black and White

3. Blurring

- Purpose
  - Blur the image to eliminate background noise
- Technique
  - Using mean filtering
  - Create a sample mean kernel
    - Kernel (3x3 matrix)
  - Carryout a 2D convolution

3.Blurring cont’

- Example

1. Blur using Conv2

Blurred

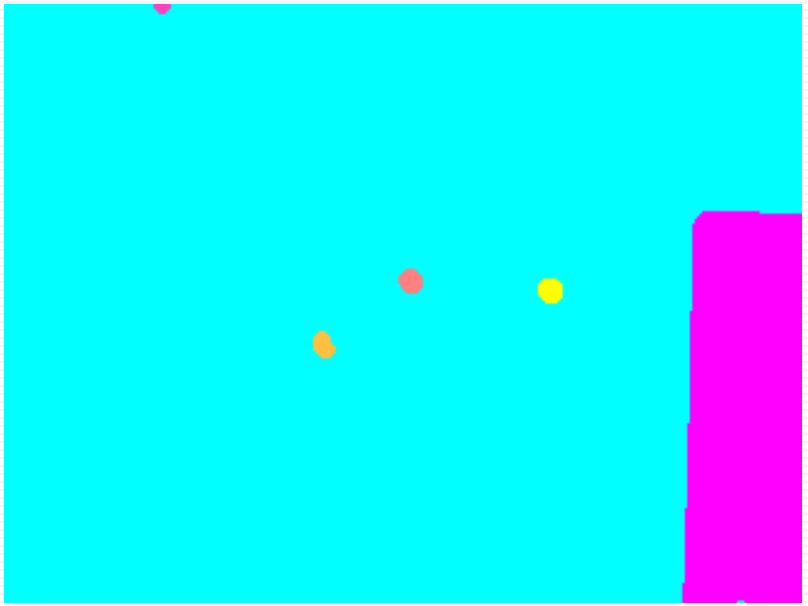
3.Blurring cont’

2. Dilate Image

Dilated

3.Blurring cont’

3. Label all white areas



Labeled

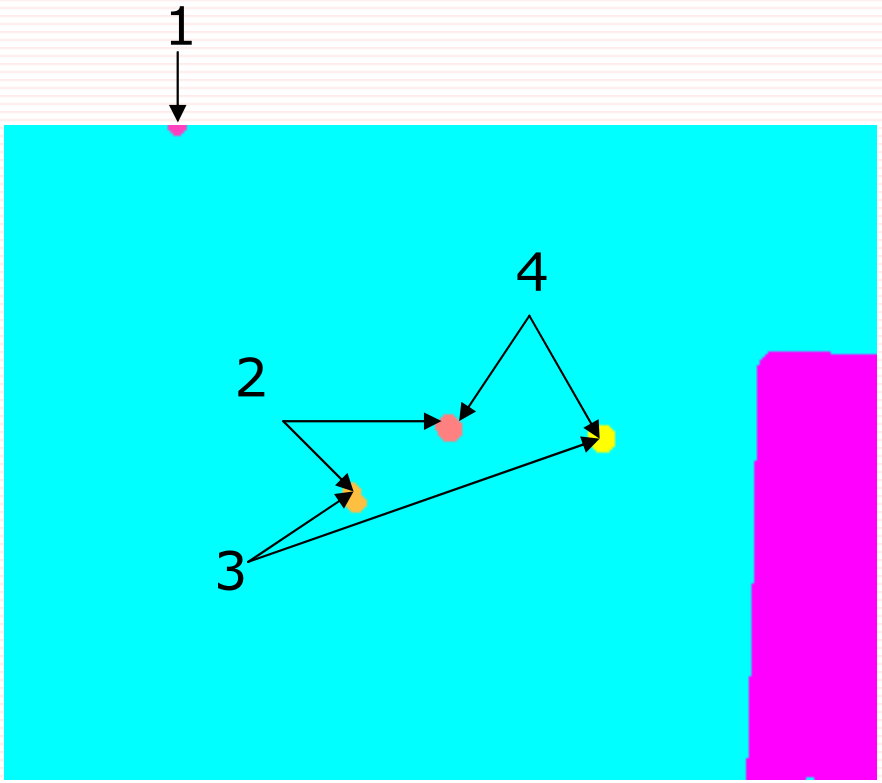
4. Validation

- Purpose
  - To isolate the pupils from possible candidates
- Technique
  - Using the holistic approach:
    - Pupil distance ranges from 40 to 50 pixel
    - Head motion from 0 to 20 degrees
    - Defining Probable Activity Area

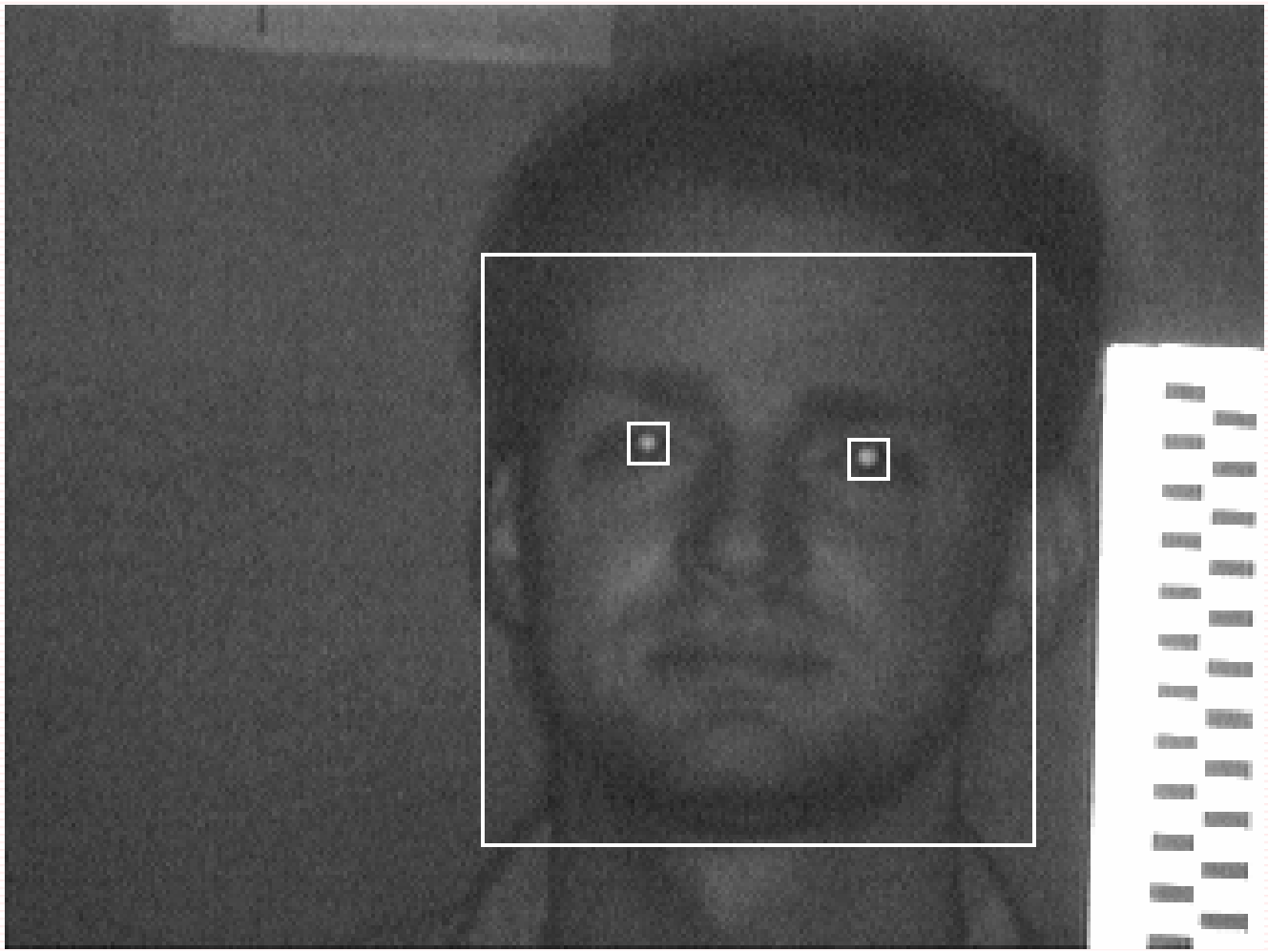
4. Validation cont’

- Example

1. Outside of Active Region
2. Improper angle
3. Improper distance
4. Real Data



Results



Results

- 80-85% detection success
- All experimental variables classified:
  - Current techniques sufficient for locating eyes without glasses

Effect of Variables

- Pose
  - Contribution from cornea of the eye at higher degrees (30 and 45)
- Glasses
  - Has the effect of covering red eye
  - Reflection dependent of the type of glasses on subject
- Skin Tone/Color
  - Effectiveness of red eye detection increase as skin gets darker
  - Less contribution from other facial area as skin gets darker

Effect of Variables

- Eye Color
  - The intensity level of the red eye lower on darker eye colors
- IR placement
  - On axis IR placement produces better result on 0 degree pose

Recommendation

- Research into other image extraction techniques
  - Frame differential
- Use this technique in conjunction with other face detection methods

Abstract

In current image-processing algorithms for face detection performance is not completely reliable, especially in situations with variable lighting, and with low-resolution images. One possible approach to implement face detection is the use of the "red-eye" effect: the reflection produced by human eyes when exposed to co-axial infrared (IR) light. We investigated the effectiveness of the red-eye technique for variability in: skin tone, eye color, pose, angle of IR illumination, scene illumination, and the effect of shine from glasses. Algorithms were developed to detect eye locations from a single IR image. Image processing steps involved: normalization, blurring, dynamic threshold calculation, and candidate eye position validation. Average eye position estimation accuracy approaches 80 to 85 percent.