Development of Optical Computer Recognition (OCR) for Monitoring Fatigue in Space

Fei Yang¹, Nicholas Michael¹, Xiang Yu¹, Dimitris Metaxas¹, David F. Dinges²
¹Center for Computational Biomedicine Imaging and Modeling Center (CBIM), Rutgers University, New Brunswick, NJ, USA
²Unit for Experimental Psychiatry, Department of Psychiatry, University of Pennsylvania School of Medicine, Philadelphia, PA, USA

INTRODUCTION

Fatigue from chronic partial sleep deprivation, circadian misalignment, and work overload is a risk factor for astronaut cognitive performance in space flight. There is a need for techniques that objectively and unobtrusively identify the presence of fatigue on-line, when astronauts are performing critical tasks in space.

Previous researches have found that tracking slow eyelid closures (referred to as PERCLOS) is one of the most reliable ways to detect lapses of attention during critical tasks. We build the Optical Computer Recognition (OCR) system which can monitor alertness in real time providing an early detection of fatigue. By tracking human faces and measuring PERCLOS using inexpensive camera equipment, our system offers a completely unobtrusive way to achieve this requirement.

SYSTEM OVERVIEW

We have developed a system that is capable of real-time tracking of facial landmarks (e.g., eyes, eyebrows, nose, mouth), using statistical deformable models and the KLT tracker. We use the tracked positions of the eyes as a basis for our eye segmentation algorithm, and then further refine it by fitting geometrical templates to the map of pixel-wise likelihood. An expectation maximization (EM) algorithm is used to cluster the pixels in HSV color space, using the first few tracked frames.

FIND COLOR DISTRIBUTION

We perform an expectation maximization algorithm to cluster the pixels in HSV color space.

ALG: Find Skin Color Distribution
1. Crop eye region, convert to HSV color space.
2. Put all pixels in set P.
3. Repeat
4. Compute Gaussian distribution \((\mu, \sigma)\) of all pixels in P.
5. For all pixels in eye region, compute the Mahalanobis distance to \((\mu, \sigma)\).
6. Take 50% pixels with smaller distances to \(\mu\) and update P.
7. Until \((\mu, \sigma)\) converges.

FIT EYE CONTOUR

We define the eye template as two parabolic sections. The parameters are then optimized to best fit the pixel-wise likelihood by using gradient descent method. Denote the parameters of the upper section as \(\theta_1\) and the parameters of the lower section as \(\theta_2\).

- The likelihood is
  \[
  L(\theta | I) = (L(\theta_1 | I) + L(\theta_2 | I))
  \]
  \[
  = \int_{x=x_1}^{x_2} \int_{y=y_1(x)}^{y_2(x)} s(I(x, y)) \, dy \, dx
  \]
- Compute \(\theta_1\) and \(\theta_2\) by fitting parabolic sections to pass the landmarks \((x_1, y_1) \ldots (x_4, y_4)\)
- Compute the derivatives of the likelihood \(L\) to the parameters \(\theta_1\) and \(\theta_2\).
- Compute the derivatives of \(L\) to \(y_1 \ldots y_4\).
- Update \(y_1 \ldots y_4\) by using gradient descent.
- Repeat the above steps until \(y_1 \ldots y_4\) converge.

REFERENCES