Eyelid Position Detection Method for Mobile Iris Recognition

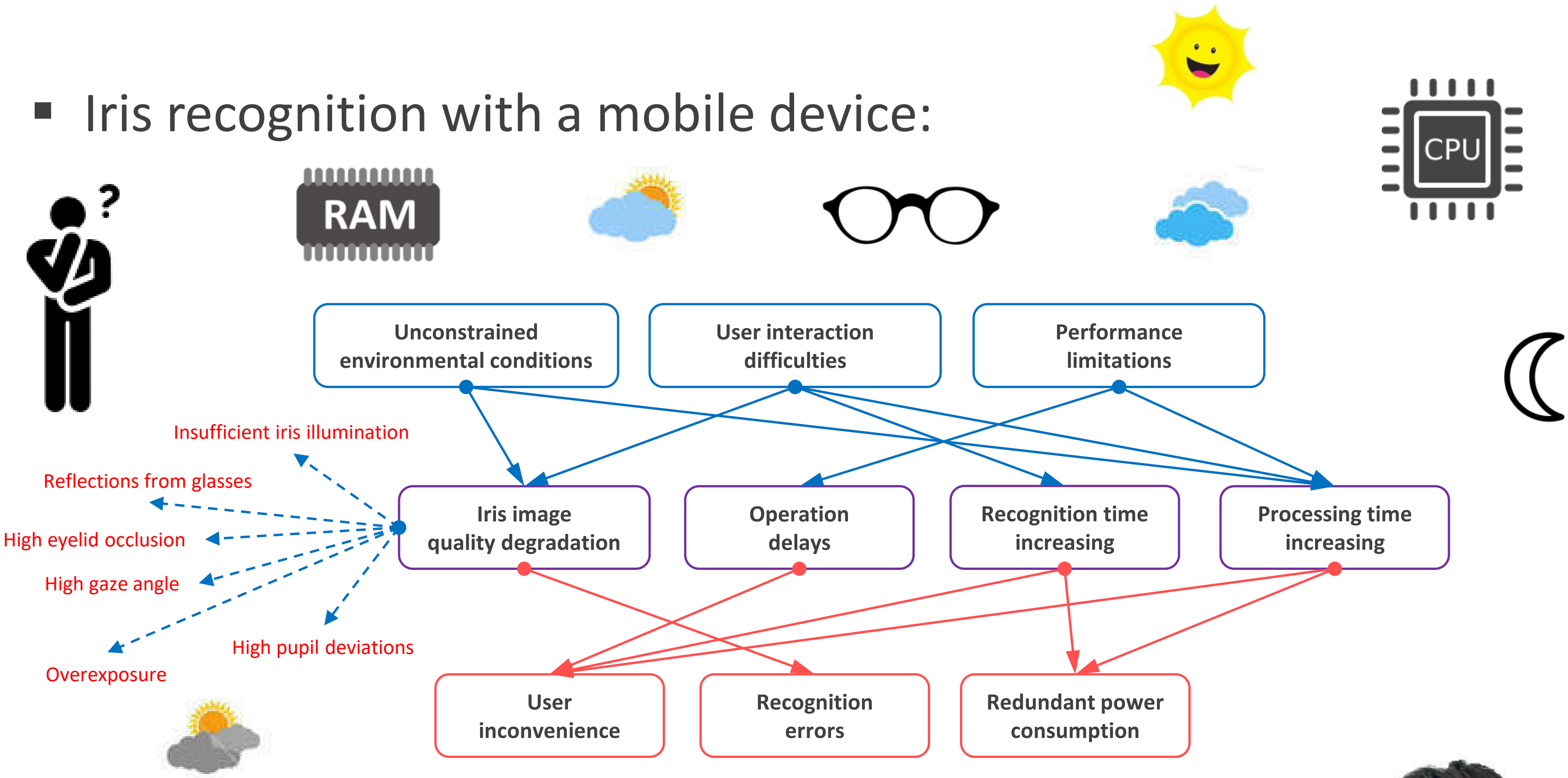
Gleb Odinokikh
FRC CSC RAS, Moscow

Outline

1. Introduction
   - Iris recognition with a mobile device
2. Problem statement
   - Conventional eyelid detection approach
   - Existing methods examples
   - Drawbacks of existing methods
3. Proposed solution
   - Idea description
4. Algorithm description
5. Experimental results
   - Eyelid detection accuracy measurement
   - Testing data information & results
   - Summary

Introduction

- Iris recognition with a mobile device:



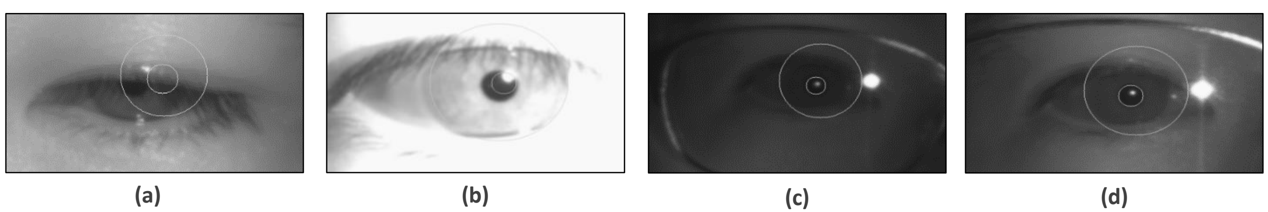
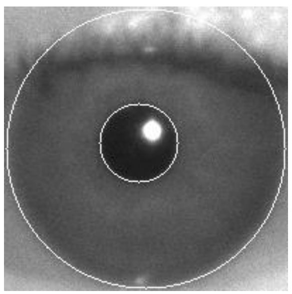


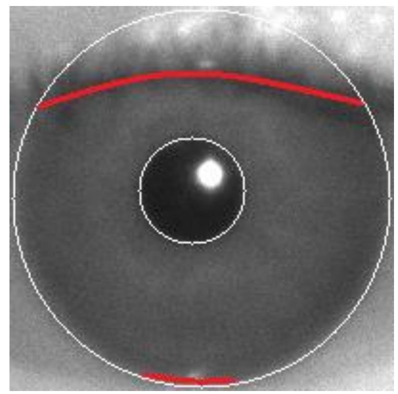
Fig. 1 Image quality degradation: (a) – gaze away, eyelid occlusion, (b) – overexposure, (c, d) – poor contrast, reflection from glass surface

Problem statement

- Conventional eyelid detection approach:
  - Eyelid noise removal
  - Iris image quality estimation



Iris-sclera boundary localization



Eyelid boundary localization

- Drawbacks of existing methods:
  - Not robust in case of unconstrained conditions
  - Most of them are computationally complex
  - Performed after complex operations of iris center definition and iris-sclera boundary localization

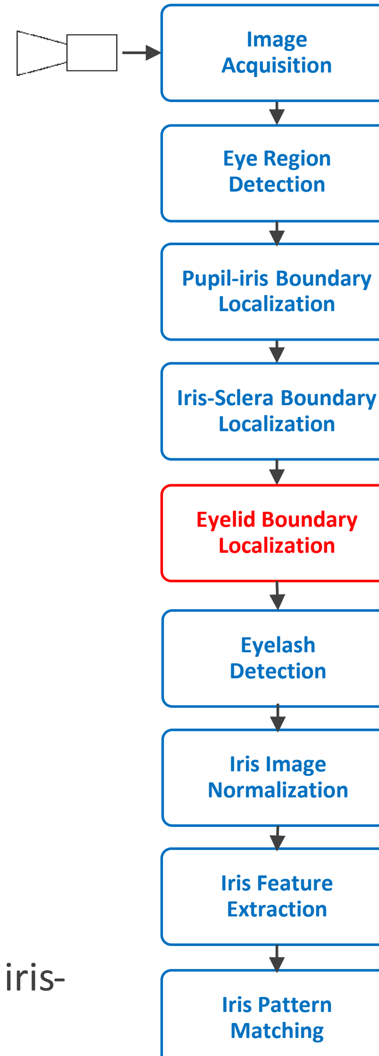


Fig. 2 Iris recognition algorithm flowchart

Problem statement

- Existing methods examples:

$$max_{a,k,h}\left|\sum_a \sum_k F * \frac{d}{dh} \sum_h (y - k^2) - 4a(x - h)\right|$$

Author	Preprocessing	Localization
Daugman	Gaussian blur	Parabolic IDO
Wildes	Sobel	Hough transform
Masek	Sobel	Line fitting (least squares)
Kang & Park	Sobel (modified)	Parabolic IDO
Xiangde et al.	1D peak shape filter	Parabolic IDO
Adam et al.	Anisotropic diffusion	Hough transform
Yang et al.	Asymmetric Canny	Parabola fitting (least squares)
Kim, Cha at al.	Histogram equalization	Local minima search
He et al.	1D rank filter	Pre-established model fitting

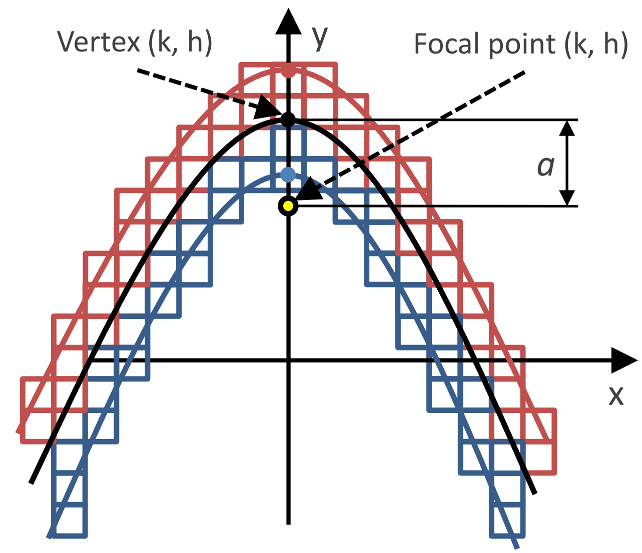


Fig. 3 Parabolic Integro-Differential Operator (Parabolic IDO)

Proposed solution

- Idea:
  - Detection of eyelid position earlier: for definition of E_u and E_l points (see pic. below) right after pupil-iris boundary localization stage
  - Use this information further for:
    - eye opening condition estimation
    - iris-sclera boundary localization algorithm parameters readjustment
    - full eyelid boundary localization/refinement
  - If eye isn't opened enough:
    - proceed to the next frame immediately
    - provide user with feedback like: "Open eye fully"

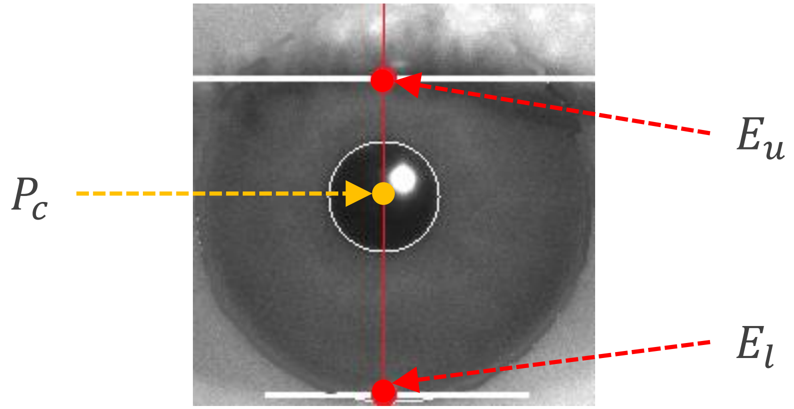


Fig. 4 Eyelid position points

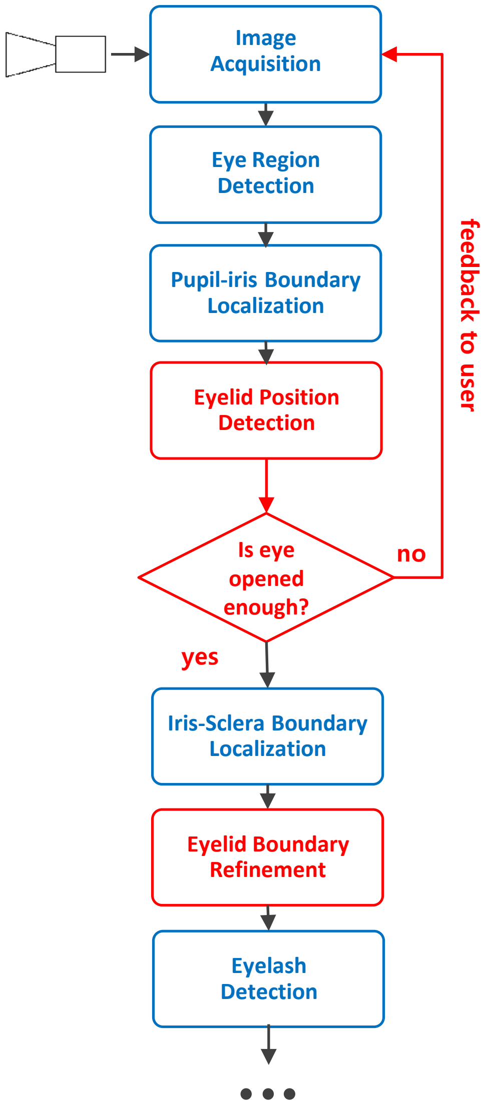


Fig. 5 Proposed flowchart modification

E_u − upper eyelid position point, E_l − lower eyelid position point, P_c − pupil center point

Algorithm Description

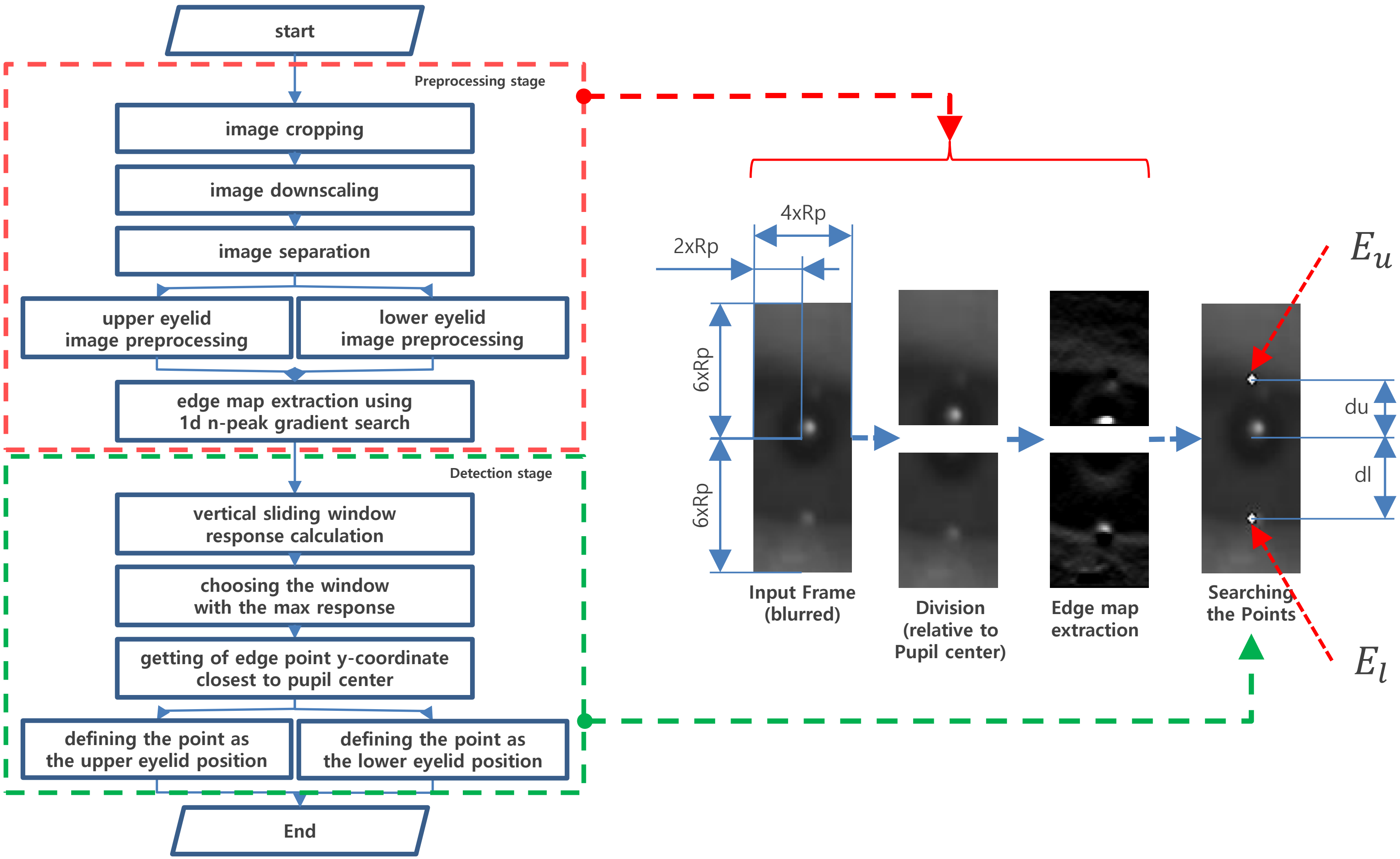


Fig. 6 proposed algorithm structure

R_p − *pupil radius*

Algorithm Description

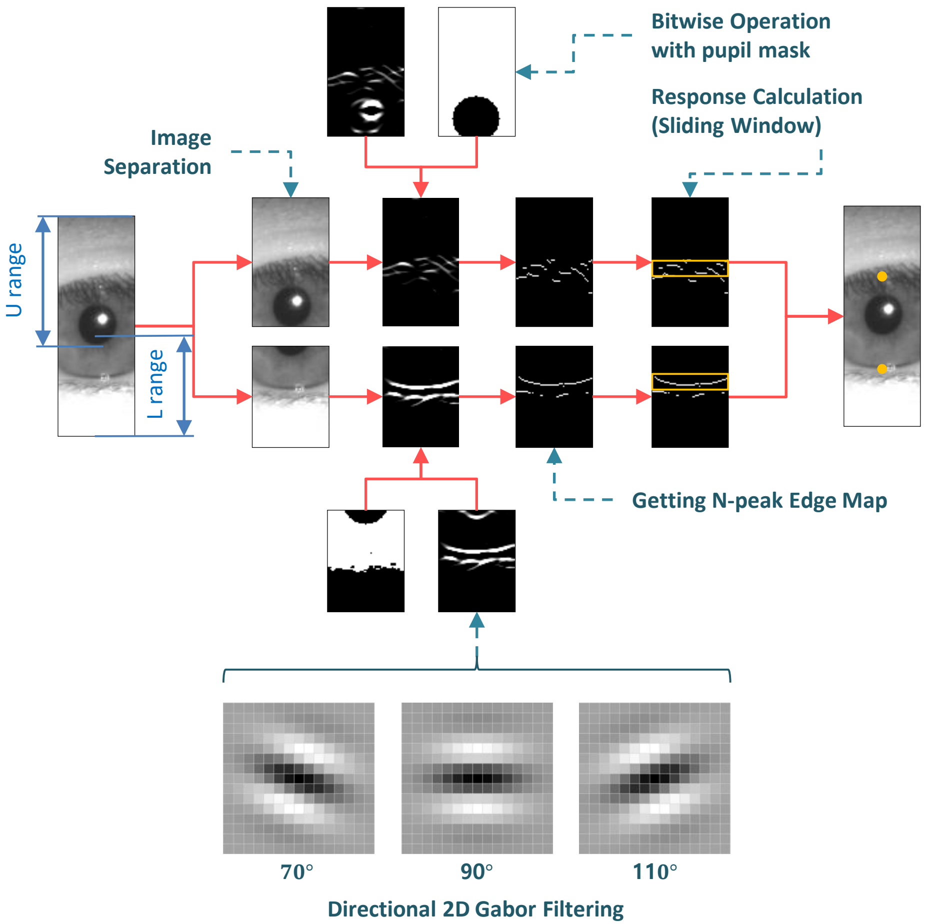


Fig. 7 Proposed algorithm steps

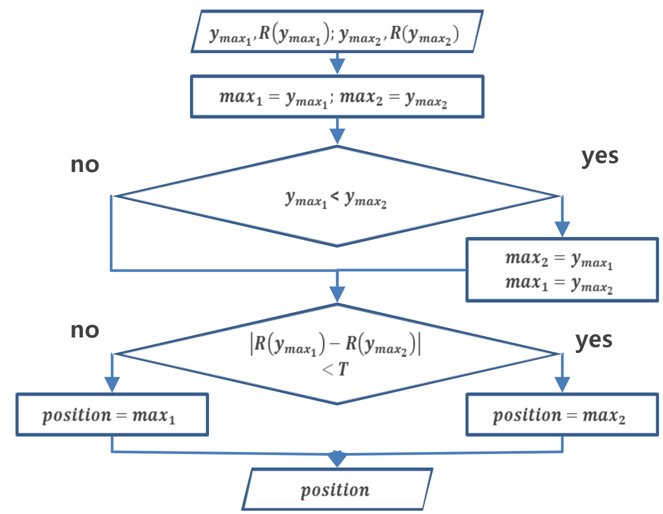


Fig. 8 Choosing between two peaks rule

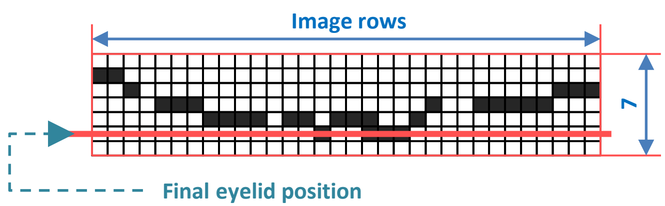


Fig. 9 Final eyelid position selection from edge map

Experimental Results

- Eyelid detection accuracy measurement:
  - evaluation method – admissible error rate: $\xi_{1..N_e}^{adm} = \{5\%, 10\%, 15\%\}$

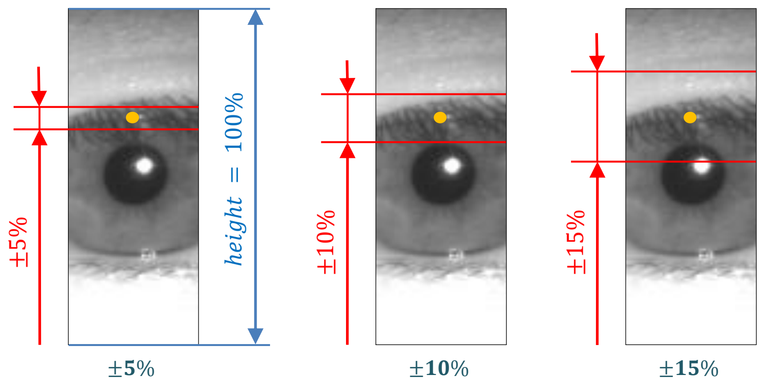


Fig. 10 Different admissible error examples

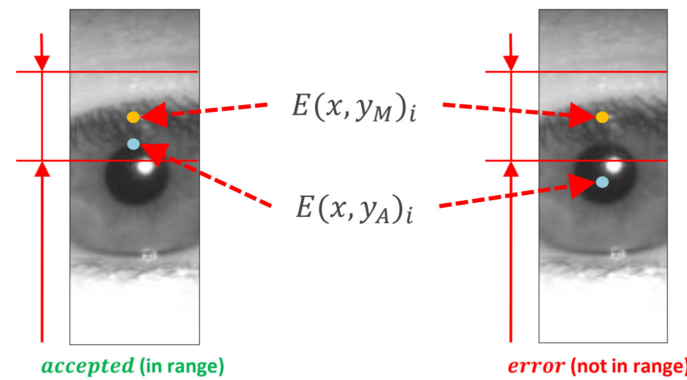


Fig. 11 Correct and incorrect eyelid position definition examples

- $\mathcal{E}_j$ define a part of the images in single dataset are ***not accepted*** (found eyelid position isn't in admissible range):

$$\mathcal{E}_j = \frac{1}{N}\left|\left\{\forall i: |E(x, y_A)_i - E(x, y_M)_i| > \xi_j^{adm} * height\right\}\right|$$

- Then $\mathcal{E}_j$ ***averaged*** for different datasets:

$$AVG_{\xi_j^{adm}} = \frac{100\%}{N_D}\sum_{i=1}^{N_D}(1.0 - \mathcal{E}_j^i)$$

where $E(x, y_A)_i$ - eyelid point detected by algorithm, $E(x, y_M)_i$ - eyelid point manually marked, N_D - number of datasets used

Experimental Results

- Testing data information & results:
  - 4 different datasets are used: MIR-Train, CASIA4-Thousand, CASIA3-Lamp and AOPTIX
  - $\geq$ 500 images of each dataset have been cropped and manually marked by expert and used for testing

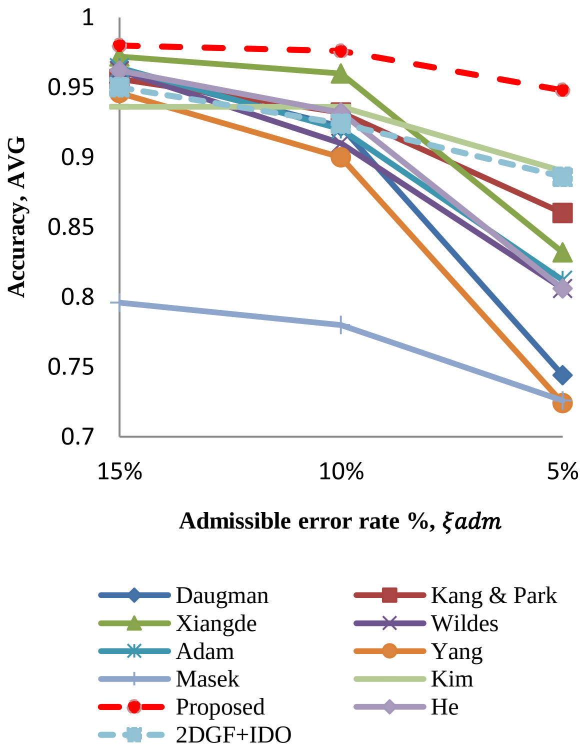


Fig. 12 Upper eyelid detection accuracy

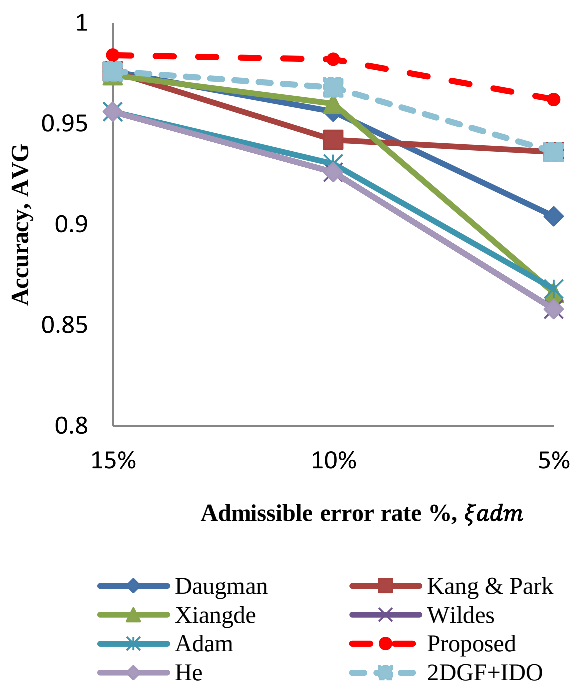


Fig. 13 Lower eyelid detection accuracy

Experimental Results

- Accuracy testing results for $\xi^{adm} = 5\%$

Table 1. Upper eyelid detection accuracy (%) for $\xi^{adm} = 5\%$

Dataset	MIR	CS4	CS3	APX	AVG
Daugman	76	70	83	84	74,4
Wildes	80	83	92	74	80,6
Masek	50	70	90	93	72,6
Kang & Park	86	89	90	88	86,0
Xiangde et al.	56	92	95	94	83,2
Adam et al.	80	83	91	78	81,2
Yang et al.	55	83	78	90	72,4
Kim, Cha et al.	89	89	99	98	89,0
He et al.	80	83	92	74	80,6
2DGF+IDO	93	90	95	91	**92,3**
Proposed	98	97	97	91	**94,8**

Table 2. Lower eyelid detection accuracy (%) for $\xi^{adm} = 5\%$

Dataset	MIR	CS4	CS3	APX	AVG
Daugman	88	86	95	94	**90,8**
Wildes	87	78	92	92	87,3
Masek	40	65	86	95	71,5
Kang & Park	96	88	95	94	**93,3**
Xiangde et al.	77	87	87	92	85,8
Adam et al.	87	79	93	95	88,5
Yang et al.	12	28	34	72	36,5
Kim, Cha et al.	30	50	22	32	33,5
He et al.	87	78	92	92	87,3
2DGF+IDO	97	86	92	96	**92,8**
Proposed	99	94	96	94	**95,8**

- Summary
  - Proposed method:
    - outperform all the existing methods by accuracy → **reliable**
    - robust on different datasets → **applicable for mobile applications, could be used for another purposes: gaze tracking, fatigue detection etc.**
    - allows to detect eyelid position on early stages → **saves processing time, allows to give user a feedback quickly**
    - processing time is about $\leq \mathbf{1}\boldsymbol{ms}$ on Snapdragon 800 (2,26GHz), single core → **fast & simple**

Q&A

Thank you.