

Low Cost

# 안경形 AEyeMouse

교수님

박정훈 교수님

전자공학과  
팀장

김민

교통시스템공학과  
팀원

송효정

# Index

목차

› 1. 프로젝트 소개

› 2. 루게릭 병

› 3. 기존 제품

› 4. 저가형 안경형 마우스

› 5. How ?

› 6. 결과 및 시연 영상

## 01. 프로젝트 소개 시선을 새롭게 연결하다, 안경形 AEyeMouse



교수님

박정훈 교수님

팀장님

전자공학과 김민

팀원

교통시스템공학과 송효정

### › 프로젝트 진행 과정

#### 1. 하드웨어 구현

안경프레임 +  
카메라 + 마이크로 컴퓨터



#### 2. 소프트웨어 개발

Starburst 알고리즘 적용



#### 3. 시스템 통합

시선추적점 최적화 알고리즘  
개발, GUI 연동

# 루게릭병, ALS Amyotrophic Lateral Sclerosis

》 근위축성 마비

》 강직 증세

》 구음 및 연하 장애

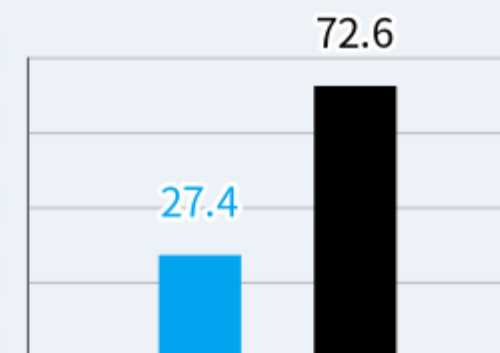
출처 : 2006년 한국루게릭협회 ,  
한양대병원 난치성세포치료센터의 예비 연구조사



## 》 루게릭 병 관련 연구 조사 자료

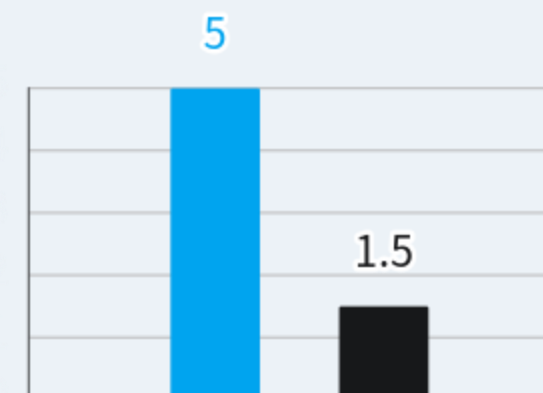
### 초기 증상

■ 연수형 초기증상  
■ 사지형 초기증상



### 환자 발생 경향성

■ 유병률  
■ 발병률




환자와 타인과의 "소통의 부재 "

## EYE Tracker 기대 효과

- ✓ **환자의 독립성 증진** 스스로 의사소통 및 컴퓨터 사용 가능 , 신체적 제약 극복
- ✓ **간병인의 부담 감소** 의사소통의 원활화를 통해 환자의 고립감 감소 , 보호자 / 간병인의 부담 완화
- ✓ **사회적 연결 유지** 환자와 가족, 의료진 간 소통 강화로 정서적 웰빙 증대

### 03. 기존 제품

## eyetracker 시중 판매품



토비 아이트래커 Tobii Eye Tracker5 아이트래킹

5.0 ★★★★★ 2개 상품평

**513,700원** (1개당 513,700원)




무료배송 | 반품비 40,000원 | 더 알아보기  
12/17 도착 예정

판매자: 마우이글로벌  
판매자 평가: ⭐ -- % (14건 이하의 평가) ①

배송사: 우체국

수량: 1개

적립 최대 25,685원 쿠팡캐시 적립 · 쿠팡이머니 결제시 [혜택보기 >](#)

PC에서도 간편한 결제  쿠팡이머니  카드  계좌이체

1 [장바구니 담기](#) [바로구매 >](#)

• 쿠팡상품번호: 6928697646 - 16758186349

## Tobii 사의 eyetracker

- 모니터에 부착하는 형태
- 양안에 적외선을 조사하고 이를 감지하여 포인터를 제어하는 방식을 이용
- 강점: 모니터에 부착가능하여 편리
- 단점: 고가의 가격

## Focus 안경형 마우스만의 차별점



**“ 저렴한 비용으로 접근성 확대 ”**

#### 04. 제품소개

## 저가형 안경형 마우스



소요 부품	단가
라즈베리파이 v2 소형카메라	₩ 27,500
3D 프린터 안경	₩ 5000
최종 단가	약 5만원



05.

# HOW ?

## 스타버스트 알고리즘(1)

- 1 **Input:** Eye image with corneal reflection removed, Best guess of pupil center
- 2 **Output:** Set of feature points
- 3 **Procedure:**
- 4 Iterate
- 5 *Stage 1:*
- 6 Follow rays extending from the starting point
- 7 Calculate intensity derivative at each point
- 8 If derivative > threshold then
- 9 Place feature point
- 10 Halt marching along ray
- 11 *Stage 2:*
- 12 For each feature point detected in Stage 1
- 13 March along rays returning towards the start point
- 14 Calculate intensity derivative at each point
- 15 If derivative > threshold then
- 16 Place feature point
- 17 Halt marching along ray
- 18 Starting point = geometric center of feature points
- 19 Until starting point converges

Figure 2: Feature-point detection method

## 스타버스트 알고리즘 (2)

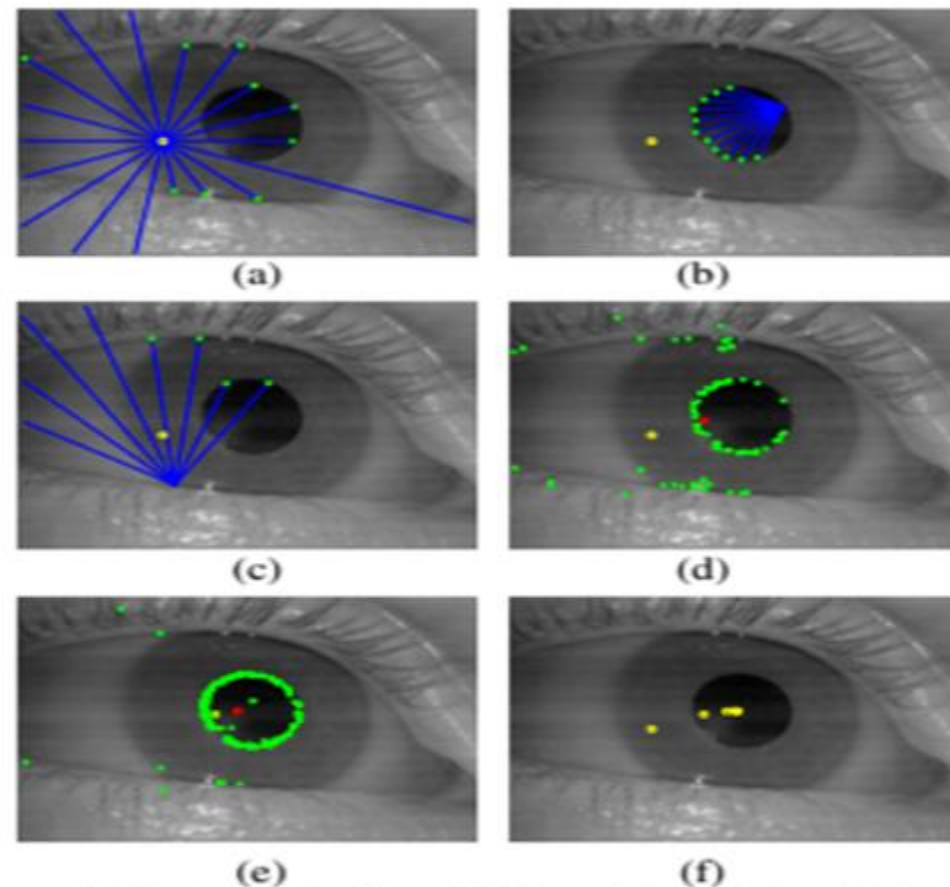


Figure 4: Feature detection. (a) The original start point (yellow circle) shoots rays (blue) to generate candidate pupil points (green crosses). (b&c) The candidate pupil points shoot rays back towards the start point to detect more candidate pupil points. (d) All the candidate pupil points are shown. The average of these locations is shown as a red circle. This location seeds the next iteration. (e) The results of the second iteration. (f) The starting locations from all iterations show a rapid convergence.

## 알고리즘 개발

```
# 현재 마우스 x,y 좌표 반환하기
def get_mouse_position():
    return pyautogui.position()

# 자동 클릭 함수_마우스가 gui 버튼창 & gui 창이 없는 기존 화면창에 있는지 두 영역으로 분리해서 위치 확인
def click_if_stationary():
    stationary_time = 0
    check_interval = 0.1
    hold_time = 1.7 # 1.5에서 1.8이 적당할 듯

    last_position = get_mouse_position() # 기존 위치를 인식

    while running:
        current_position = get_mouse_position() # 현재 마우스 위치 반환

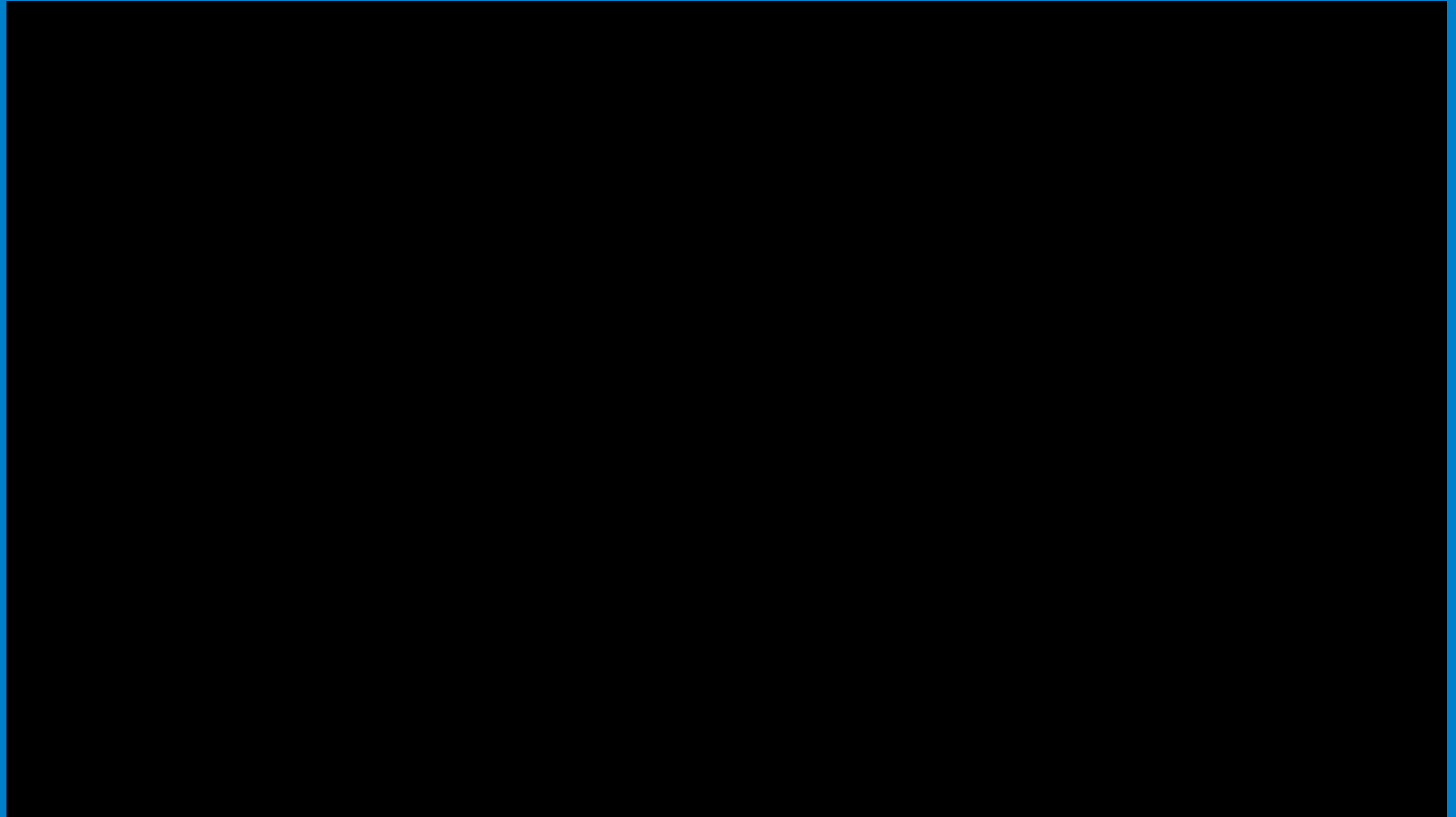
        # 버튼의 위치 확인
        button_positions = [(btn.wininfo_rootx(), btn.wininfo_rooty()) for btn in buttons]
        button_size = [(btn.wininfo_width(), btn.wininfo_height()) for btn in buttons]

        # 버튼 영역 안에 마우스가 있는지 확인
        for (posx, posy), (width, height) in zip(button_positions, button_size):
            if (current_position[0] in range(posx, posx + width) and
                current_position[1] in range(posy, posy + height)):
                stationary_time += check_interval
                if stationary_time >= hold_time:
                    move_command_mouse()
                    print(f"Clicked at {current_position} (Button Area)")
                    stationary_time = 0
                    break
        else: # break 걸리면 for 문이 if문으로 변함되면서 else 실행됨
            if current_position == last_position:
                stationary_time += check_interval
                if stationary_time >= hold_time:
                    move_command_mouse()
                    print(f"Clicked at {current_position} (Mouse Area)")
                    stationary_time = 0
            else:
                stationary_time = 0
```

- # 시선추적점 최적화 알고리즘

06.

## 결과 및 시연 영상



06.

## 향후 방향성



### ✓ 의료기관과의 협업

정밀 환자 모니터링, 재활 분야로의 확장

### ✓ 협동 로봇과의 연계

물리적 제약을 넘어설 기회 : 노약자, 장애인도 사용할 수 있는 지원시스템

### ✓ 다양한 IoT 기기와의 통합

저가형 안경형 마우스

Thank you