

# Virtuális Egér

---

Horváth Zsolt, Schnádenberger Gábor, Varjas Viktor

2011. március 20.

## Feladat kiírás:

Egy olyan rendszer megvalósítása, melyben kamera értelmezi a kéz és az ujjak mozgását. Ezek a mozgások megfeleltethetők egy igazi egér eseményeinek, mintha egy touch pad-ot kezelnénk. Kamera elhelyezhető a monitor tetején, vagy egyéb helyen.

## Megvalósítás:

Olcsó webkamera segítségével, mely felülről figyeli a kéz gesztusait.

*Kéz alapértelmezett állapota:* mutató ujj előre felé, kinyújtott helyzetben. A kéz mozgása megfeleltethető az egérmutató mozgásának.

*Bal klikk esemény:* Hüvelyk ujj kinyitása egy bizonyos küszöb szög feletti állásba, a mutató ujjhoz képest.

*Bal gomb dupla kattintás esemény:* Hüvelyk ujj kinyitása egy időintervallumon belül kétszer, a küszöb szög fölé.

*Bal gomb nyomva tartás esemény:* Hüvelyk ujj kinyitása a küszöb szög fölé és a szög fölött tartása egy meghatározott időn túl.

*Jobb klikk esemény:* Az összes ujj kinyújtása.

## Megkötések:

Fekete egérpad alkalmazása a zavaró árnyékok kiküszöbölése végett. Az téglalap alakú egérpad egy átlója mentén, két átellenes csúcspontjában egy-egy kör alakzat.

Megfelelő megvilágítás biztosítása, természetes fény, vagy mesterséges fény segítségével.

Kamera elhelyezése egy meghatározott távolságban a kéztől.

## Algoritmus terv:

### 1. A kamera paramétereinek beállítása.

Szélesség: 640px

Magasság: 480px

FPS: 30 kép/s

### 2. Színes kép --> szürkeárnyaltos kép konverzió

### 3. Gauss szűrés alkalmazása

A zajos kamera kép simítása,  $\sigma = 2$ , 11x11-es Gauss maszkkal történő konvolúció segítségével.

A kép részletgazdagsága nem szükséges a feladat megoldásához, a zaj eltávolítása fontos a szegmentálás korrekt kivitelezéséhez.

A 2D Gauss függvény:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

#### 4. Küszöbölés

A sötét részek eldobása a világosak kihangsúlyozása. A küszöb érték feletti intenzitásértékek maximálisra (255), a küszöb alattiak pedig feketére (0) állítása.

#### 5. Kéz detektálása

##### a. A kéz pixelek észlelése terület kizárással

Az egérpadon elhelyezett két kör észlelésével egyértelműen meghatározható egy téglalap. A téglalap területébe eső magas intenzitásértékek már nagyrészt a kéz területéhez tartoznak, köszönhetően a fekete háttérnek. Az azon kívül eső pixelek zaj vagy egyéb tárgyak, melyek pixeleit eldobhatjuk (0 intenzitásértékre állítás).

A két kör észlelése Hough transzformációval megoldható. Az él-képen található kör körívét alkotó pontok, egy-egy kört alkotnak a Hough-térben. Ezen körök körívén található a potenciális középpont is, ezért sok ilyen kör közös metszéspontja meghatározza az él-képen található kör középpontját.

A kör egyenlete:

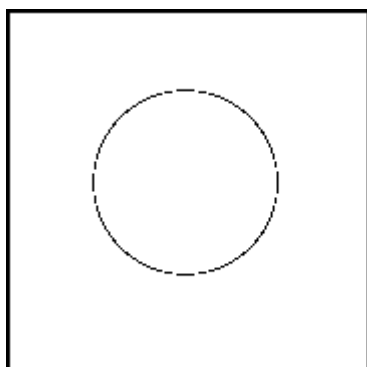
$$r^2 = ax^2 + by^2 + c$$

Kör reprezentációja:

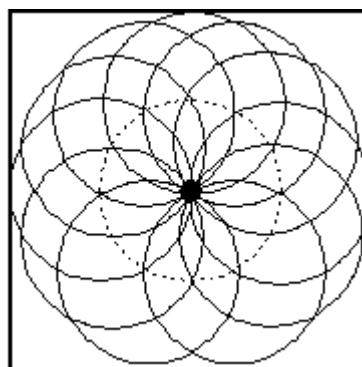
$$x = x_0 + r \cos \Theta$$

$$y = y_0 - r \sin \Theta$$

$(x, y)$  egy pont koordinátái,  $(x_0, y_0)$  keresett kör középpontja,  $\Theta$  szög a Hough-térből,  $r$  a keresett kör sugara



Él-kép



Maximumhely a Hough térben

Ha adott sugarú kört keresünk, akkor csak  $(a, b)$  paraméterekkel kell számolnunk. A transzformáció során akkumulátor tömböt használunk. Ebben a

tömbben adott x, y pozíciójában, ha lokális maximumot találunk, akkor az egy kör középpontját jelenti.

A keresés 17 és 20 px közötti sugarú köröket érint, minimum 400 px távolságra egymástól.

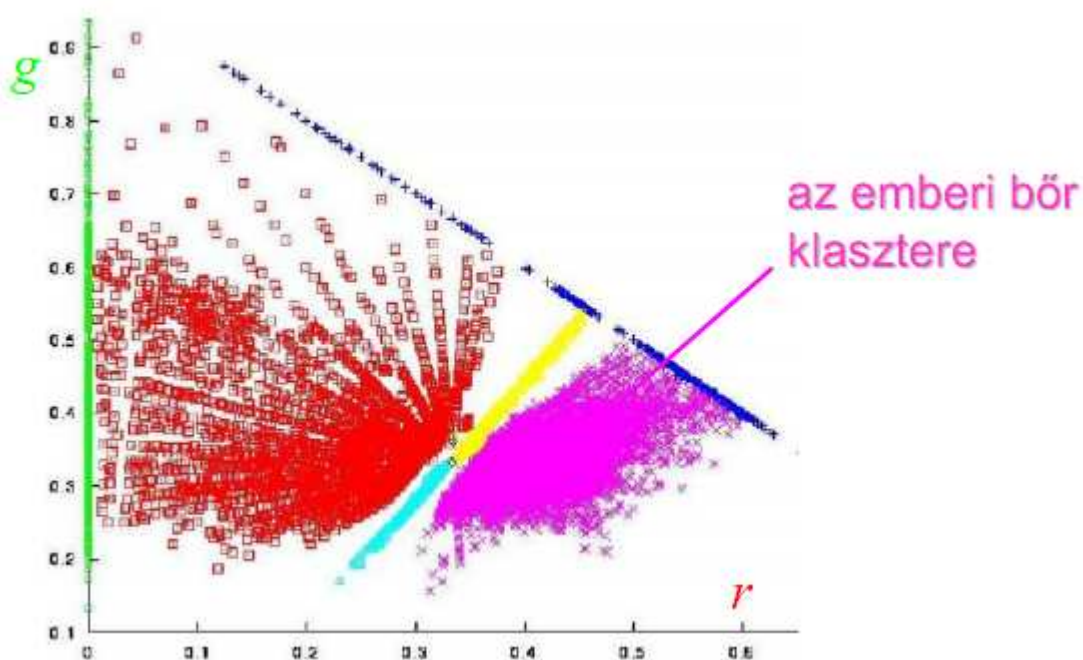
Ha maximum két kört találunk, akkor az általuk határolt téglalapon kívül a pixel intenzitások felülírása 0 értékkel. Ezután a téglalap belső területén már csak a kéz pixelértékei veszik fel a 255-ös értéket.

#### b. A kéz pixelék észlelése bőrdetektálással

A kéz pixeleinek megtalálása a bőr színének elkülönítésével a külvilágtól. Ekkor nincs feltétlen szükség sem küszöbölésre, sem fekete háttérre, a szegmentálás elvégezhető a kéz pixeleinek ismeretében.

A detektáláshoz többféle színtér konverzió is a rendelkezésre áll. Egy egyszerű megoldás lehet az RGB  $\rightarrow$  2D normalizált színtér konverzió.[5]

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$$



#### Kurzor mozgása:

##### 1. Globális minimum y koordinátájú pont megkeresése

A képen sorfolytonosan végigmegyünk, képen fentről lefelé haladva, majd a legelső 255-ös intenzitás értékkel rendelkező pont koordinátáit elmentjük. Ez a pont a mutató ujj vége, melyet minden képkockán újra meg lehet keresni.

## 2. Kurzor koordinátáinak beállítása

Az előző pontban megtalált pont segítségével új pozícióba irányítjuk a kurzort. Mivel a kéz mozgási területe nem egyezik meg a monitor felbontásával, ezért az értékeket skálázni kell.

$X_{\text{mouse}}, Y_{\text{mouse}}$ :	egérmutató koordinátái
$x_{\min}, y_{\min}$ :	mutatóujj legelső pontjának koordinátái
HorizontalRes, VerticalRes:	monitor felbontása
$K1x, K2x$ :	legfelső és legalsó kör x koordinátái.
$K1y, K2y$ :	legfelső és legalsó kör y koordinátái

$$X_{\text{mouse}} = \frac{x_{\min} - K1x}{K2x - K1x} * \text{Horizontal Res}$$

$$Y_{\text{mouse}} = \frac{y_{\min} - K1y}{K2y - K1y} * \text{Vertical Res}$$

## Ujjmozgás detektálás:

### 1. Kontúrok és konvex burok keresése

Küszöbölt szürkeárnyaltos képen a kontúrvonalak megkeresése. Külső nagy határoló vonalak és kis belső kontúrokra osztás. Ha a nagy kontúr által határolt terület meghalad egy bizonyos terület méretet, akkor megtaláltuk a kezet.

Meghatározzuk a konvex burkot Sklansky algoritmus segítségével. A burok pontja között lesznek az egyes ujjak is. Ezek közül csak a hüvelykujjra és a mutató ujj pontjaira van szükség, a többi eldobhatjuk.

### 2. Klikkelések

Hüvelykujj és a mutató ujj pontjainak távolsága egy meghatározott érték fölött bal klikkelésnek minősül.

Dupla kattintás egy idő intervallumon belül történő küszöböt átlépő távolságváltozás.

Nyomva tartás az eltelt időn túli, küszöb fölötti két ujj távolságának tartása.

Jobb klikk a kéz konvex burkának terület növekedése.

**Hivatkozások**

1. G. Bradski, A. Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library (The OpenCV book)
2. S. Malik, J. Laszlo. Visual Touchpad: A Two-Handed Gestural Input Device. *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces (2004)*, pp. 289-296.
3. Z. Zhang, Y. Wu, Y. Shang. Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. *Workshop on Perceptive User Interfaces (PUI 2001)*, Nov. 15-16, 2001. Orlando, Florida.
4. V. I. Pavlovic, R. Sharma, T. S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (1997)*, pp. 677-695
5. Paláhyi Kálmán: Képfeldolgozás haladóknak 2. előadás