Képfeldolgozás a gyakorlatban

Szeghalmy Szilvia

Tartalom

- Követelmények
- Kép
- Mire jó a képfeldolgozás?
- OpenCV kezdőlépések

Követelmények

- A ZH legalább 60%-os teljesítése:
  - teszt (ha van, max. 10%)
    - kódértelmezés az alapokhoz kötődően (pl. kép típusa, műveletek)
    - képfeldolgozási eszközök hatásának felismerése
    - problémához illeszkedő képfeldolgozási módszer kiválasztása
    - stb.
  - rövid programok készítése
    - órán látott eszközökkel, órán nem látott képek feldolgozása
- Projektmunka legalább 60%-os teljesítése.
- A fontos dátumok (ZH és projektmunkához kapcsolódó határidők az e-learningben / a syllabuszban lesznek elérhetők:

  www.ik.unideb.hu/syllabi

Projektmunka – program fejlesztés

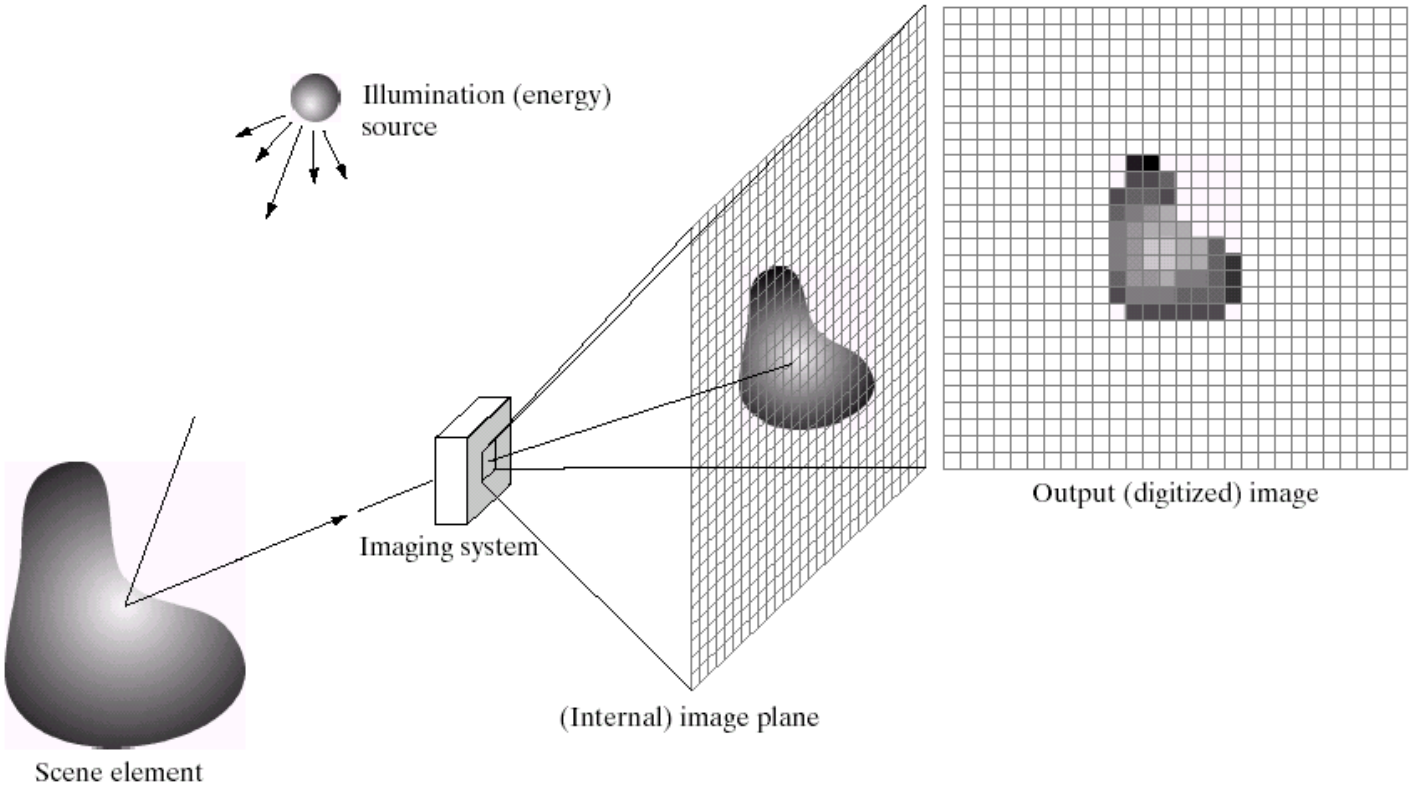
- 3 fős csapatok
  (kivéve szakdolgozat/TDK munka)
- Témaválasztás és a feladatok TISZTA felosztása a csapattagok között (felelősségi körök)
- Képek, felvételek készítése/gyűjtése
- Algoritmus tervezése
- Program fejlesztés
- Tesztelés
- Prezentáció készítés
- A munka bemutatása

Projektmunka – tutorial

- 2 fős csapatok
- Témaválasztás és a feladatok **TISZTA** felosztása a csapattagok között
  (ki miért felelős a projektben)
- Elméleti megoldások áttekintése (google tudós a barátjuk)
- Legalább egy szabadon elérhető megvalósítás (github a barátjuk) tesztelése:
  - a szerző által megadott képekre (ha vannak)
  - az elvártnak megfelelő, de nem a szerző által megadott képekre
- A prezentáció során legyen hangsúlyos a választott megoldás lépéseinek ismertetése (algoritmus) és a kódrészlet bemutatása.
- Ismertessék a tesztelés során tapasztaltakat (vö. tesztelés)

Digitális kép

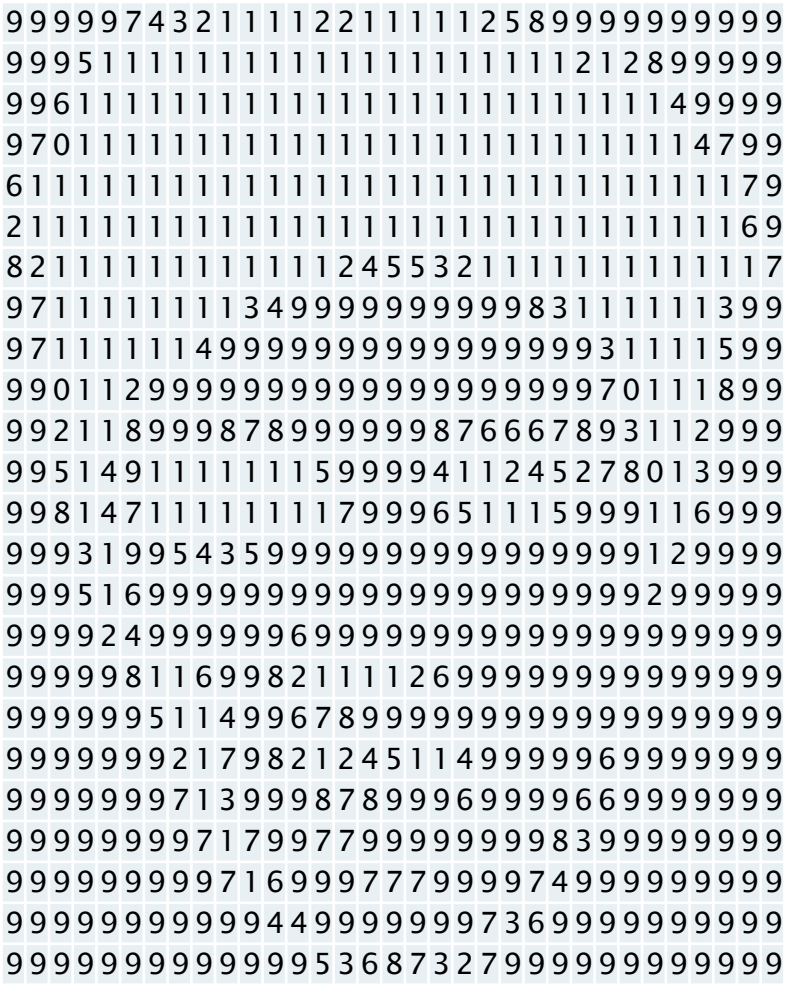
Intenzitásértékeket tároló mátrix

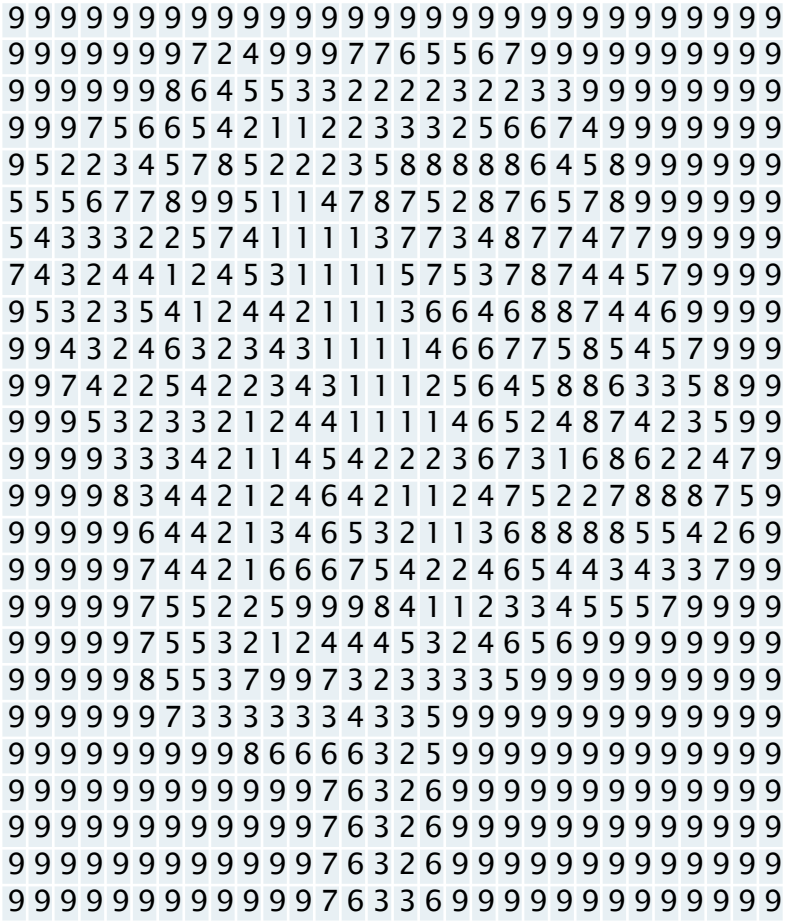


a
b c d e

FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Csak számok...


9999974321111221111125899999999999
9995111111111111111111111111212899999
9961111111111111111111111111111149999
9701111111111111111111111111111114799
6111111111111111111111111111111111179
2111111111111111111111111111111111169
8211111111111112455321111111111117
9711111111134999999999983111111399
9711111149999999999999999931111599
9901129999999999999999999701118999
9921189998789999998766678931129999
9951491111111159999411245278013999
9981471111111117996511159991169999
9993199543599999999999999991299999
9995169999999999999999999999299999
9999249999996999999999999999999999
9999981169982111126999999999999999
9999995114996789999999999999999999
9999999217982124511499999699999999
9999999713998789996999966999999999
9999999971799779999999983999999999
9999999997169997779999974999999999
9999999999445999999736999999999999
9999999999995368732799999999999999


9999999999999999999999999999999999
9999999724999776556799999999999999
9999999864553322223223399999999999
9997566542112233325667499999999999
9522345785222358888864589999999999
5556778995114787528765789999999999
5433225741111377348774779999999999
7432441245311115753787445799999999
9532354124421113664688744699999999
9943246323431111466775854579999999
9974225423431125645886335899999999
9995323212441111465248742359999999
9999333421145422236731686224799999
9999834421246421124752278887599999
9999964421346532113688855426999999
9999974421666754224654434337999999
9999975522599984112334557999999999
9999975532124453246569999999999999
9999985537997323335999999999999999
9999997333333433599999999999999999
9999999998666632599999999999999999
9999999999999763263999999999999999
9999999999999763263999999999999999
9999999999999763263999999999999999
9999999999999763363999999999999999

A gép nem lát, ezt mi látjuk

Elvis Presley

és a mikrofonja

Képfeldolgozás

- Image processing: (alacsonyszintű képfeldolgozás)
  - pl.: képjavítás, zajszűrés, jellemzőkinyerés (képfüggetlen)
- Image analysis:
  - pl.: szegmentálás, képregisztrálás,
    jellemzőkinyerés (értelemmel bíró jellemzők)
- Computer vision (magasszintű képfeldolgozás):
  - pl.: objektum detektálás, követés, felismerés,
    értelmezés (mi történik a képen?)

Mire jó a képfeldolgozás?

- Képjavítás

hisztogram kiegyenlítés

https://www.mathworks.com/content/mathworks/www/en/discovery/image-enhancement/jcr:content/mainParsys/image_1.adapt.full.high.jpg/1527655847357.jpg

elmosódás korrigálása

https://www.mathworks.com/content/mathworks/www/en/discovery/image-enhancement/jcr:content/mainParsys/image_2.adapt.full.high.jpg/1527655847370.jpg

Mire jó a képfeldolgozás?

- Szegmentálás:
  - a kép valamilyen szempontból homogén részekre bontása

Nathan Chen[1]

Egy szegmentálási eljárás eredménye

[1]Harry How/Getty Images, https://cdn.vox-cdn.com/thumbor/kiCF6pCcPPvUYKTOBkGTRYLKfKk=/0x0:5199x3552/920x613/filters:focal(2365x559:3195x1389):format(webp)/cdn.vox-cdn.com/uploads/chorus_image/image/58721383/918842020.jpg.0.jpg

Mire jó a képfeldolgozás?

Minőségellenőrzés

https://www.opto-engineering.com/albert

Címkevizsgálat

https://www.mt.com/ca/en/home/applications/Product-Inspection_2/label-quality-inspection.html

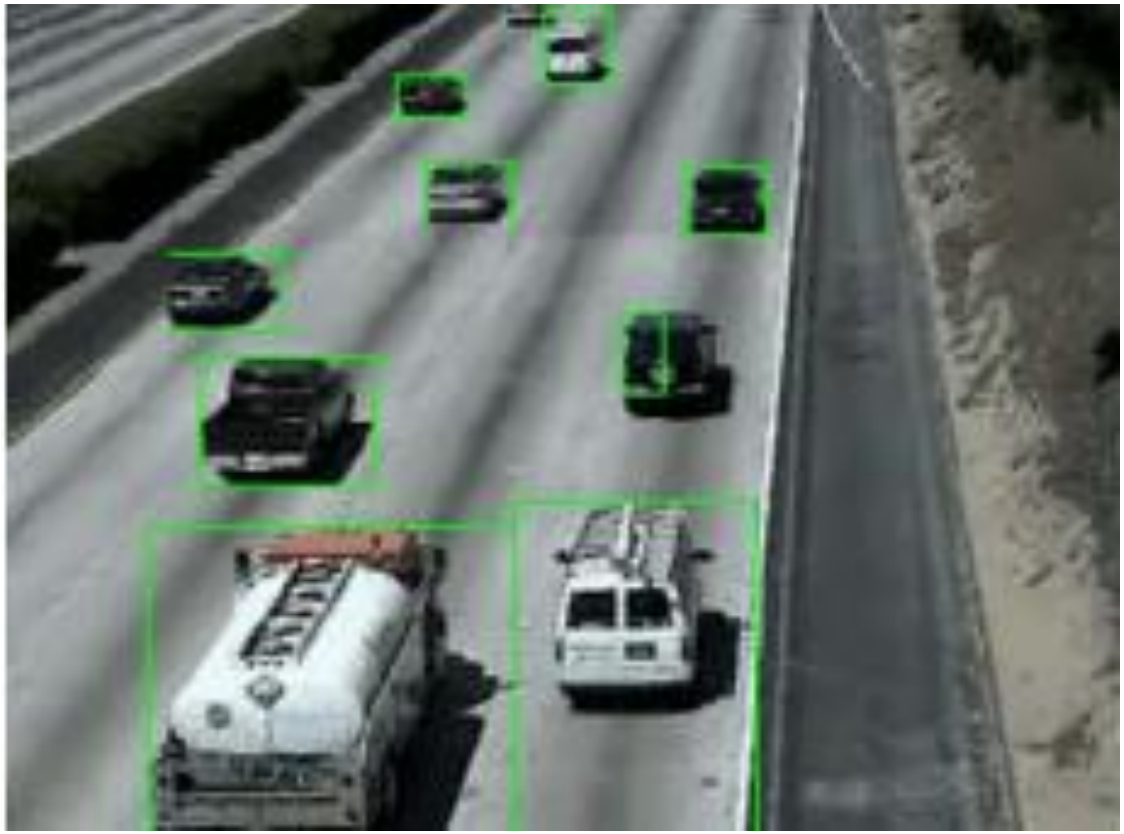
Mire jó a képfeldolgozás?

- Rendszámtábla felismerés

https://upload.wikimedia.org/wikipedia/commons/9/9c/California_license_plate_ANPR.png

Mire jó a képfeldolgozás?

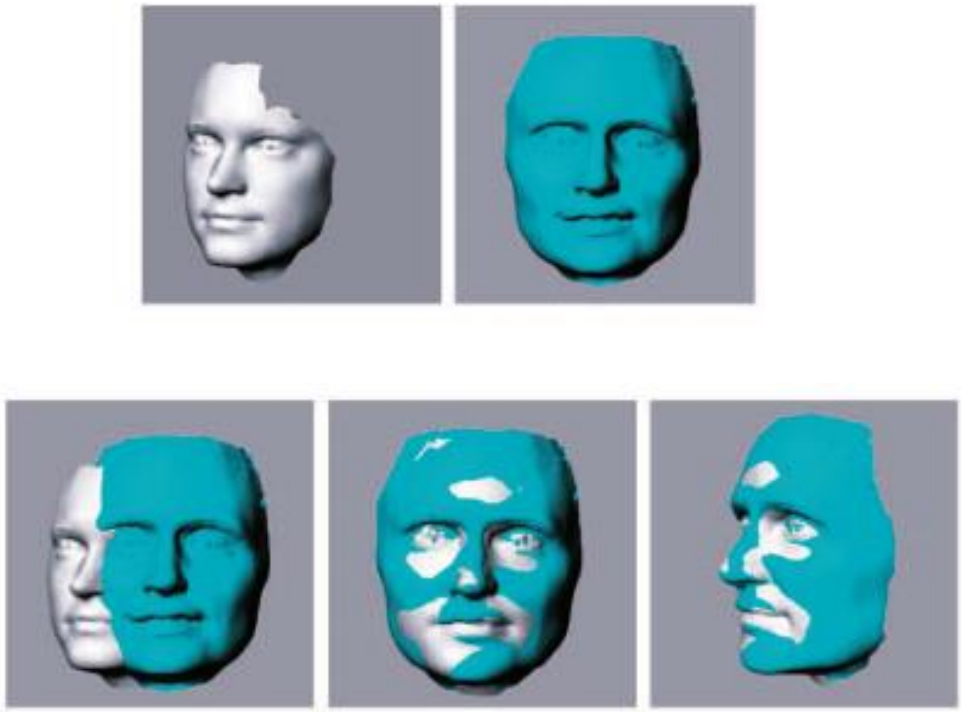
Forgalomszámlálás



Al Hussain Akoum, Automatic Traffic Using Image Processing, Journal of Software Engineering and Applications, 2017, 10, 765-776

Mire jó a képfeldolgozás?

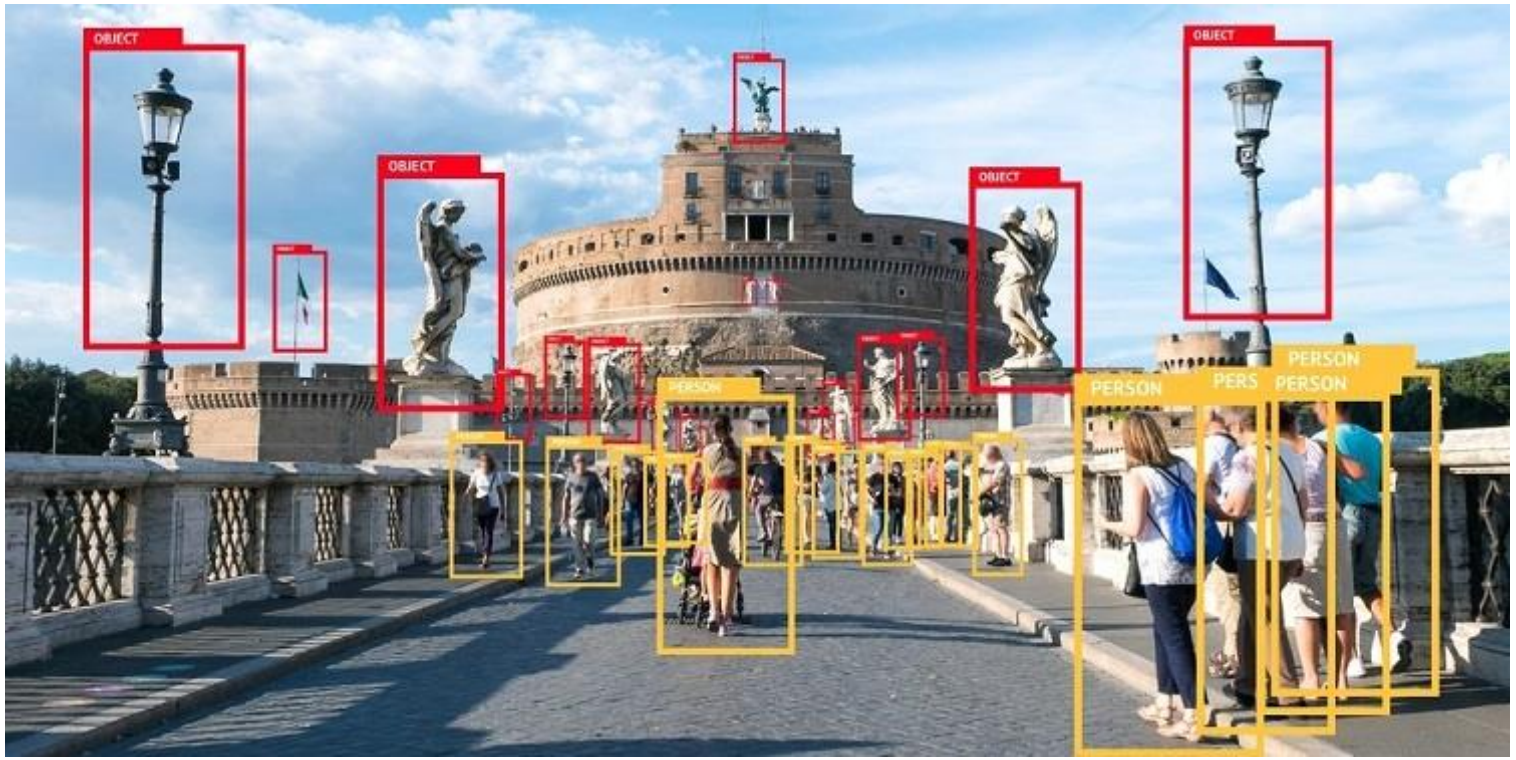
Objektum regisztráció



Huang, Xiaolei & Paragios, Nikos & Metaxas, Dimitris. (2006). Shape registration in implicit spaces using information theory and free form deformations. IEEE transactions on pattern analysis and machine intelligence. 28. 1303-18.

Mire jó a képfeldolgozás?

Objektumdetektálás és osztályozás



https://software.intel.com/sites/default/files/managed/a7/55/object-detection-recognition-and-tracking-fig00.jpg

Mire jó a képfeldolgozás?

Arcfelismerés

http://www.zerohedge.com/sites/default/files/images/user3303/imageroot/2017/12/05/20171206_face1_0.jpg

Mire jó a képfeldolgozás?

- Ember-gép kapcsolat
  - detektálás
  - korbecslés
  - arckifejezések felismerése ("érzelemfelismerés")

http://pragma.international/sites/default/files/styles/blog_header/public/articles/18/02/facial.jpg?itok=lu2YfwRN

OpenCV

- Képfeldolgozási függvénykönyvtár
- Hivatalos honlap:

  http://opencv.org/

- Fontos modulok:
  - Core (pl.: adatszerkezetek) — core.hpp
  - High-level GUI — highgui.hpp
  - Image processing — imgproc.hpp
  - Machine learning — ml.hpp

Órai környezet: Visual Studio + OpenCV

- Töltse le az OpenCV-t (Releases->Win package)

  `http://opencv.org/`
- Bontsa ki az alábbi könyvtárba:

  `c:\Programok`
- Az OpenCV telepítési könyvtár innentől kezdve:

  `c:\Programok\opencv\`

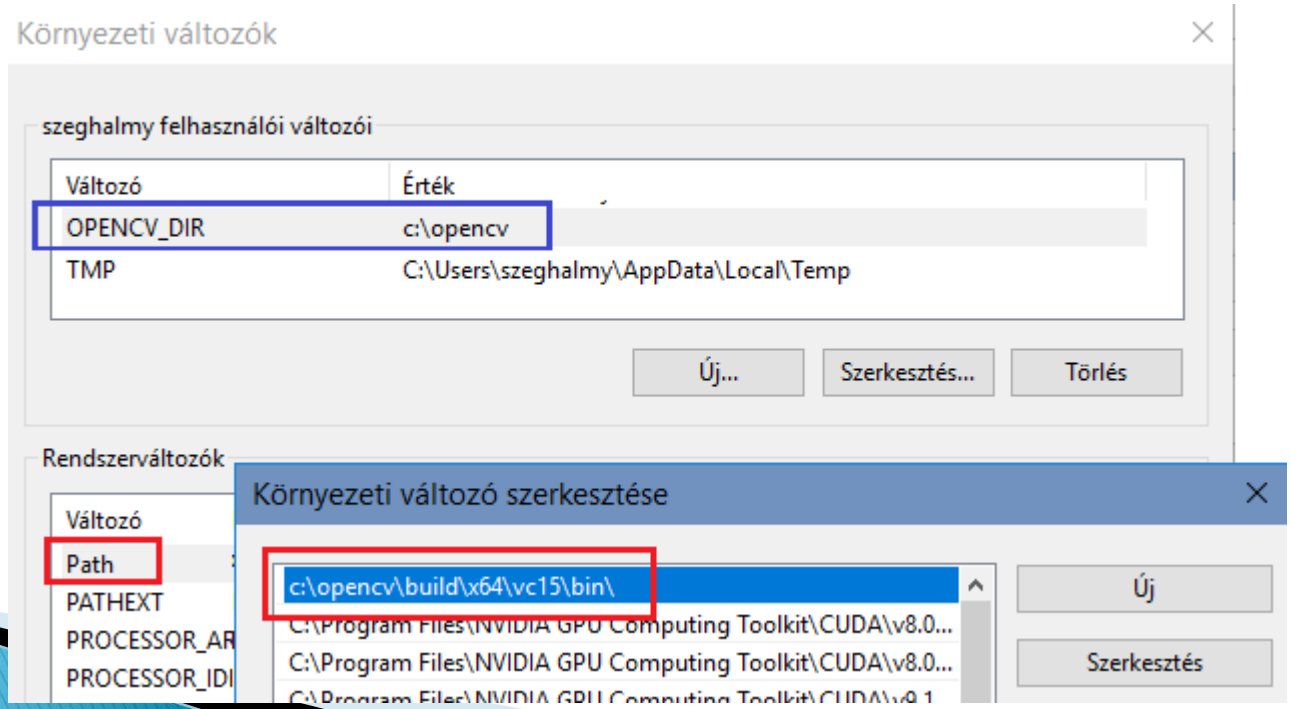
- Visual Studio aktiválás: (várhatóan 30 nap múlva kéri)

  user: deikvisualstudio@outlook.com

  pass: DEIKmsvs

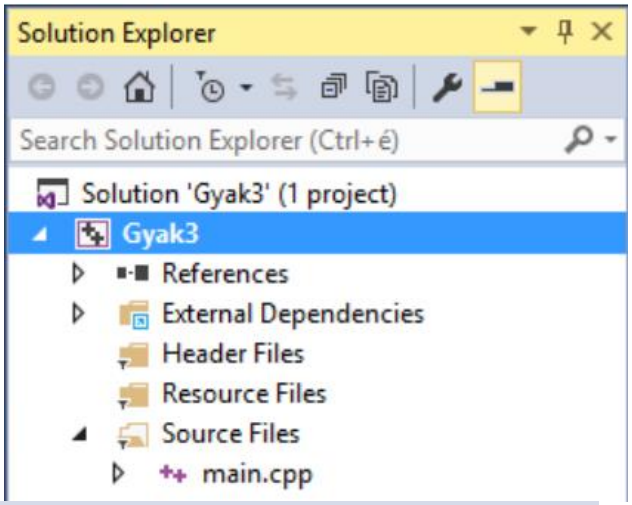
Saját gépen érdemes a PATH-ot módosítani

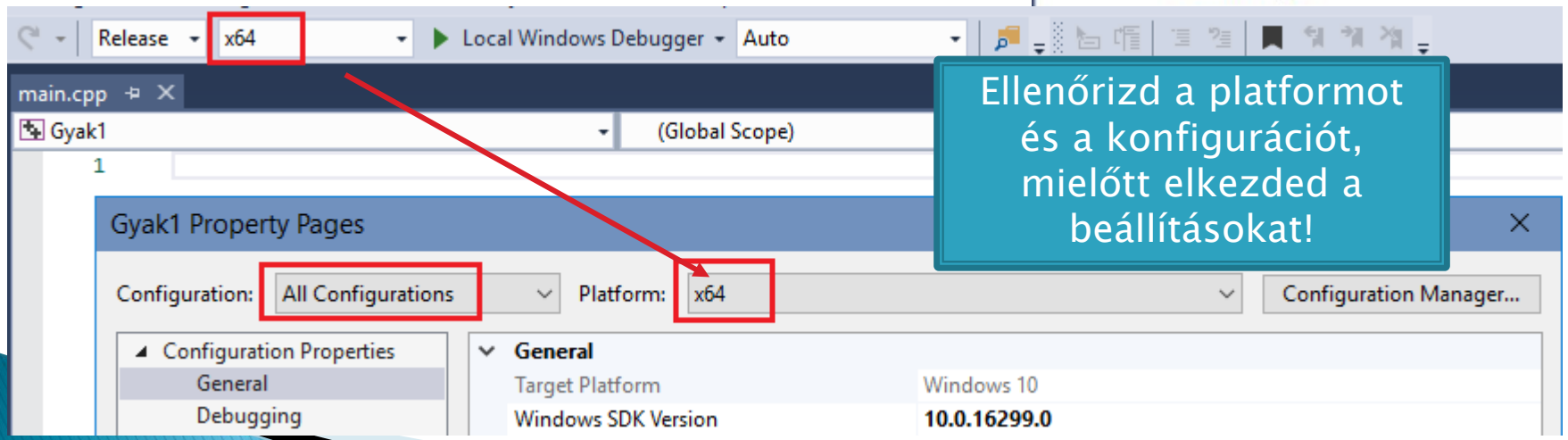
- A Path-hoz adjuk hozzá az OpenCV dll-ek elérési útját:
  - <opencv telepítési könyvtár>\build\x64\<vs>\bin\
    - <vs> := vc15, ha VS 2017-et v. újabb változatot használsz
      vc14, ha VS 2015-öt használsz
  - A környezeti/felhasználói változók közé felvehetjük az OpenCV telepítési könyvtárát:



Projekt beállítása VS alatt I.

- Hozz létre egy üres *C++* projektet (empty)
- Adj hozzá egy main.cpp fájlt
- Nyisd meg a projekt beállításokat:
  - (projekt helyi menüje: Properties)
- A *Platform* legyen x64
- A *Configuration:* All

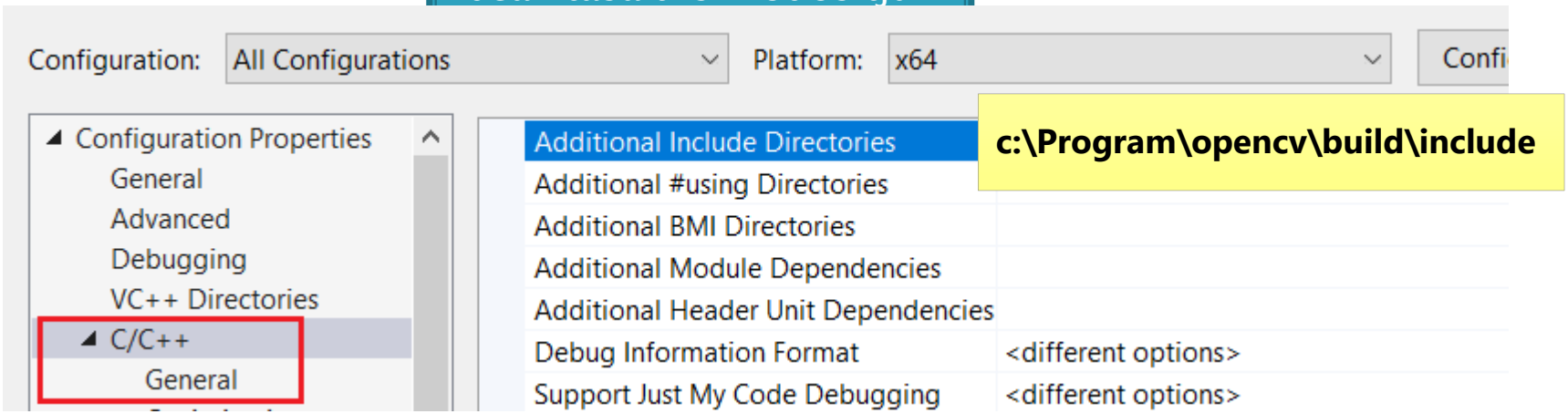




Projekt beállítása VS alatt II.

- Válaszd ki a *C/C++* pontot!
- Add meg az include fájlok elérési útját:
  <opencv telepítési könyvtár>\build\include\

A Release és Debug mód beállításait is módosítjuk



Megj.: ha környezeti változóként (pl. OPENCV_DIR néven) beállítottad az OpenCV telepítési könyvtárát, akkor $(OPENCV_DIR)\build\include\ írható ide.
Előny: a projekt hordozható, csak a környezeti változót kell minden gépen beállítani.

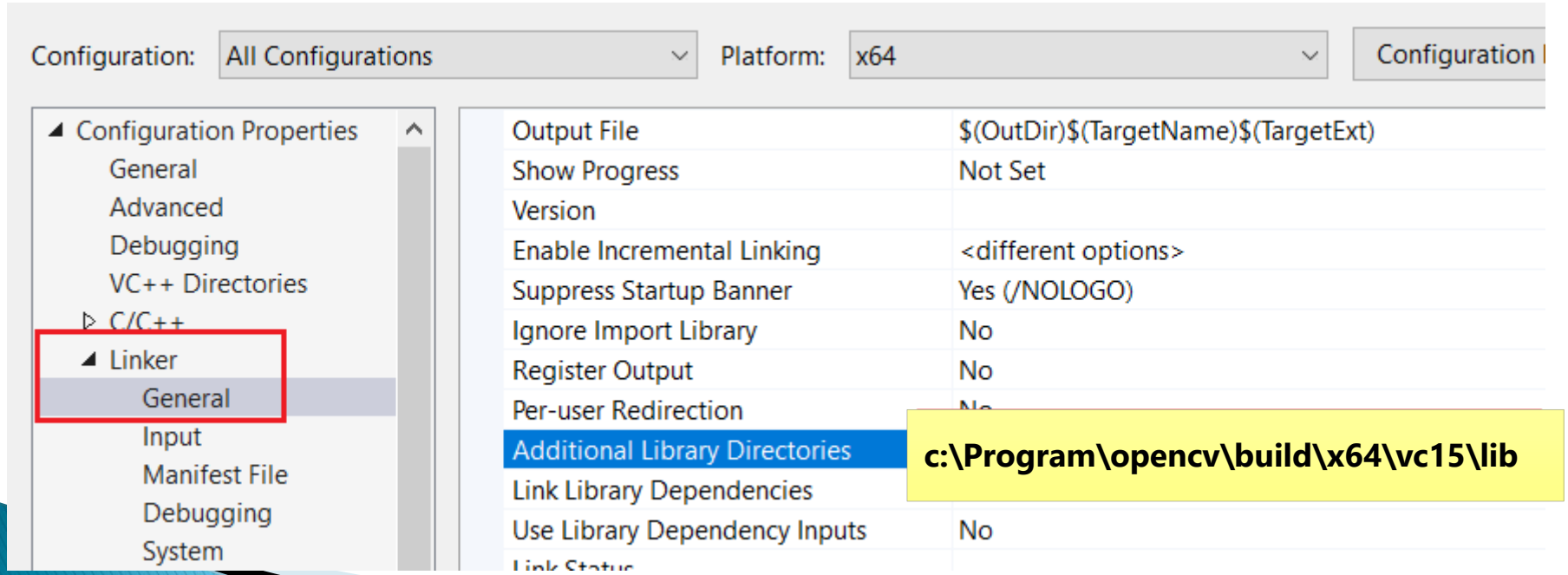
Projekt beállítása VS alatt III.

- Válaszd ki a *Linker* pontot!
- Add meg a lib fájlok elérési útját:

  <opencv telepítési könyvtár>\build\x64\<vs>\lib\

  - <vs> := vc15, ha VS 2017-et,

    vc14, ha VS 2015-öt használsz



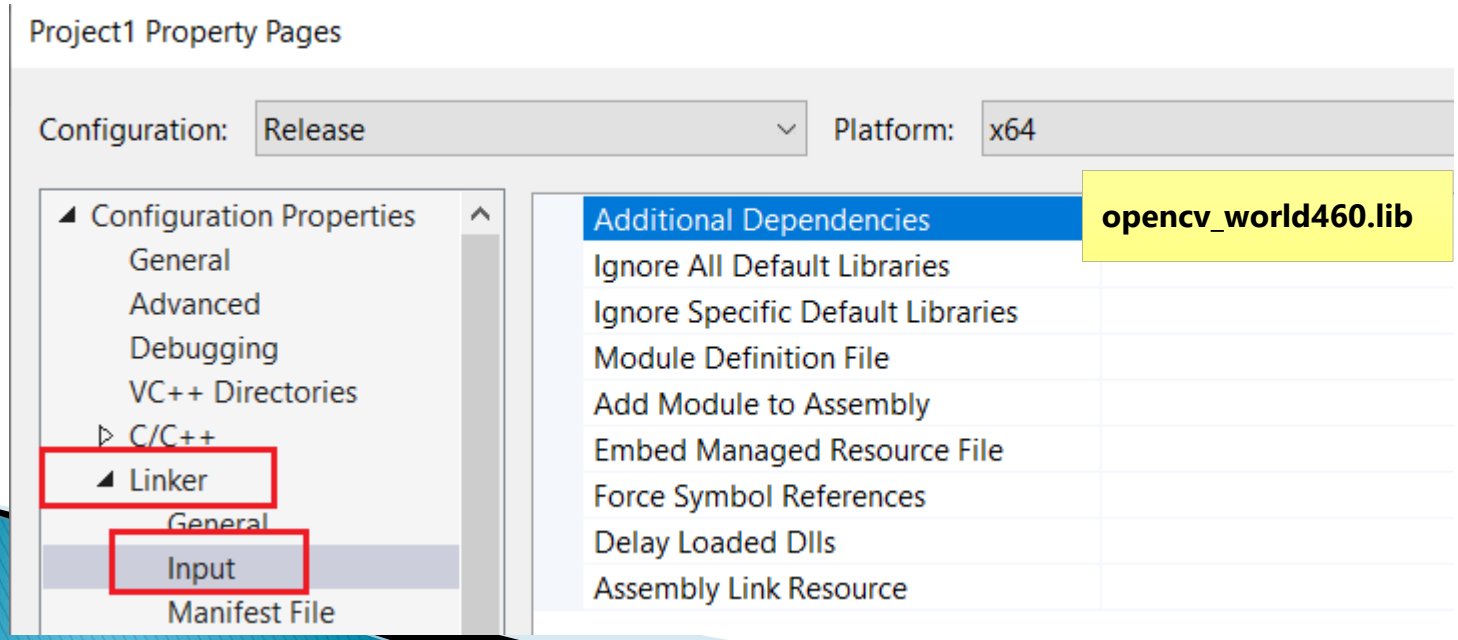
Projekt beállítása VS alatt IV.

2022:
opencv_world460.lib
opencv_world460d.lib

- Válaszd ki a **Linker** alatti **Input** menüpontot!
- Válaszd ki a **Release** konfigurációt és add meg a .lib nevét.

  opencv_world<opencv_verzio>.lib
- Válaszd ki a **Debug** konfigurációt és add meg a .lib nevét.

  opencv_world<opencv_verzio>**d**.lib



Puding próba

- Kép beolvasás, megjelenítés, várakoztatás:

```
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>

int main() {
    cv::Mat img = cv::imread("<képnév elérési úttal>");
    cv::imshow("proba", img);
    cv::waitKey();

    return 0;
}
```

Ha nem találja az include fájlokat:
1) Jó-e az elérési út az Additional **Include** Directories-nél?
2) Stimmelt a konfiguráció?

Ha nem találja a lib-et:
1) Jó-e az elérési út az Additional **Library** Directories-nél?
2) Stimmelt a konfiguráció és a platform?
3) Jól van-e megadva az input lib?

- Névtér megadása:

```
using namespace cv;
```

Ha hiányolja a dll-et:
Nem (jól) adtad hozzá az elérési útvonalhoz a dll-ek helyét. Javítsd/másold a projekt mellé a dll-t.

Mátrix (cv::Mat img)

- Fejlécben szereplő információk lekérdezése
  - img.rows: sorok száma
  - img.cols: oszlopok száma
  - img.channels(): csatornák számának lekérdezése
  - img.type(): típus lekérdezés
  - img.empty(): létezik-e a kép

- Adatrész elérése
  - **img.at<típus>(idx)**
  - img.data[idx]: bájtsorozat (uchar*)

Mátrix létrehozása

- Konstruktorral:

```
Mat name(nrow, ncol, type);

Mat name(Size(ncol, nrow), type);
```

  - nrow, ncol: sorok és oszlopok száma
  - type: az elemek típusa
  - pl.:

```
Mat a(2, 3, CV_8UC1);
Mat b(Size(2, 3), CV_32F);
```

0	255	255
255	0	195

0.3	0.23
0.58	9.5
2.5	19.5

A geometriai objektumoknál a megszokott sorrend a szélesség, magasság, x, y.

A mátrixnál viszont mindig sor, oszlop lesz a sorrend.

- az értékek nem definiáltak, ha 0 mátrixot/fekete képet akarsz:

```
Mat::zeros(nrow, ncol, type);
```

  - pl.:

```
Mat c = Mat::zeros(2, 3, CV_8UC1);
```

Mátrix létrehozása

- A create függvénnyel:

```
Mat nev;  // csak a fejlec
nev.create(sorok_száma, oszlopok_száma, típus);
```

- Előny: ha az adott méretű mátrix már létezik, akkor nem történik újra helyfoglalás.

- Tipikus felhasználás:

```
void my_function(const Mat src, Mat& dest){
     dest.create(src.size(), src.type());
     ...
}
```

OpenCV Mátrix típusai

- Mátrix típusa:

  `CV_<bits><format>[C<channels>]`

  - bits : 8, 16, 32, 64 per channel
  - format: U – unsigned, S – signed, F – float
  - channels: csatorna szám, pl.: 1 – szürkeskála, 3 – színes kép

Mátrix típus	egy elem C típusa	Kép típus (1 csatorna)	egy elem C típusa	Kép típusa (3 channel)	egy elem OpenCV típusa
CV_8U	unsigned char OpenCV: uchar	CV_8UC1	unsigned char OpenCV: uchar	CV_8UC3	Vec3b
CV_8S	char	CV_8SC1	char	CV_8SC3	Vec<char,3>
CV_16U	unsigned short	CV_16UC1	unsigned short	CV_16UC3	Vec3w
CV_16S	short	CV_16SC1	short	CV_16SC3	Vec3s
CV_32S	int	CV_32SC1	int	CV_32SC3	Vec3i
CV_32F	float	CV_32FC1	float	CV_32FC3	Vec3f
CV_64F	double	CV_64FC1	double	CV_64FC3	Vec3d

Feladat: alapvető információk lekérdezése

- Tölts le egy képet a netről (pl.: .png, .bmp, .jpg)
- Olvasd be a képet. Változatlan formában.

```
Mat img = imread("<sajat_kep>", cv::IMREAD_UNCHANGED );
```

- Jelenítsd meg a sorainak és oszlopainak számát
- Kérd le a kép típusát.
- Írd ki a standard outputra, hogy milyen típusú képről van szó:
  - 8-bites szürkeárnyalatos
  - 24-bites (3x8bit) RGB kép
  - vagy egyéb

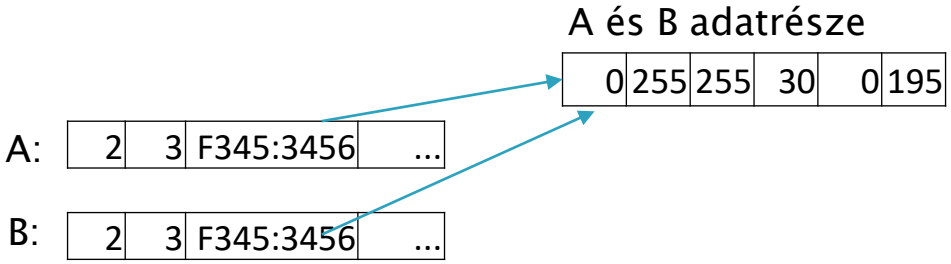
Feladat

- Olvass be egy tetszőleges színes képet.
- Hozz létre ugyanolyan méretben egy 8 bites, szürkeárnyalatos képet.
- A kép legyen tiszta fekete.
- Ellenőrzésként jelenítsd meg mind a két képet.

Mátrix másolása

- Sekély másolat (csak a fejlécet másolja)

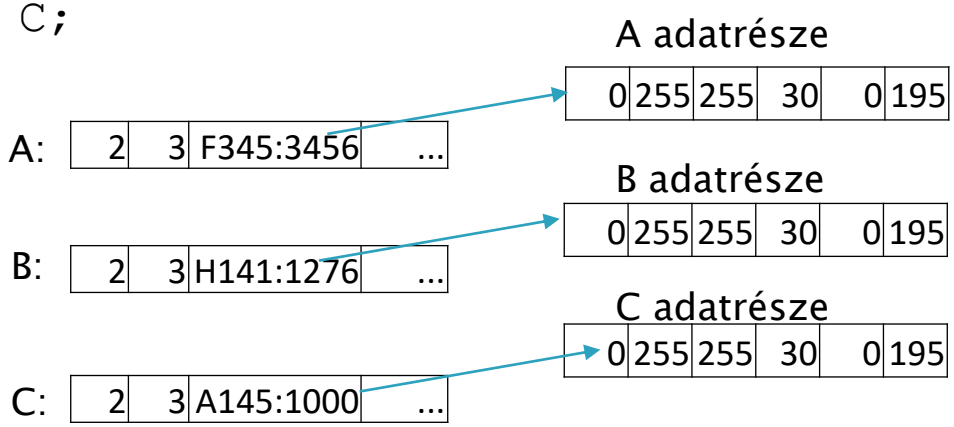
```
Mat A(2, 3, CV_8U), B;
...
B = A;
```



- Mély másolat:

```
Mat A(2, 3, CV_8U), B, C;
...
B = A.clone();

A.copyTo(C [,maszk]);
```



Hasznos mátrix operátorok

- mátrix *operátor* skalár (pontonkénti)

```
uchar data[]{2, 0, 3, 255};       //C++ tömb
Mat A(1, 4, CV_8UC1, data), B;

B = A + 1; // eredmeny: 3, 1, 4, 255 //tartományörző CV_8UC1
B = A - 1; // eredmeny: 1, 0, 2, 3
B = A * 2; // eredmeny : 4, 0, 6, 255
B = A / 2; // eredmeny : 1, 0, 1, 127
```

- mátrix *operátor* mátrix (pontonkénti):

```
B = A + A; // eredmeny : 4, 0, 6, 255
B = A - A; // eredmeny : 0, 0, 0, 0
```

- mátrix megjelenítése:

```
cout << A;
```

Hasznos mátrix operátorok

- mátrix *operátor* skalár (pontonkénti):

```
uchar data[]{2, 0, 4, 255};      //C++ tömb
Mat A(1, 4, CV_8U, data), c;

C = A > 2;  // eredmeny: 0, 0, 1, 1
C = A <= 2; // eredmeny: 1, 1, 0, 0
C = A == 2; // eredmeny: 1, 0, 0, 0
...
```

- mátrix *operátor* mátrix (pontonkénti) :

```
uchar data[]{1, 3, 4, 8};        //C++ tömb
Mat B(1, 4, CV_8U, data), b;

C = A == B; // eredmeny: 0, 0, 1, 0
B = A > B;  // eredmeny: 1, 0, 0, 1
...
```

Hasznos mátrix operátorok

- mátrixok elemeinek **pontonkénti szorzata**

```
Mat C; // for the results

float data[]{2.5, 0, 3};
Mat A(1, 3, CV_32F, data);

float data[]{2, 1, 0};
Mat B(1, 3, CV_32F, data) ;

C = A.mul(B);       // eredmény: 5, 0, 0
```

- matrixszorzás:

```
C = A * B.t();    // B.t() a B transzponáltja
```

$$// \; C = [2.5 \quad 0 \quad 3] \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = 2.5 * 2 + 0 * 1 + 3 * 0 = [5]$$

Feladat: Kép "fakítás"

- Olvass be egy képet szürkeskálában
  - imread(<elérési_út>, ImreadModes::IMREAD_GRAYSCALE);
- Adj hozzá az összes képponthoz egy pozitív egész értéket, pl. 50-et.
- Jelenítd meg az eredetit és a fakóbb változatot is
  - imshow(ablaknév, kép);

A kép, mint paraméter (fejléc vs. adat)

```
void vilagosit1(Mat img) { //Miért változik az eredeti kép?
    img = img + 100;
}
void vilagosit2(Mat img, Mat& dest) { //Miért kell a & jel?
    dest = img + 100;
}

int main() {
    Mat img = cv::imread("d:/kurama.jpg", 0);
    vilagosit1(img);
    imshow("fako1", img);

    Mat dest;                   //A kép még üres
    vilagosit2(img, dest);
    imshow("fako2", dest);
    waitKey();

    return 0;
}
```

Feladat: Invertálás

- Olvass be egy képet szürkeskálában
  - imread(<elérési_út>, ImreadModes::IMREAD_GRAYSCALE);
- Ellenőrizd, hogy sikeres-e a beolvasás
  - empty()
- Invertáld a képet
  - az elméleti maximum 255
- Jelenítsd meg az eredetit és az invertált képet is
  - imshow(ablaknév, kép);

Feladat: Mosd össze

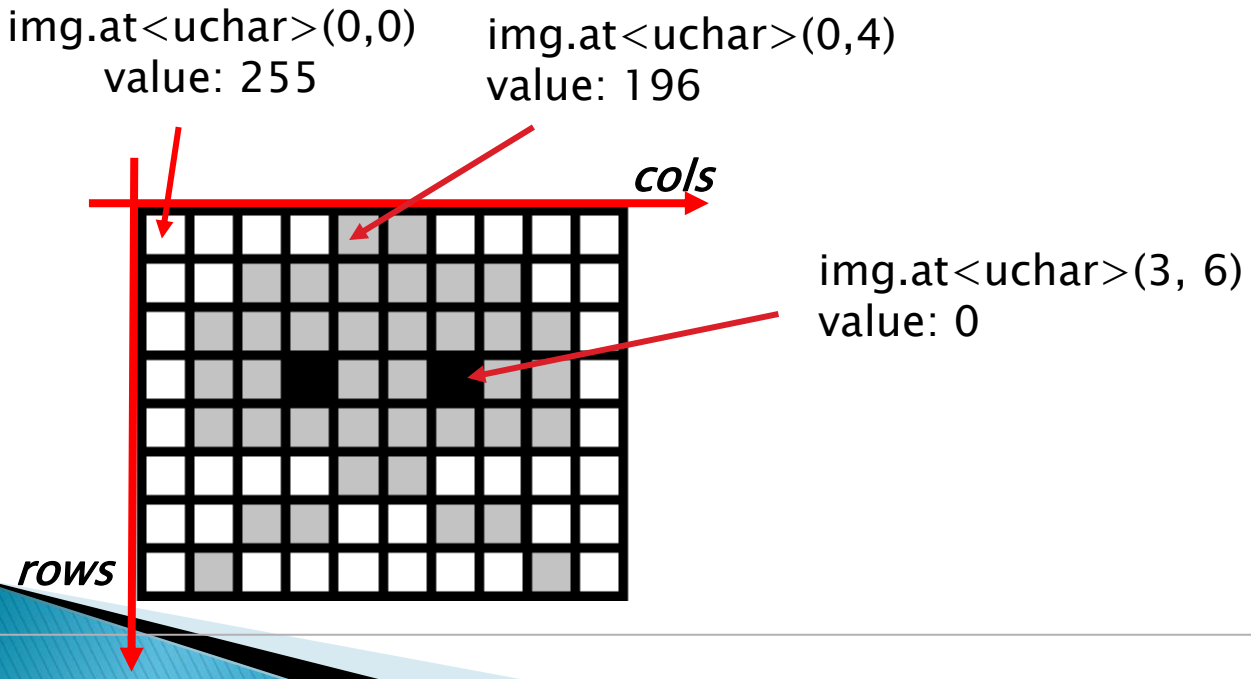
- Olvass be egy háttérképet (*h*)
- Olvass be egy rajzot *(r)*
- Ellenőrizd, hogy sikeres-e a beolvasás
- Méretezd át a hátteret úgy, hogy illeszkedjen az előtérhez:
  - #include <opencv2/imgproc.hpp>
  - resize(eredeti_kép, cél_kép, cél_méret)
- Add össze a két képet: $\mathrm{u} = \alpha \cdot h + (1 - \alpha) \cdot r, \quad 0 < \alpha < 1$
- Jelenítsd meg: imshow(ablaknév, kép);

Képpontok elérése

- *.at* (template) metódussal

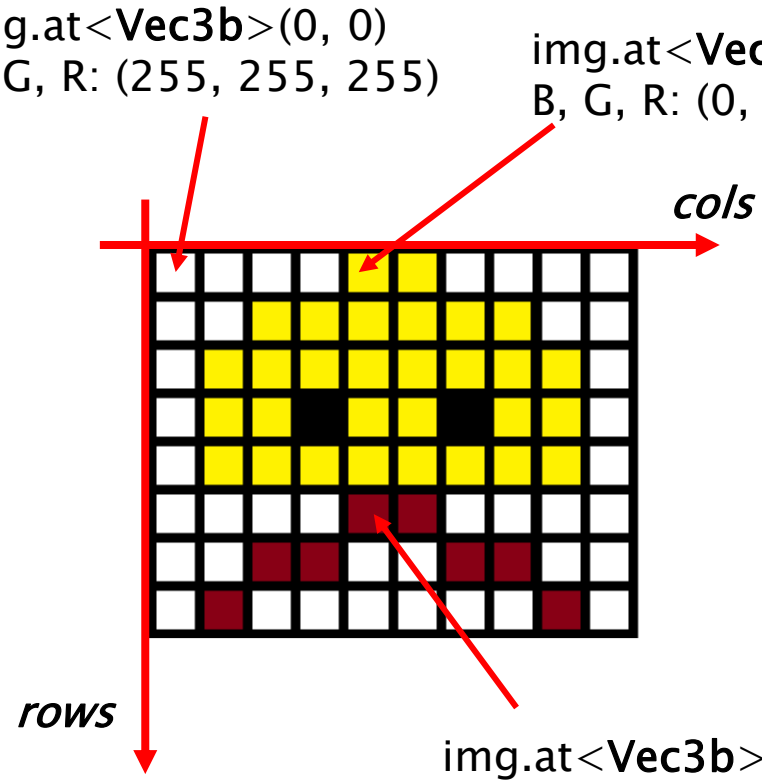
```
Mat img = imread(...)
img.at<pixel_type>(row_idx, column_idx)
img.at<pixel_type>(Point(column_idx, row_idx))
```

- pl.: *img* egy 8 bites, szürkeskálás kép (CV_8UC1)



Képpontok elérése

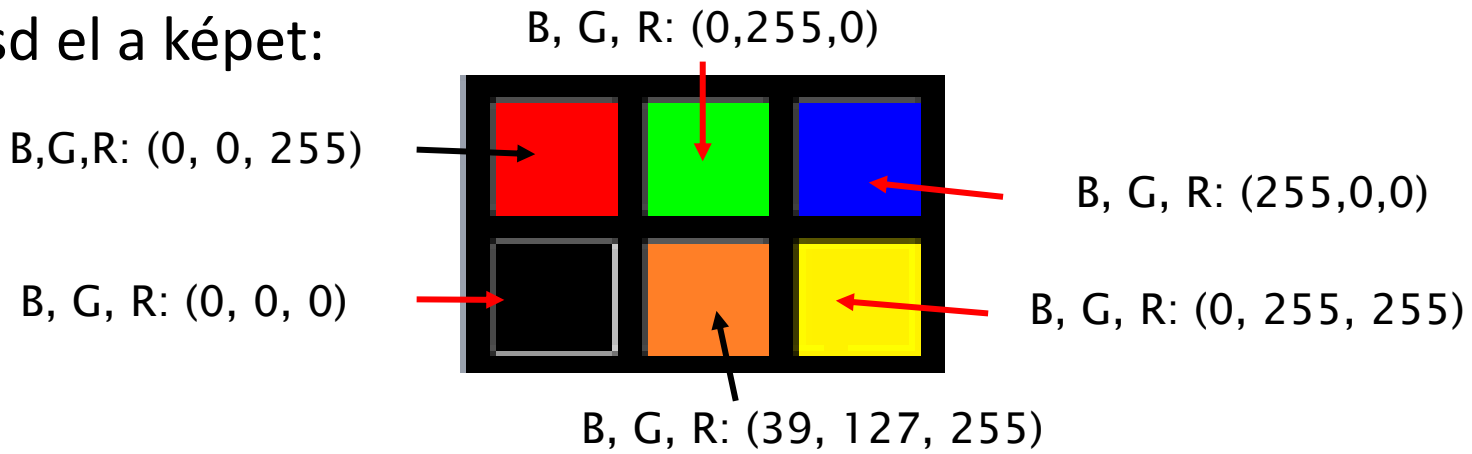
- pl.: az *img* egy 24 bites RGB kép (CV_8UC3)
- OpenCV-ben az alapértelmezett csatornasorrend a B,G,R



Kék:
img.at<Vec3b>(0, 3)**[0]** //0
Zöld:
img.at<Vec3b>(0, 3)**[1]** //255
Piros komponens:
img.at<Vec3b>(0, 3)**[2]** //255

Feladat

- Készítsd el a képet:




```
...
Mat m(2, 3, CV_8UC3)  //egy 3 csatornás kép készítése

// TO DO: állíts be a pixelek színeit a .at segítségével.

namedWindow("sample", WINDOW_NORMAL); //átméretezhető ablak
resizeWindow("sample", 120, 80)       //nagyítás
imshow("sample", m);                  //megjelenítés
waitKey(0)                            //várakozás
...
```

Template mátrix

- Mat_
- Akkor használható, ha a kép típusa ismert (ez gyakran igaz)
- Egyszerűsödik a képpont elérés

- Példa:

```
Mat_<uchar> img = imread("<egy kép>", IMREAD_GRAYSCALE);
cout << img(0, 0);


Mat_<Vec3b> img2 = imread("<egy kép>", IMREAD_COLOR);
cout << img2(1, 2);

img2(1, 2) = Vec3b(2, 3, 4)
img2(1, 2) = (2, 3, 4)  // VIGYÁZZ !!!
```

Ha elhagyod a Vec3b-t a jobb oldal értéke **4**.
(C++ vessző operátor: a, b, c, …, z értéke z.

Ez olyan, mintha a Vec3b(4, 0, 0)-t használnád.


99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.
127 little bugs in the code...