Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2020-2021

We

Bogdan

Iuliana

You

407 (AI)

Computer Vision		
Nr.	**Nume si prenume**	**Grupa**
1	Badea Armeanu Razvan Ilie	405
2	Grigoras Dorin	405
3	Năbîrgeac Cezar Ştefan	405
4	Onetiu Radu	405
5	Andrei Tudor Ionescu	406
6	Anei Alexandra Gabriela	406
7	Apostu Robert	406
8	Brînzea Alexandru Eusebiu	406
9	Cîlţea Marian	406
10	Costachi T. Ana Maria	406
11	Darii Dan	406
12	Matei Bianca Gabriela	406
13	Munteanu Beatrice Madalina	406
14	Petrasco Sandu	406
15	Pogonaru Stefan	406
16	Stinga Andreea	406
17	Valeria Bolun	406
18	Hodorogea Dragos Cristian	410
19	Acsintoae Andra Maria	411
20	Aslantas Serdal	411
21	Chirila Diana Gabriela	411
22	Chirita Catalina	411
23	Coman Catalina Mihaela	411
24	Cutic Eugen	411
25	Dutu Daniel	411
26	Fulea Andrei	411
27	Menadil Rafael Edy	411
28	Patulea Mihaela Alina	411
29	Deaconu Stefan Eduard	412
30	Iordache Ioan Bogdan	412
31	Manea Andrei Alexandru	412

MASTER Anul I, 2020 - 2021, Învățământ cu frecvență
„ARTIFICIAL INTELLIGENCE”, Program de 4 semestre

(Sem.I – 14 săptămâni; Sem.II – 14 săptămâni)

Nr. crt.	Cursuri obligatorii	Semestrul I (14 săpt.)				Semestrul II (14 săpt.)			
		C	S/L	E/V	ECTS	C	S/L	E/V	ECTS
1	Ob.11 **Învățare automata aplicată** Practical Machine Learning	2	1	V	6	-	-	-	-
2	Ob.12 **Programare probabilistă** Probabilistic Programming	2	1	E	6	-	-	-	-
3	Ob.13 **Reprezentarea cunoștințelor și inferență** Knowledge representations and reasoning	2	1	E	6	-	-	-	-
4	Op.14 **Curs optional** Optional Course	2	1	E	6	-	-	-	-
5	Ob.15 **Practică**/Practical Training	-	2	V	4	-	-	-	-
6	Ob.16 **Deontologie Academică**/Academic Deontology	1	-	V	2				
7	Ob.21 **Învățare automată** Advanced machine learning	-	-	-	-	2	1	E	6
8	Ob.22 **Vedere Artificială** Computer Vision	-	-	-	-	2	1	E	6
9	Ob.23 **Procesarea limbajului natural 1** Natural language processing 1	-	-	-	-	2	1	E	6
10	Op.24 **Curs optional** Optional Course					2	1	E	6
11	Ob.25 **Practică**/ Practical Training	-	-	-	-	-	2	V	6
TOTAL		**9**	**6**	**3E 3V**	**30**	**8**	**6**	**4E 1V**	**30**

MASTER Anul I, 2020- 2021, Învățământ cu frecvență
„DATA SCIENCE”, Program de 4 semestre

(Sem.I – 14 săptămâni; Sem.II – 14 săptămâni)

Nr. crt.	Cursuri obligatorii și opționale	Semestrul I (14 săpt.)				Semestrul II (14 săpt.)			
		C	S/L	E/V	ECTS	C	S/L	E/V	ECTS
1	Ob.11 **Învățare automată aplicată** Practical Machine Learning	2	1	E	6	-	-	-	-
2	Ob.12 **Programare probabilistă** Probabilistic Programming	2	1	E	6	-	-	-	-
3	Ob.13 **Analiza exploratorie a datelor** Exploratory Data Analysis	2	1	E	6	-	-	-	-
4	Ob.14 **Curs optional 1** Optional Course 1	2	1	E	6	-	-	-	-
5	Ob.16 **Deontologie Academică**/Academic Deontology	1	-	V	2				
6	Ob.15 **Practică**/Practical Training	-	2	V	4	-	-	-	-
7	Ob.21 **Învățare automată** Advanced Machine Learning	-	-	-	-	2	1	E	6
8	Ob.22 **Big Data** Big Data	-	-	-	-	2	1	E	6
9	Ob.23 **Metode statistice pentru data science** Statistics for Data Science	-	-	-	-	2	1	E	6
10	Ob.24 **Curs optional 2** Optional Course 2					2	1	E	6
11	Ob.25 **Practică**/ Practical Training	-	-	-	-	-	2	V	6
TOTAL		**9**	**6**	**4E 2V**	**30**	**8**	**6**	**4E 1V**	**30**

ADMINISTRATIVE

- Lecture: Tuesday, 16-18 (Teams)
- Lab: Thursday, 18-20 (Iuliana) or Friday, 8-10 (Bogdan)
- One lab session every two weeks

- Bogdan – TEAMS (lecture + lab)
  - channel Lecture: slides + recordings
  - channel Lab: pdfs + recordings + solutions

- Iuliana – Google Meet (lab)
  - group 3: https://hangouts.google.com/group/DutRAdQKHSKZ2rYP6
  - group 4: https://hangouts.google.com/group/bJXFKij6u665dwGK9

Exam – evaluation in June

Grade = min(10, P1 + P2)

- P1 = project 1 = 5 points (+ some bonus?)
- P2 = project 2 = 5 points (+ some bonus?)
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in July (restanță)

Grade = max($2 \times P3$, $P1 + P3$, $P2 + P3$)

- P3 = project 3 = 5 points
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in September (reexaminare)

$$\text{Grade} = \max(2 \times P4,\ P1 + P4,\ P2 + P4,\ P3 + P4)$$

- P4 = project 4 = 5 points
- no constraints, with 4.99 you fail, with 5 you pass

Exam – evaluation in September (mărire)

Grade = $2 \times P4$

- P4 = project 4 = 5 points

Resources

github.com/jbhuang0604/awesome-computer-vision/

Awesome Computer Vision: awesome

A curated list of awesome computer vision resources, inspired by awesome-php.

For a list people in computer vision listed with their academic genealogy, please visit here

Contributing

Please feel free to send me pull requests or email (jbhuang@vt.edu) to add links.

Table of Contents

Books

Computer Vision

- Computer Vision: Models, Learning, and Inference - Simon J. D. Prince 2012
- Computer Vision: Theory and Application - Rick Szeliski 2010
- Computer Vision: A Modern Approach (2nd edition) - David Forsyth and Jean Ponce 2011
- Multiple View Geometry in Computer Vision - Richard Hartley and Andrew Zisserman 2004
- Computer Vision - Linda G. Shapiro 2001
- Vision Science: Photons to Phenomenology - Stephen E. Palmer 1999
- Visual Object Recognition synthesis lecture - Kristen Grauman and Bastian Leibe 2011
- Computer Vision for Visual Effects - Richard J. Radke, 2012
- High dynamic range imaging: acquisition, display, and image-based lighting - Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G., Myszkowski, K 2010
- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics - Justin Solomon 2015

OpenCV Programming

- Learning OpenCV: Computer Vision with the OpenCV Library - Gary Bradski and Adrian Kaehler
- Practical Python and OpenCV - Adrian Rosebrock
- OpenCV Essentials - Oscar Deniz Suarez, Mª del Milagro Fernandez Carrobles, Noelia Vallez Enano, Gloria Bueno Garcia, Ismael Serrano Gracia

Courses

Computer Vision

- EENG 512 / CSCI 512 - Computer Vision - William Hoff (Colorado School of Mines)
- Visual Object and Activity Recognition - Alexei A. Efros and Trevor Darrell (UC Berkeley)
- Computer Vision - Steve Seitz (University of Washington)
- Visual Recognition Spring 2016, Fall 2016 - Kristen Grauman (UT Austin)
- Language and Vision - Tamara Berg (UNC Chapel Hill)
- Convolutional Neural Networks for Visual Recognition - Fei-Fei Li and Andrej Karpathy (Stanford University)
- Computer Vision - Rob Fergus (NYU)
- Computer Vision - Derek Hoiem (UIUC)
- Computer Vision: Foundations and Applications - Kalanit Grill-Spector and Fei-Fei Li (Stanford University)
- High-Level Vision: Behaviors, Neurons and Computational Models - Fei-Fei Li (Stanford University)
- Advances in Computer Vision - Antonio Torralba and Bill Freeman (MIT)
- Computer Vision - Bastian Leibe (RWTH Aachen University)
- Computer Vision 2 - Bastian Leibe (RWTH Aachen University)
- Computer Vision Pascal Fua (EPFL):
- Computer Vision 1 Carsten Rother (TU Dresden):
- Computer Vision 2 Carsten Rother (TU Dresden):
- Multiple View Geometry Daniel Cremers (TU Munich):

Computer Vision

- Computer Vision Talks - Lectures, keynotes, panel discussions on computer vision
- The Three R's of Computer Vision - Jitendra Malik (UC Berkeley) 2013
- Applications to Machine Vision - Andrew Blake (Microsoft Research) 2008
- The Future of Image Search - Jitendra Malik (UC Berkeley) 2008
- Should I do a PhD in Computer Vision? - Fatih Porikli (Australian National University)
- Graduate Summer School 2013: Computer Vision - IPAM, 2013

Recent Conference Talks

- CVPR 2015 - Jun 2015
- ECCV 2014 - Sep 2014
- CVPR 2014 - Jun 2014
- ICCV 2013 - Dec 2013
- ICML 2013 - Jul 2013
- CVPR 2013 - Jun 2013
- ECCV 2012 - Oct 2012
- ICML 2012 - Jun 2012
- CVPR 2012 - Jun 2012

What is Computer Vision?

Are we done?

Slide credit K. Graumann

What is Computer Vision?

Make computers understand images and video.

What kind of scene?

Where are the cars?

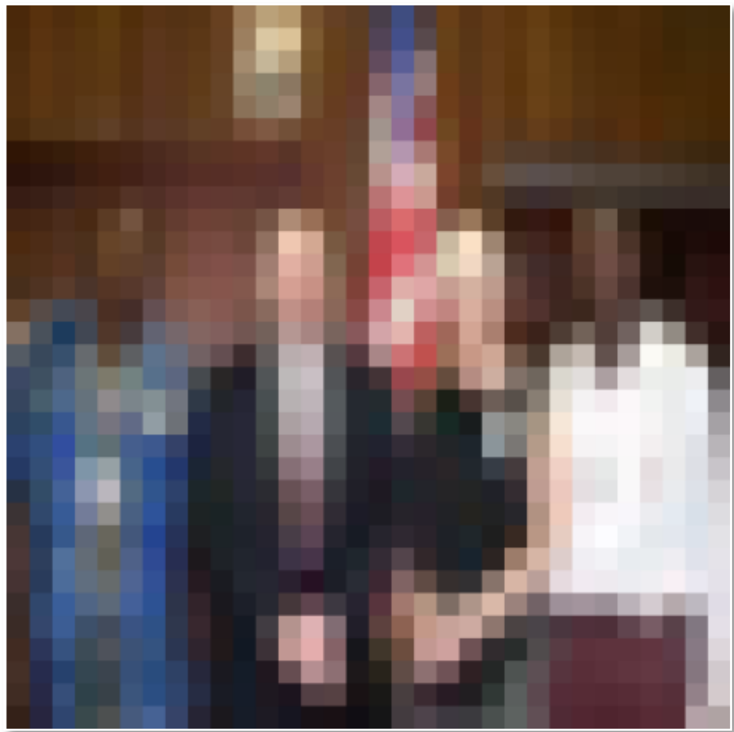
How far is the building?

…

Slide credit D. Hoiem

The goal of Computer Vision

- Extract information from pixels



Slide credit S. Lazebnik

The goal of Computer Vision

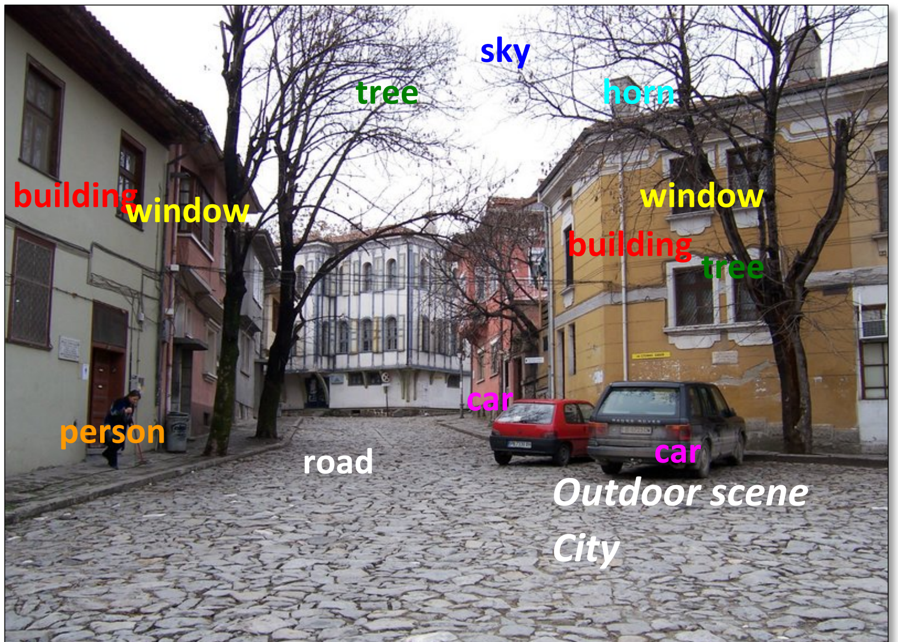
- Extract information from pixels

Human vision

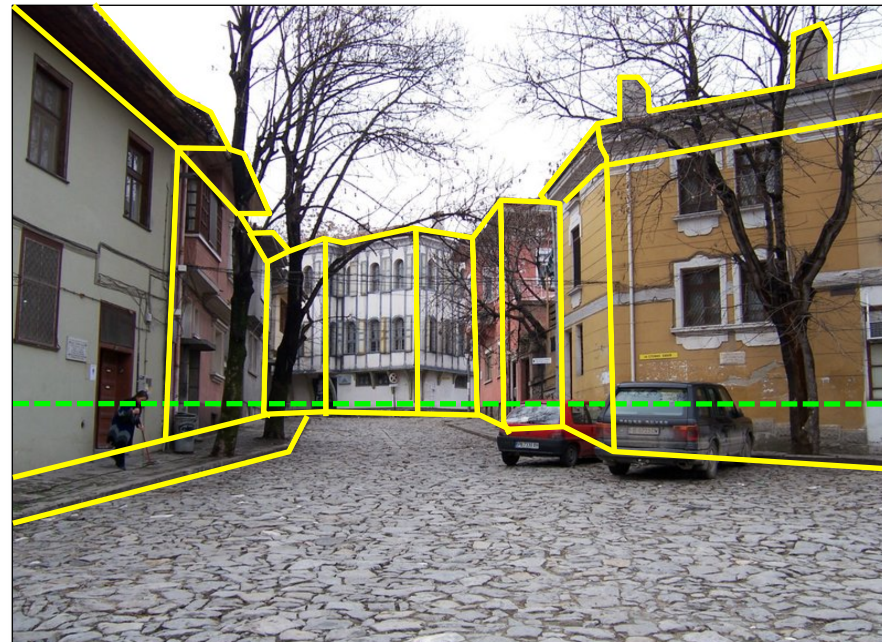
Computer vision

Slide credit S. Lazebnik

What kind of information?



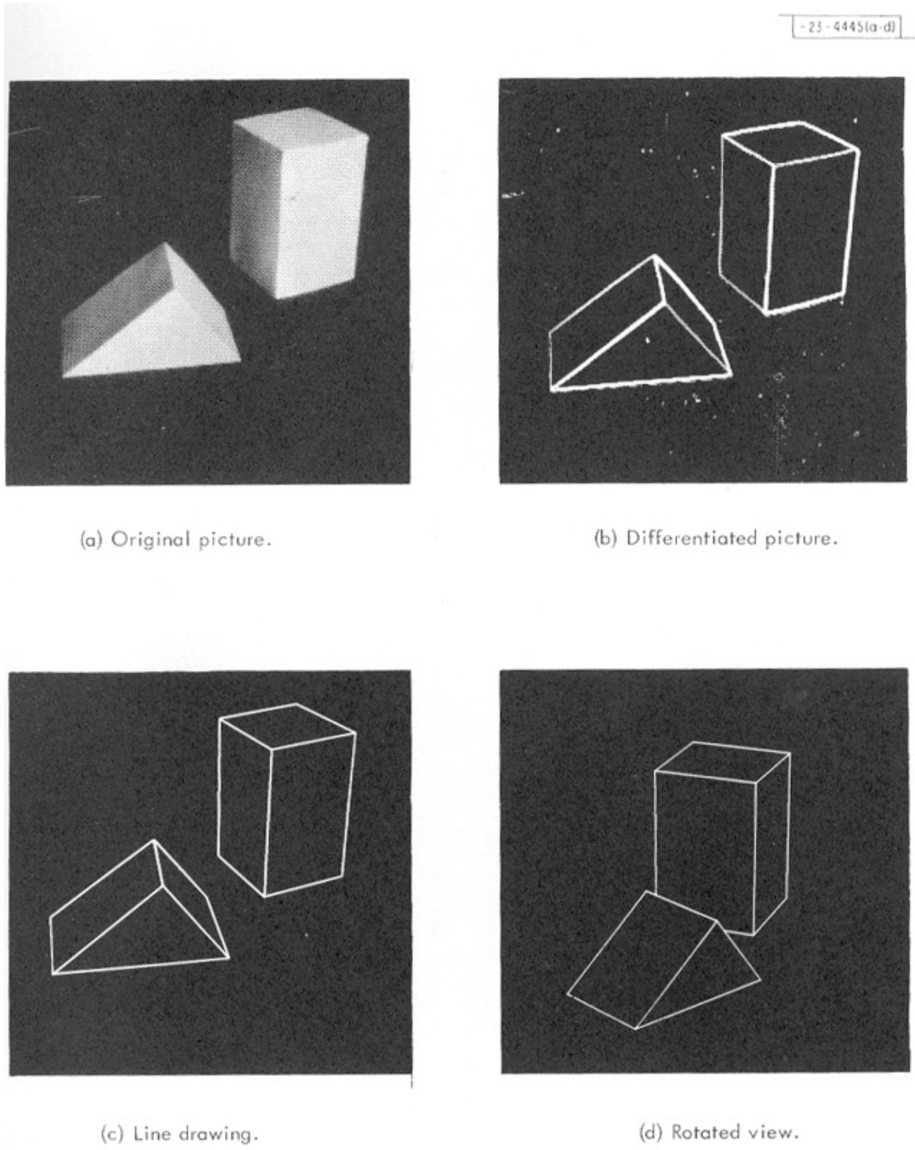
Semantic information



Geometric information (3D)

Slide credit S. Lazebnik

Visual data in 1963



(a) Original picture.

(b) Differentiated picture.

(c) Line drawing.

(d) Rotated view.

L. G. Roberts
Machine Perception of Three Dimensional Solids,
Doctoral thesis, MIT, 1963.

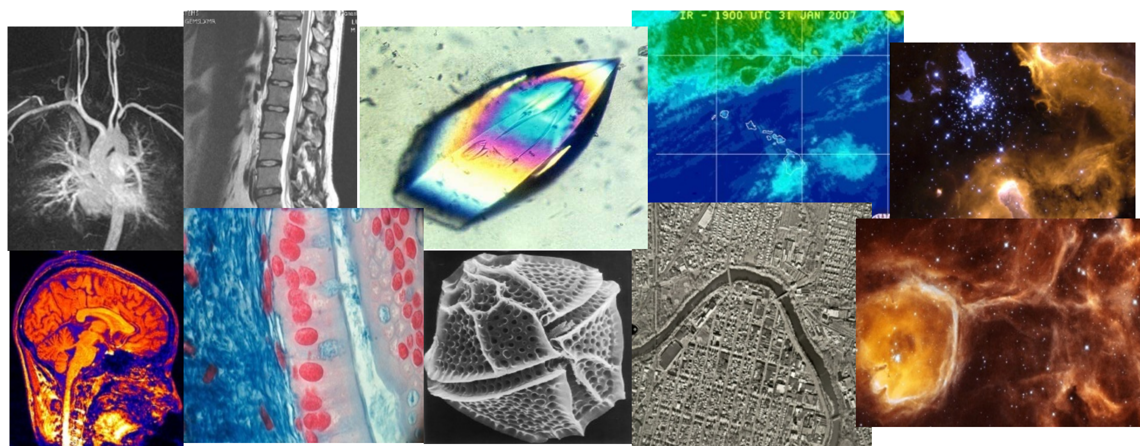
Visual data in 2021

Photo albums

Movies, news, sports

Google Image Search · Picasa · flickr · facebook · twitter · YouTube Broadcast Yourself

Video surveillance and security



Medical imaging

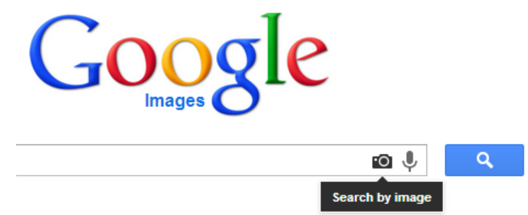
Slide credit S. Lazebnik

Successful applications in Computer Vision

OCR

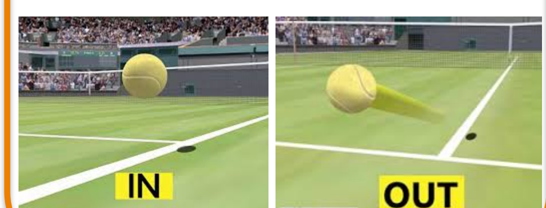
Xbox Kinect

Visual search

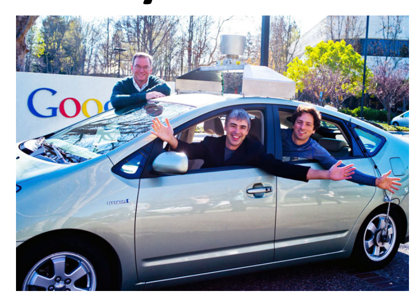


Face detection / recognition

Hawk-eye decision system



Driving assistance systems



Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

Computer Vision projects for undergraduate students

Context aware image resizing

Initial image

Resize the image:
increase/decrease width/height of the image

Main idea: add/remove
irregularly shaped "seams"

How to chose?

[Shai & Avidan, SIGGRAPH 2007]

Slide credit K. Grauman

Context aware image resizing

Context aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

Slide credit K. Grauman

Main idea: Context aware image resizing

Context aware resizing

Intuition:

- Preserve the most "interesting" content
  → Prefer to remove pixels with low gradient energy
- To reduce or increase size in one dimension, remove or add irregularly shaped "seams"
  → Optimal solution via dynamic programming.

Slide credit K. Grauman

Main idea: seam carving

image I

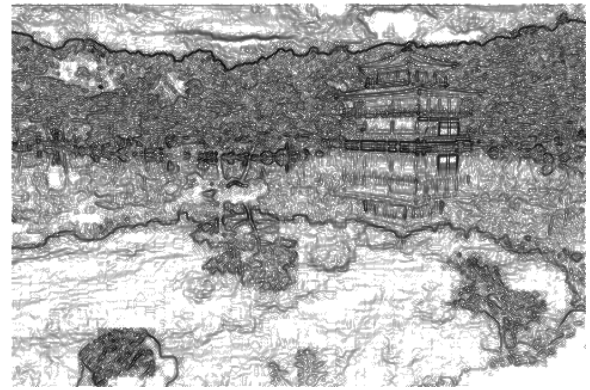


Image gradient

$$\nabla I = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$$

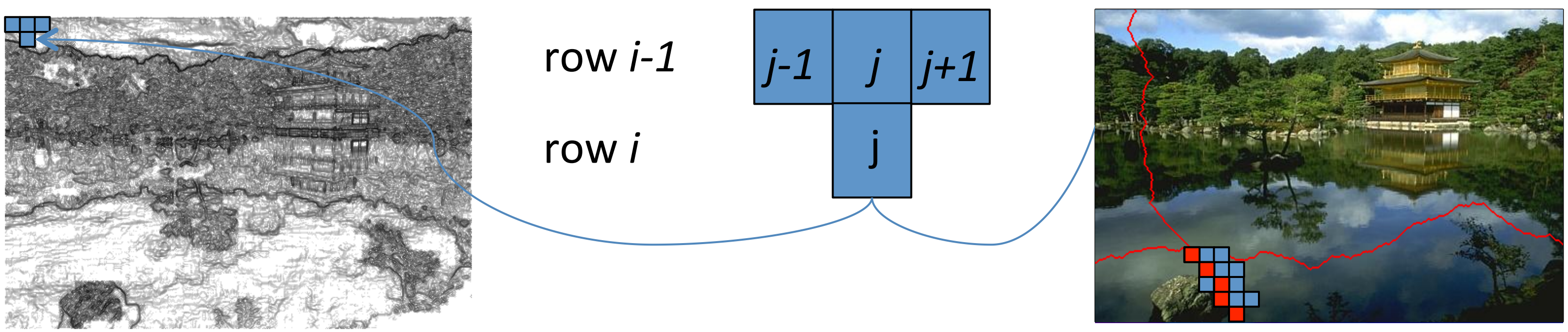
- Want to remove seams where they won't be very noticeable:
  - measure "energy" as gradient magnitude
- Choose seam based on **minimum total energy path** across image, subject to 8-connectedness.

Slide credit K. Grauman

Seam carving: algorithm

- Compute the cumulative minimum energy for all possible connected seams at each entry *(i,j)*:

$$\mathbf{M}(i,j) = MagGradient(i,j) + \min(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1))$$



Energy matrix (gradient magnitude)

M matrix: cumulative min energy (for vertical seams)

- Then, min value in last row of **M** indicates end of the minimal connected vertical seam.

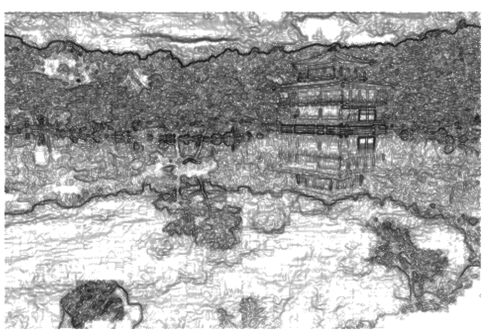
- Backtrack up from there, selecting min of 3 above in **M**.

Slide credit K. Grauman

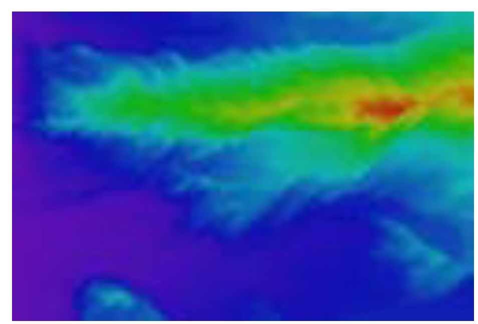
Example

$$\mathbf{M}(i, j) = Energy(i, j) + \min\left(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1)\right)$$

1	3	0
2	8	9
5	2	6



Energy matrix
(gradient magnitude)



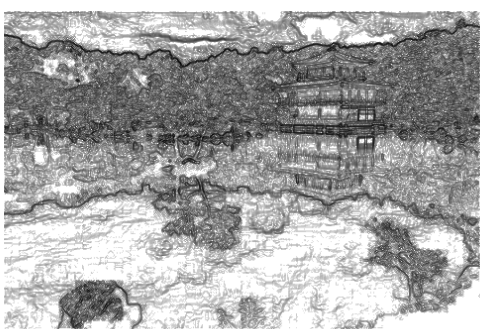
M matrix
(for vertical seams)

Slide credit K. Grauman

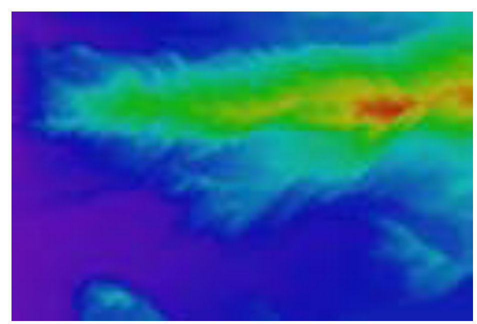
Example

$$\mathbf{M}(i,j) = Energy(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$

$$\begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{2} & 8 & 9 \\ 5 & \boxed{2} & 6 \end{matrix} \qquad\qquad \begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{3} & 8 & 9 \\ 8 & \boxed{5} & 14 \end{matrix}$$



Energy matrix (gradient magnitude)



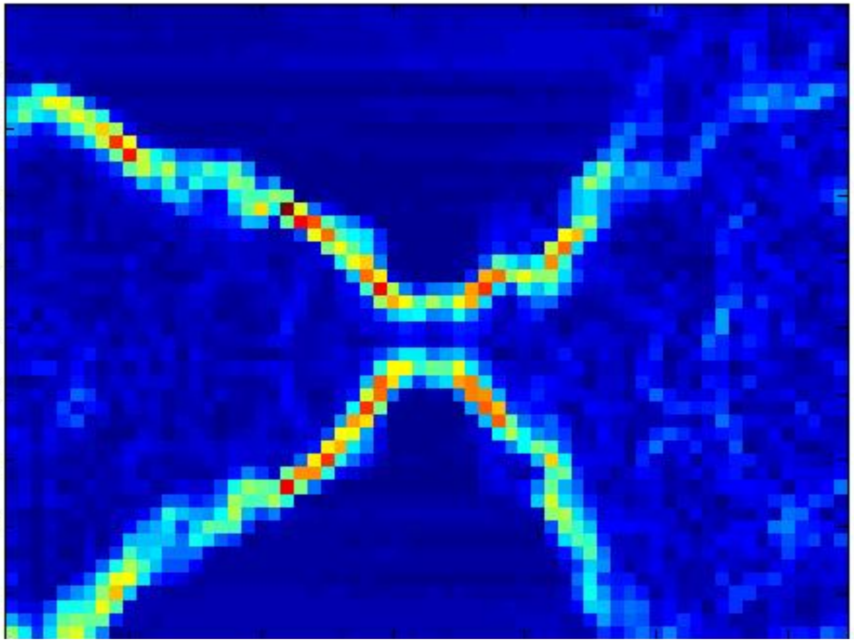
M matrix (for vertical seams)

Slide credit K. Grauman

Real image example

Original Image

Energy Map



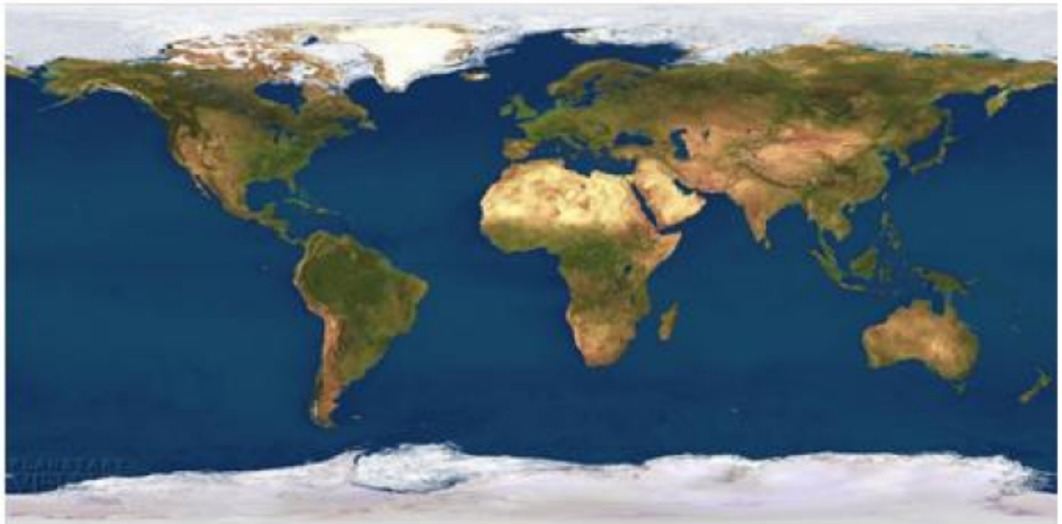
Blue = low energy
Red = high energy

Slide credit K. Grauman

Real image example

Slide credit K. Grauman

Results



Context aware image resizing

Traditional resizing

Slide credit K. Grauman

“Failure cases” with seam carving

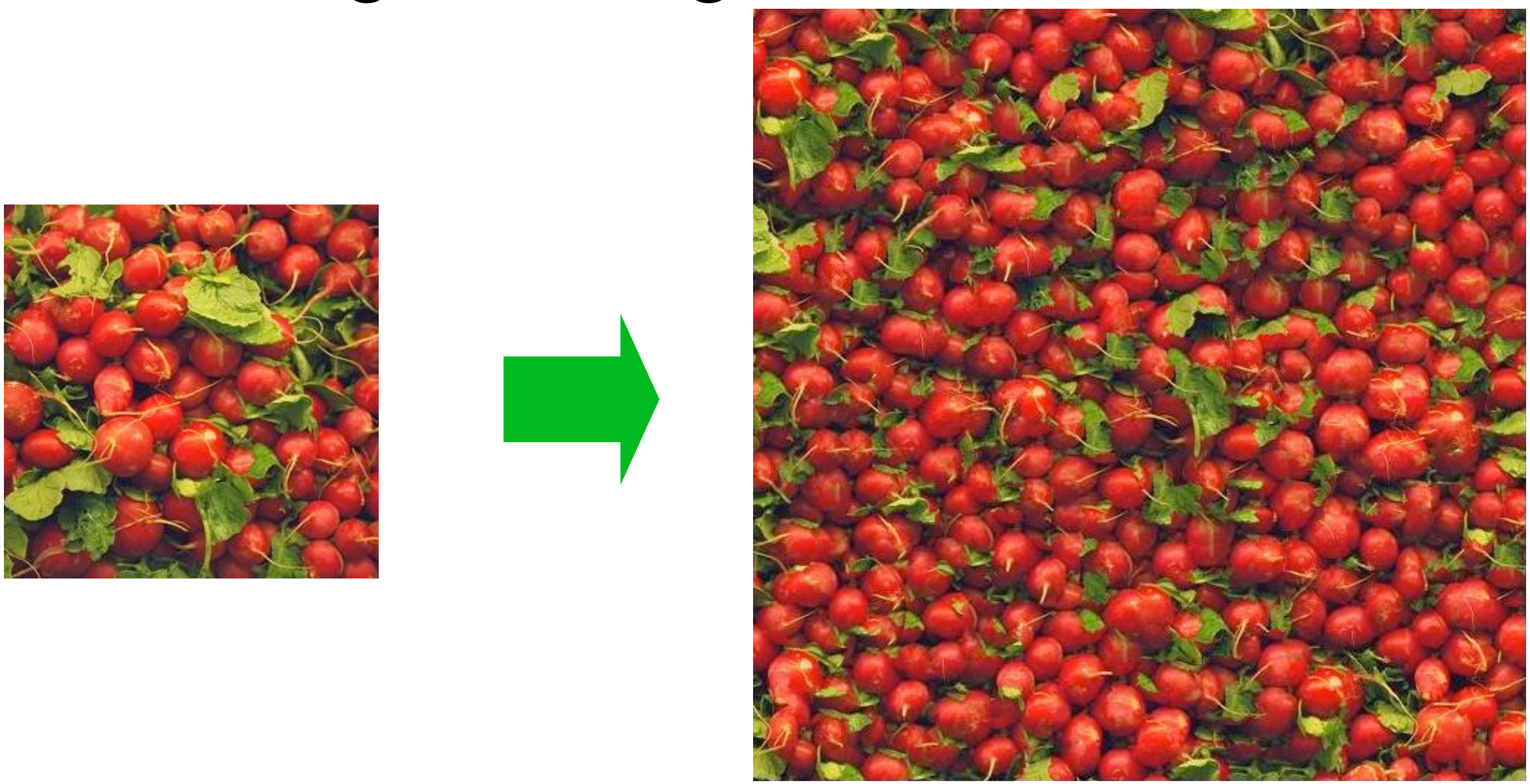
Slide credit K. Grauman

Removal of a marked object

Slide credit K. Grauman

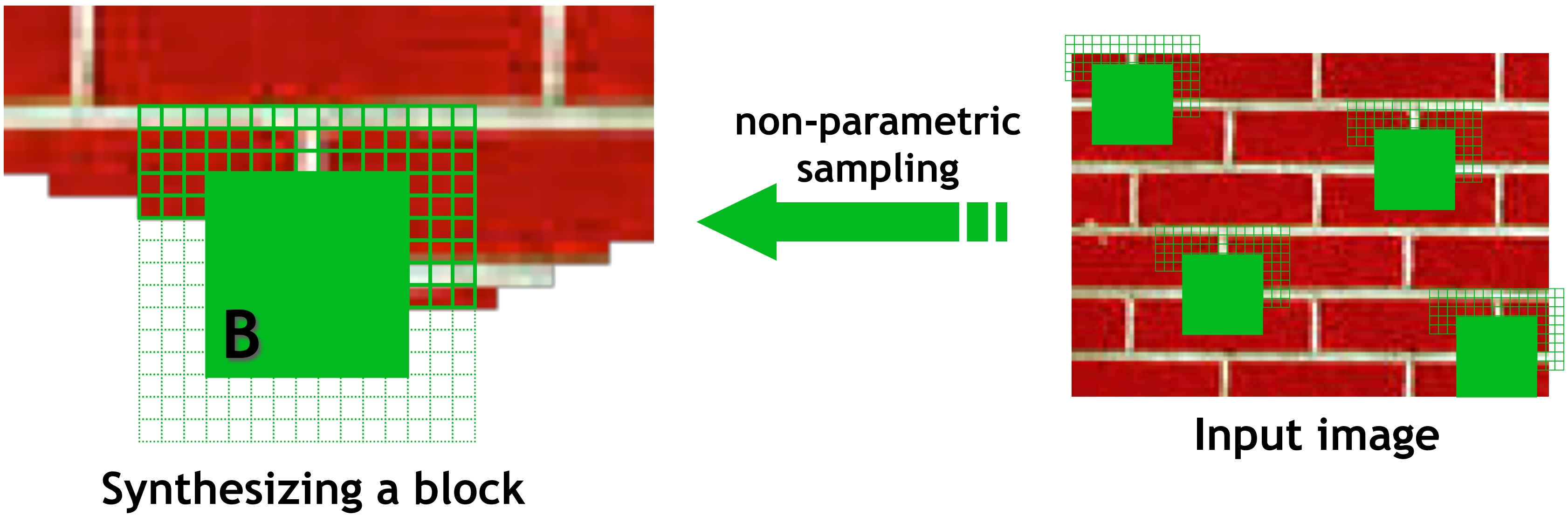
Texture synthesis

- Goal: create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces



Slide credit A. Efros

Image Quilting [Efros & Freeman 2001]

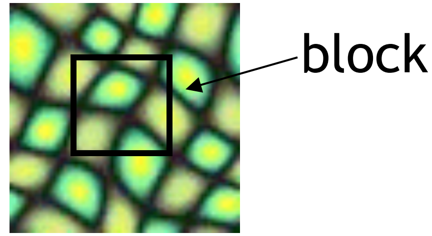


- Observation: neighbor pixels are highly correlated

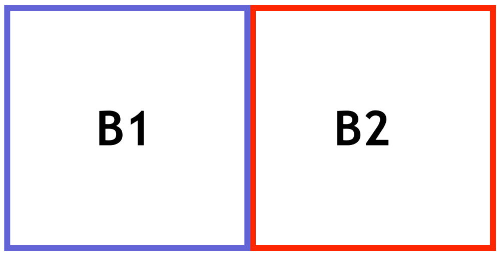
Idea: unit of synthesis = block

- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

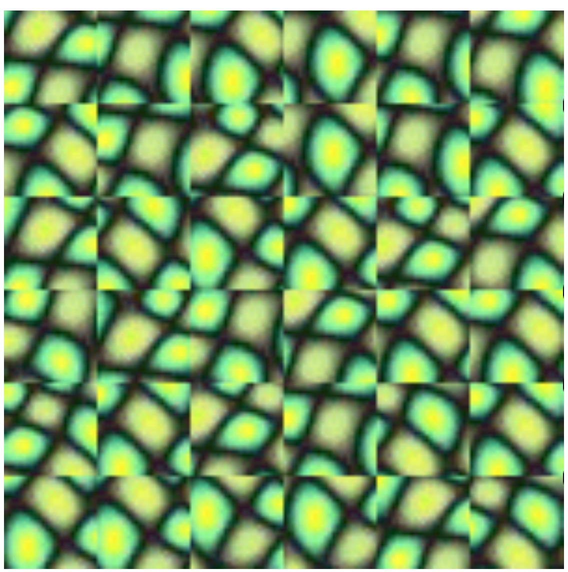
Slide credit A. Efros

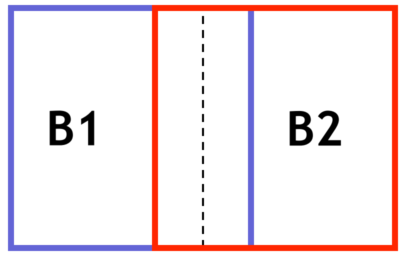


Input texture

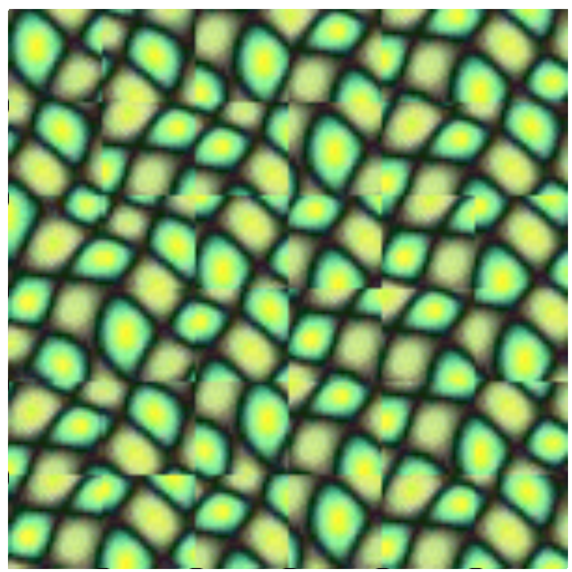


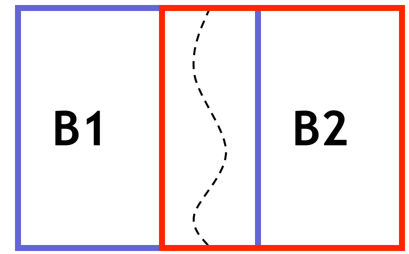
Random placement of blocks



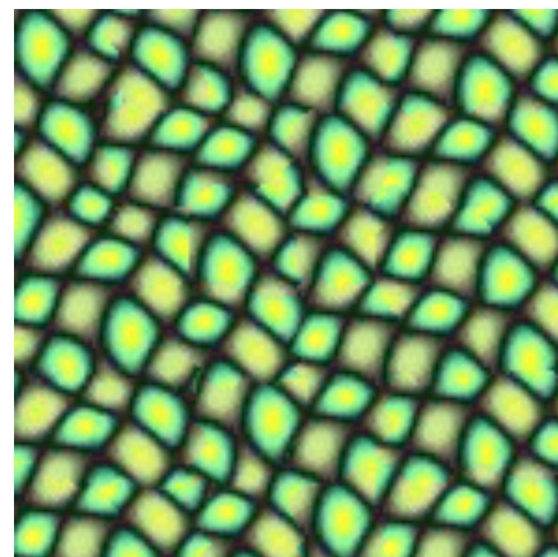


Neighboring blocks constrained by overlap



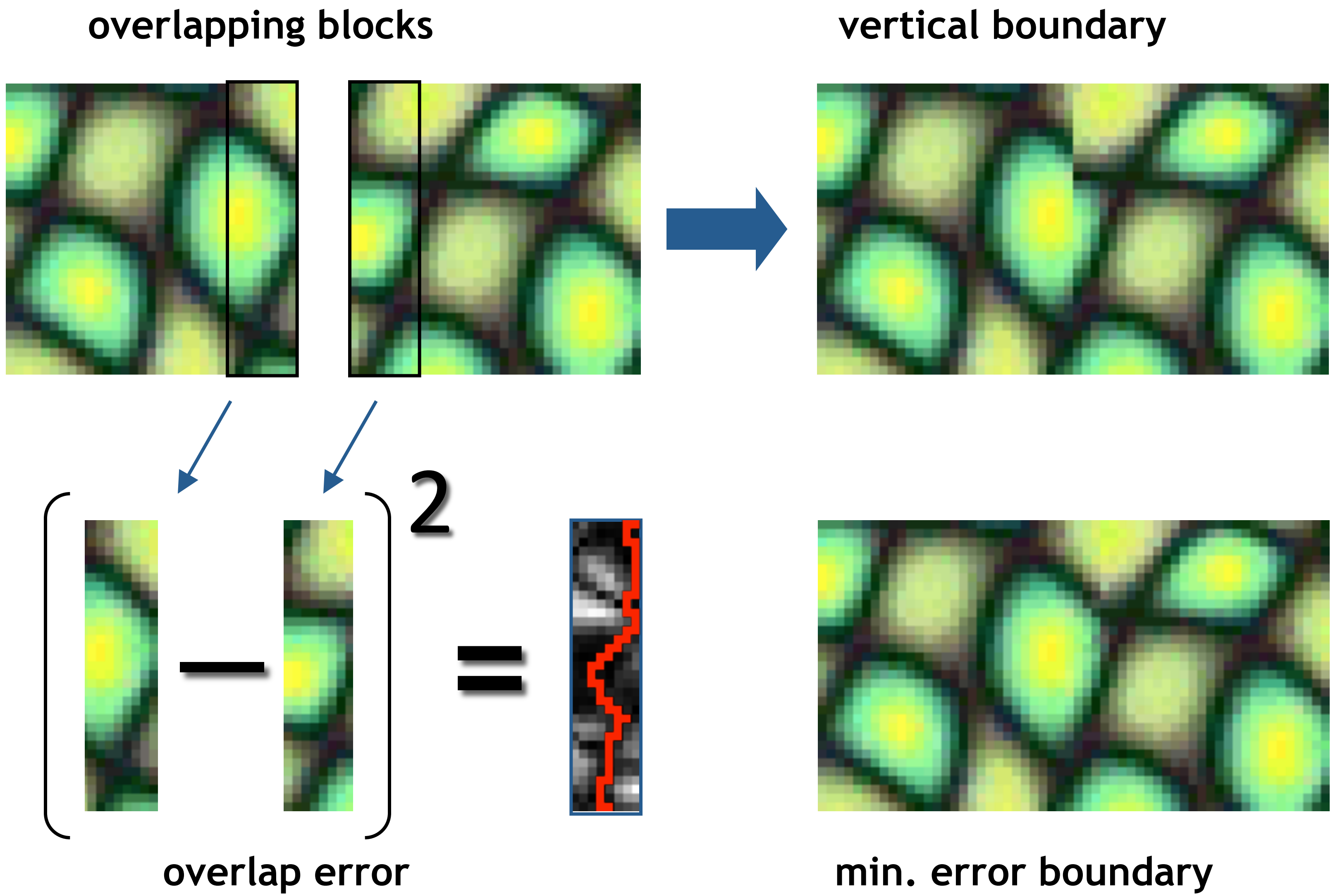


Minimal error boundary cut

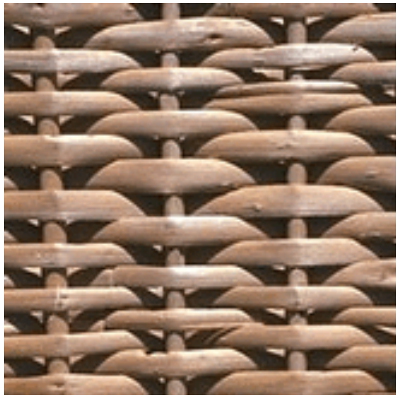


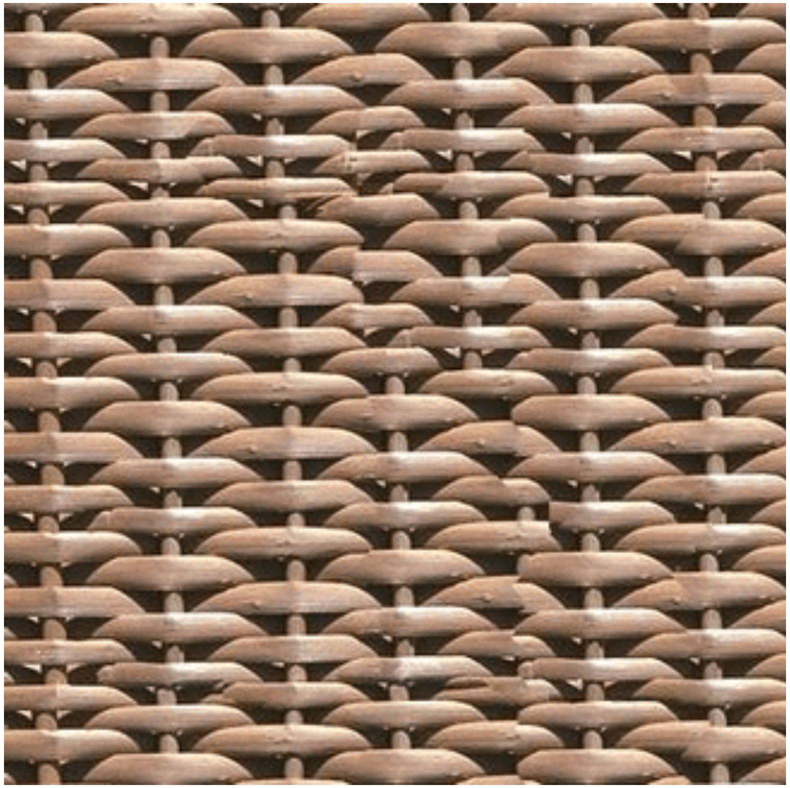
Slide credit A. Efros

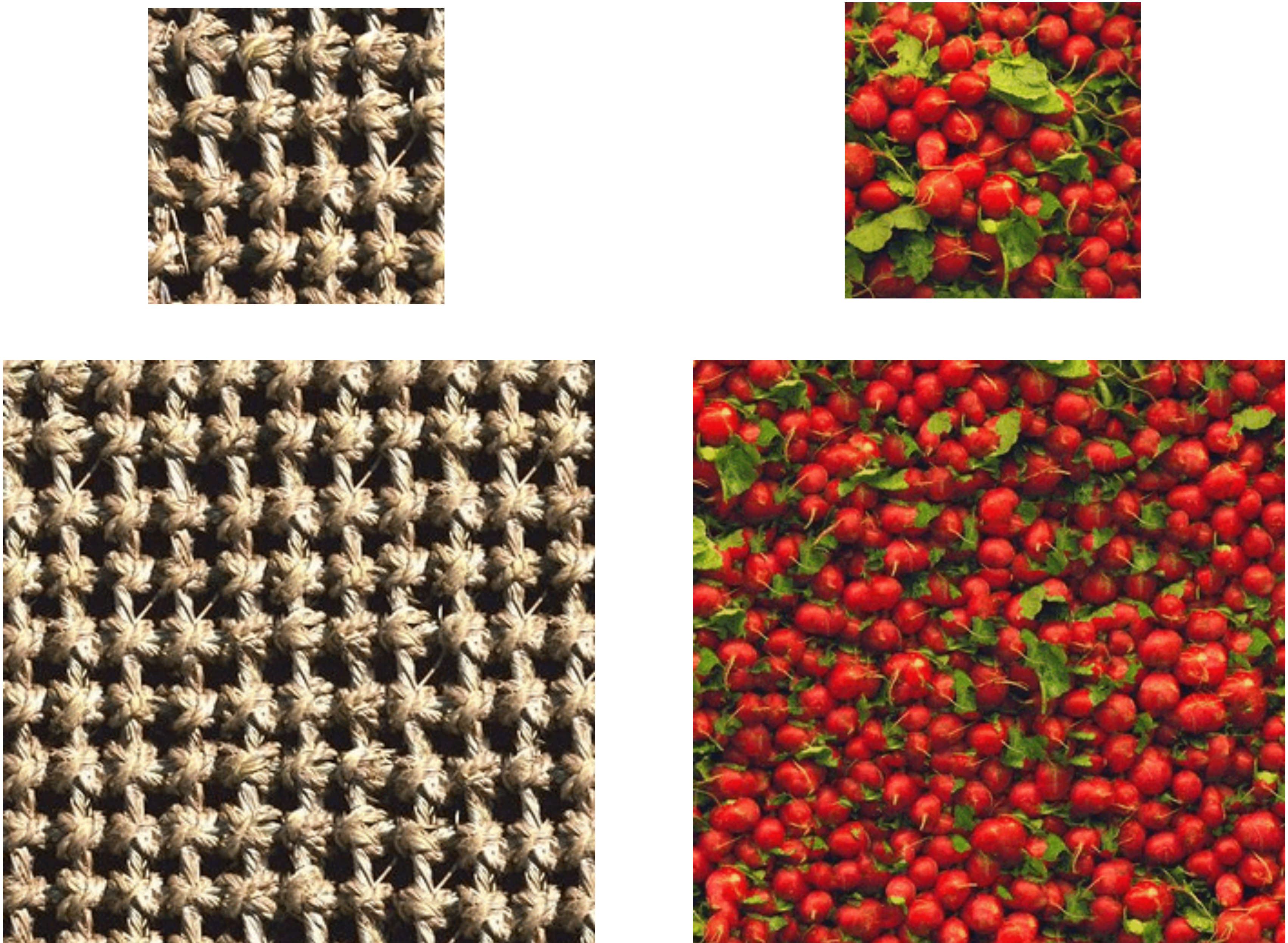
Minimal error boundary



Slide credit A. Efros






College Inn
Chicken Broth

Failures

Texture Transfer

- Take the texture from one object and “paint” it onto another object
  - This requires separating texture and shape
  - That’s HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading

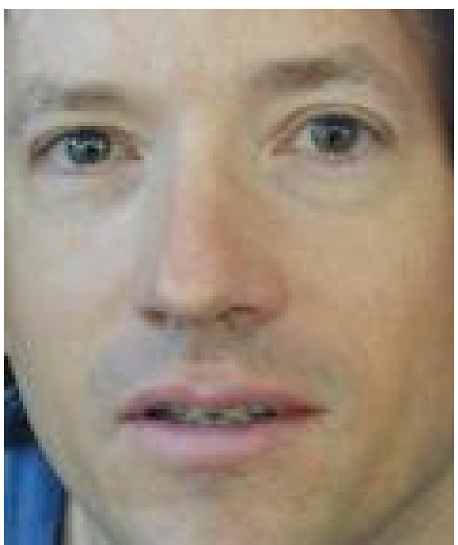
Then, just add another constraint when sampling: similarity to underlying image at that spot

Slide credit A. Efros

+

parmesan

=

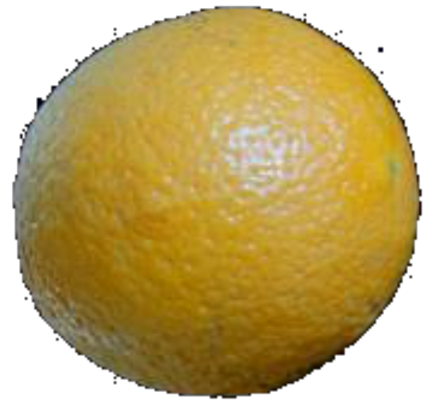


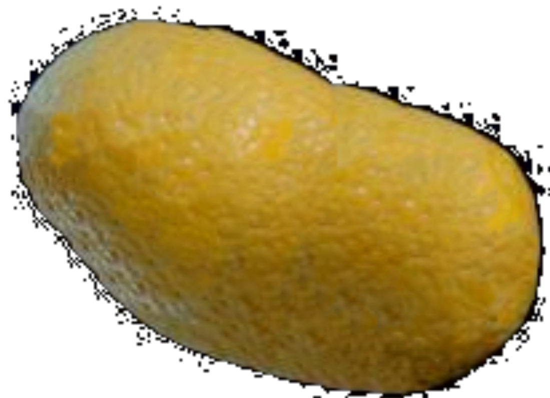
+

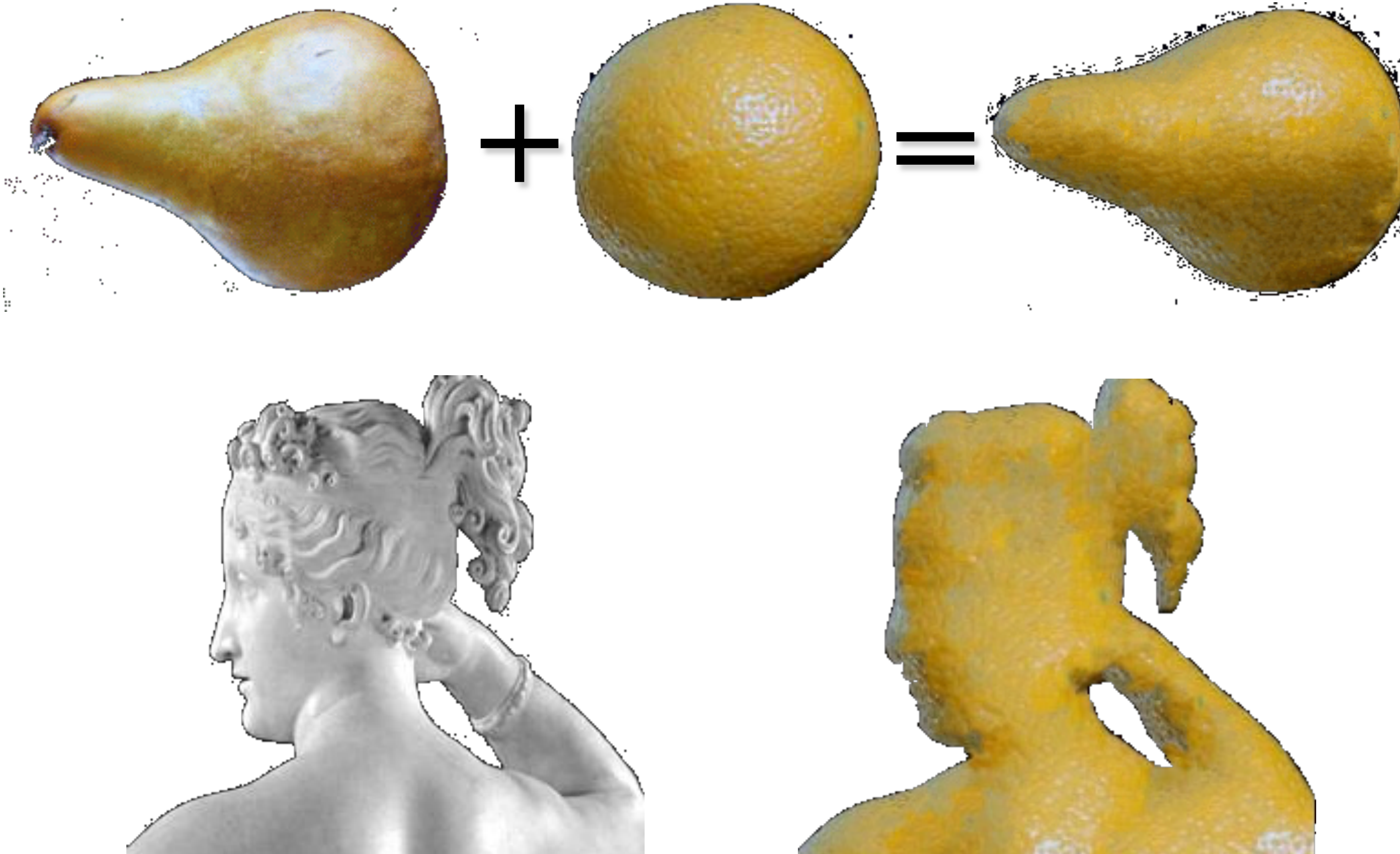
rice

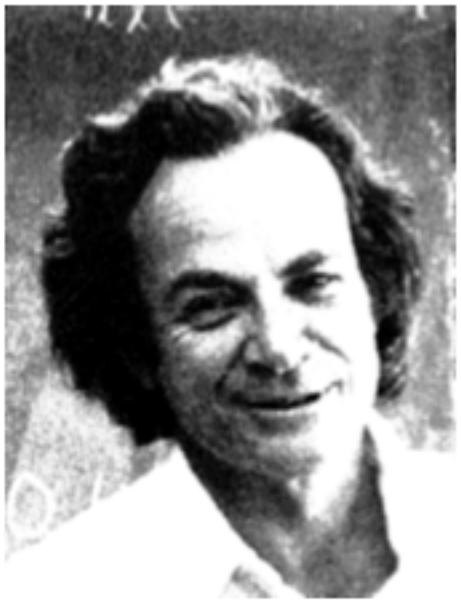
=

Slide credit A. Efros

 +  =







+

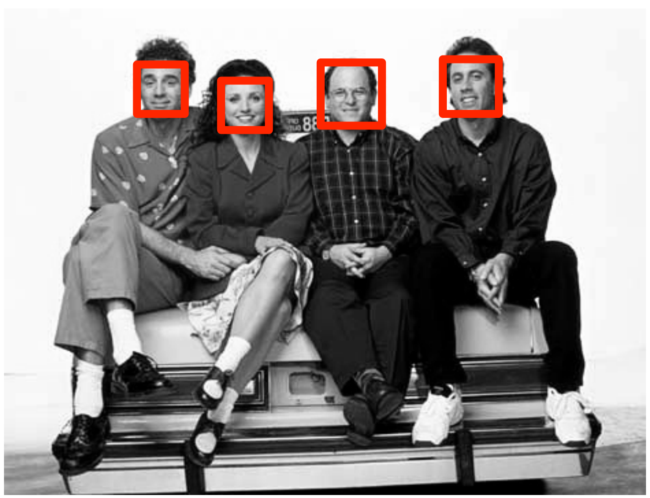
=

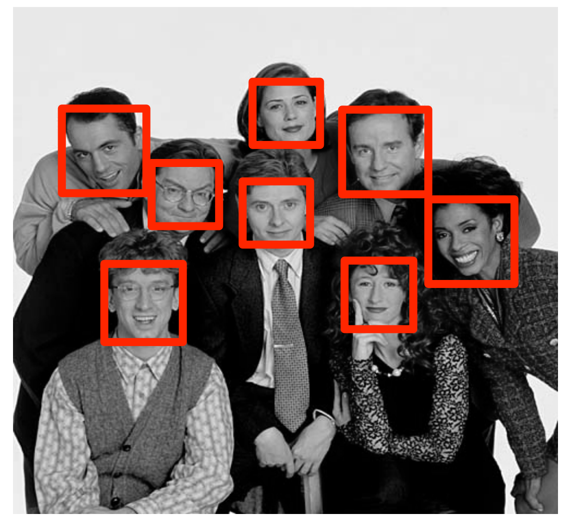
Face detection using sliding window and histograms of oriented gradients

Where are the faces in an image?

Face detection using sliding window and histograms of oriented gradients

Where are the faces in an image?

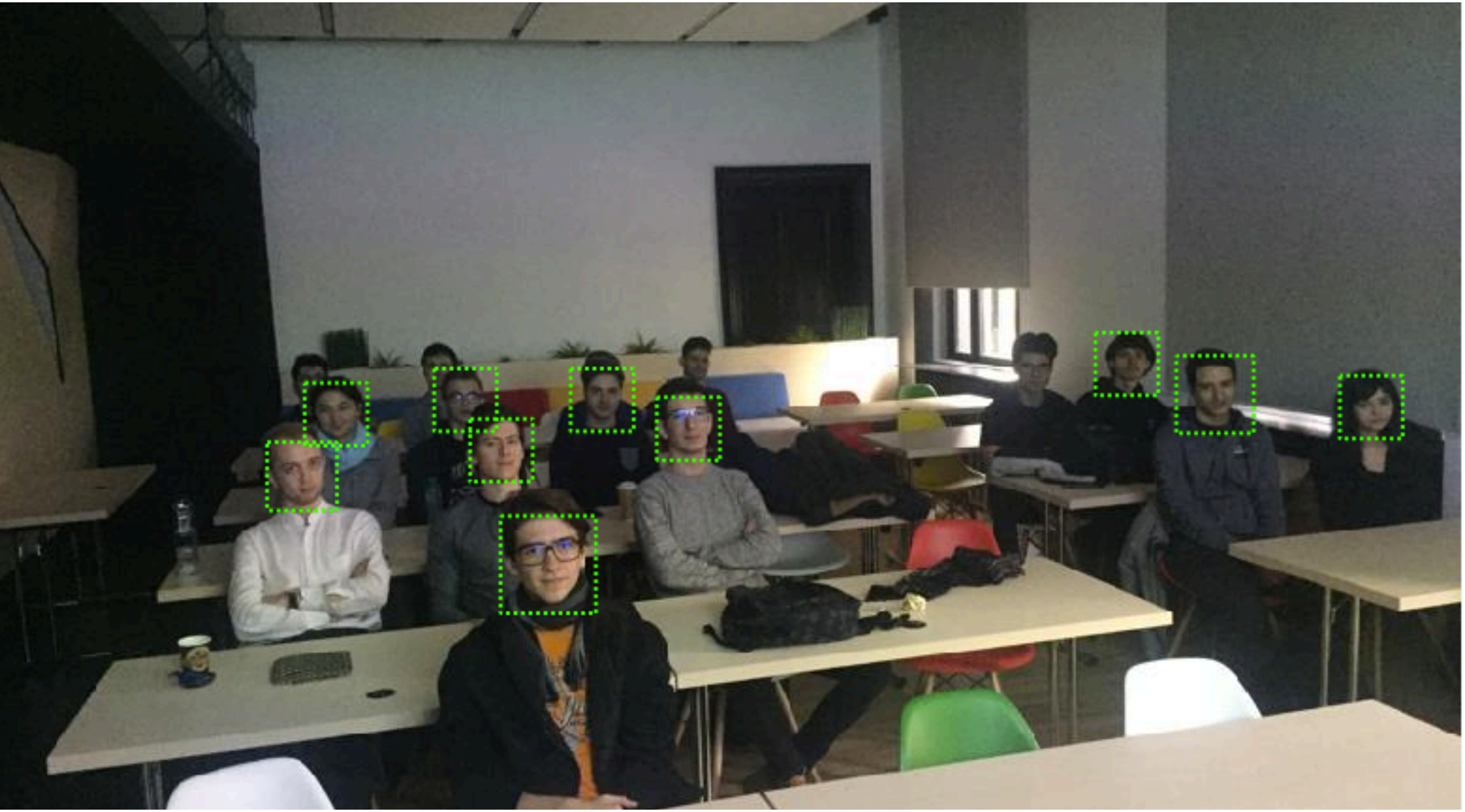


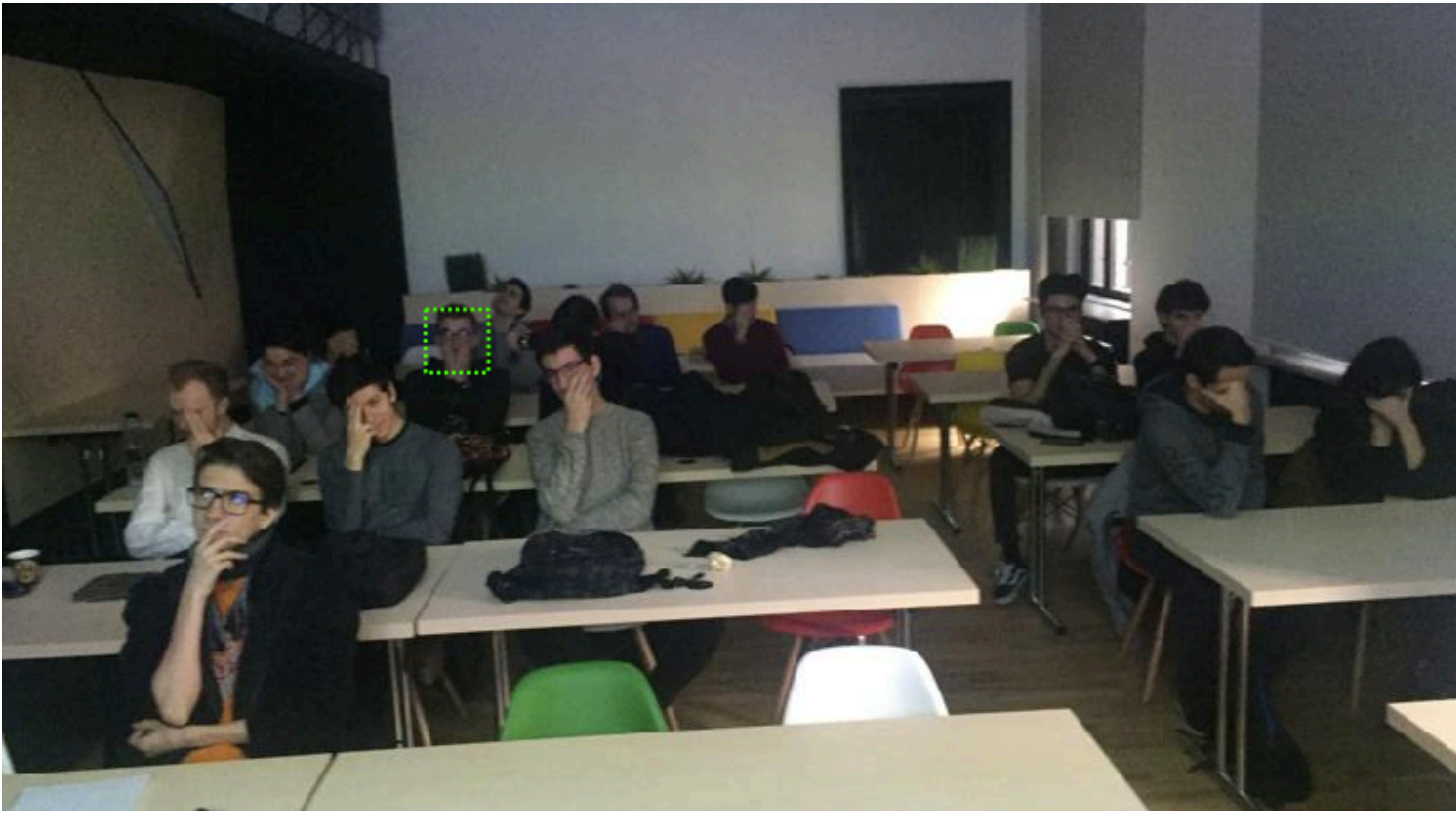


Localization done at the level of a bounding box.

Object detection using sliding window

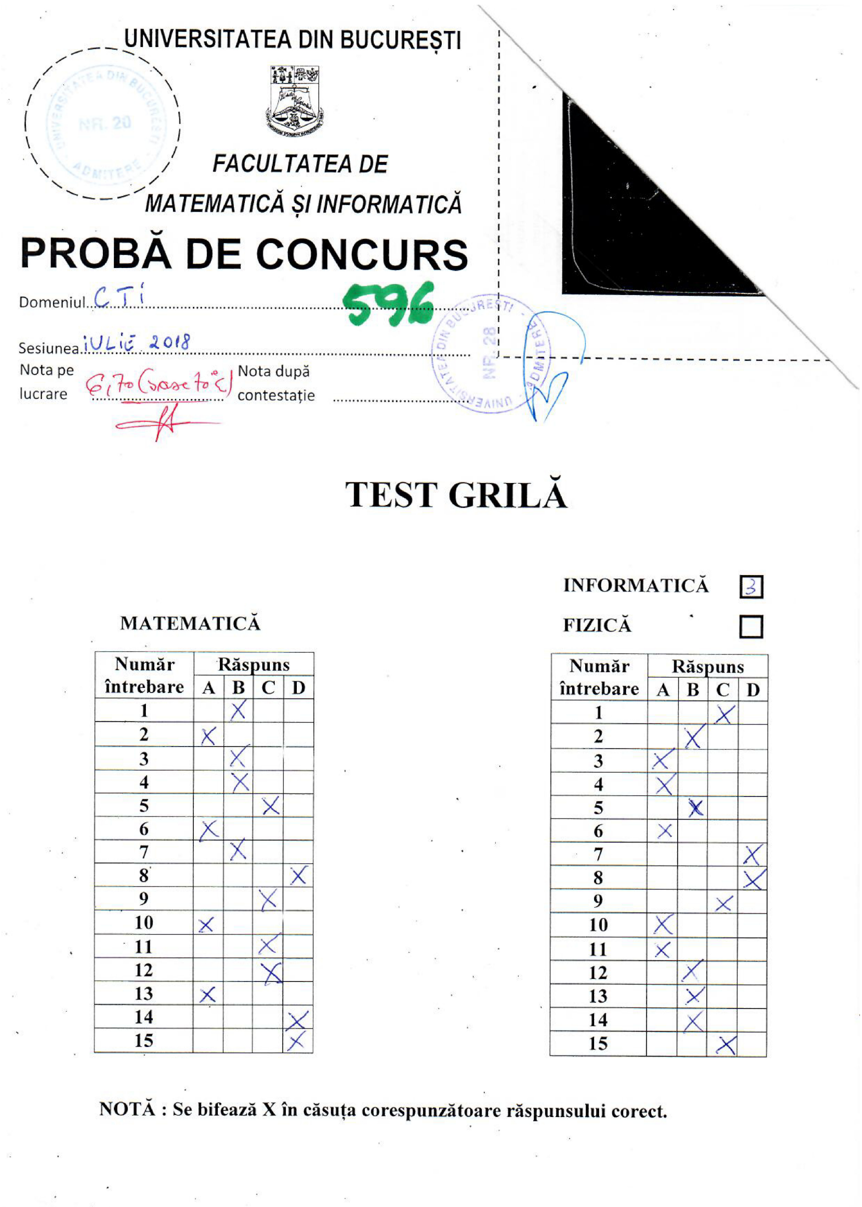
- detection via classification: binary classifier for face → does a window contain a face?
- consider windows positioned at each pixel, for different sizes (to achieve scale invariance)





Computer Vision projects for master students: Project 1, 2 and 3 proposed last year

Automatic grading of multiple choice tests



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROBĂ DE CONCURS

Domeniul CTI 596

Sesiunea iulie 2018

Nota pe lucrare 6,70 (șase 70 c) Nota după contestație

TEST GRILĂ

INFORMATICĂ 3

FIZICĂ

MATEMATICĂ

Număr întrebare	Răspuns A	B	C	D
1		X		
2	X			
3		X		
4		X		
5			X	
6	X			
7		X		
8				X
9			X	
10	X			
11			X	
12			X	
13	X			
14				X
15				X

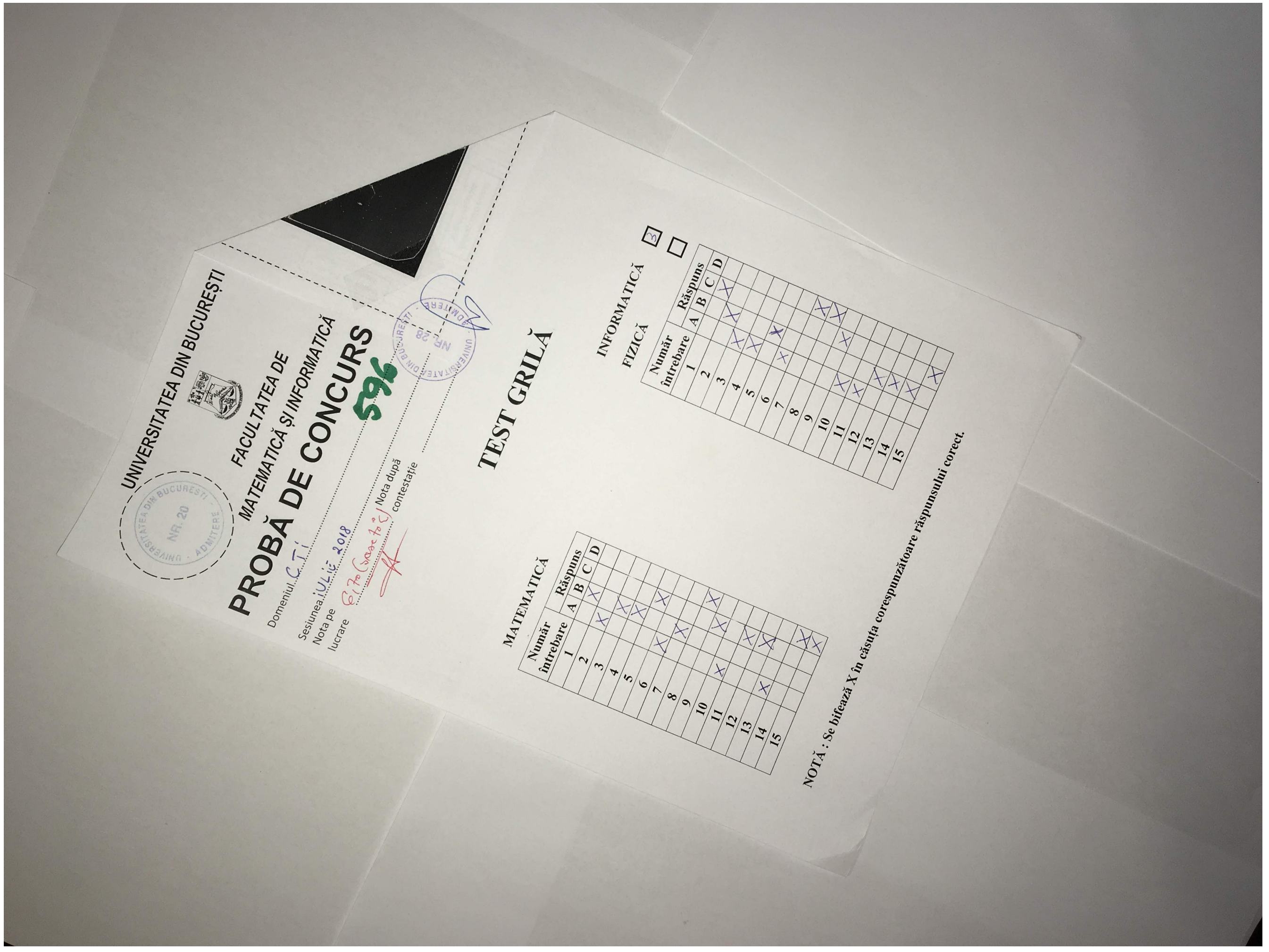
Număr întrebare	Răspuns A	B	C	D
1			X	
2		X		
3	X			
4	X			
5		X		
6	X			
7				X
8				X
9			X	
10	X			
11	X			
12		X		
13		X		
14		X		
15			X	

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.

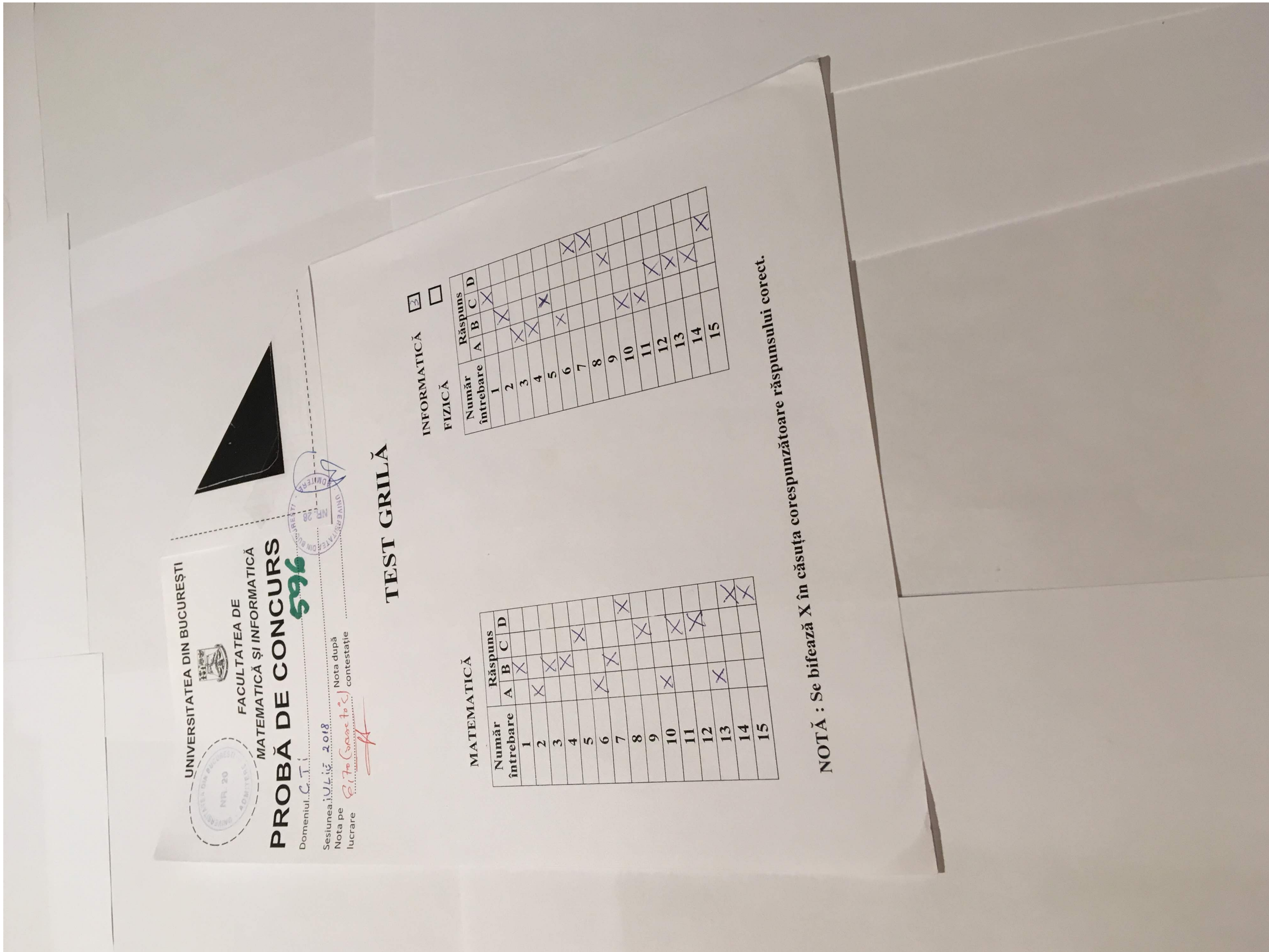
image_2.txt

```
I 3
1 B
2 A
3 B
4 B
5 C
6 A
7 B
8 D
9 C
10 A
11 C
12 C
13 A
14 D
15 D
16 C
17 B
18 A
19 A
20 B
21 A
22 D
23 D
24 C
25 A
26 A
27 B
28 B
29 B
30 C
R 19
```

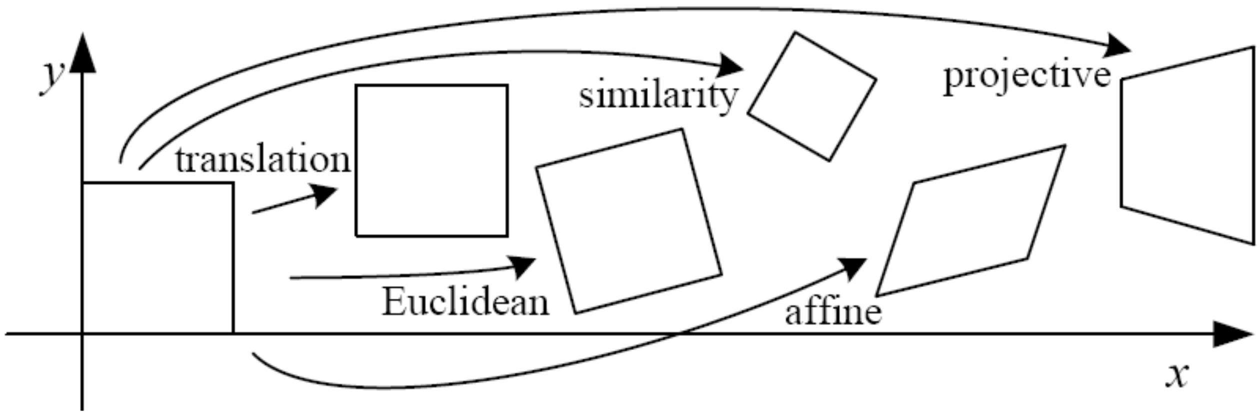
Rotation + scaling



Perspective

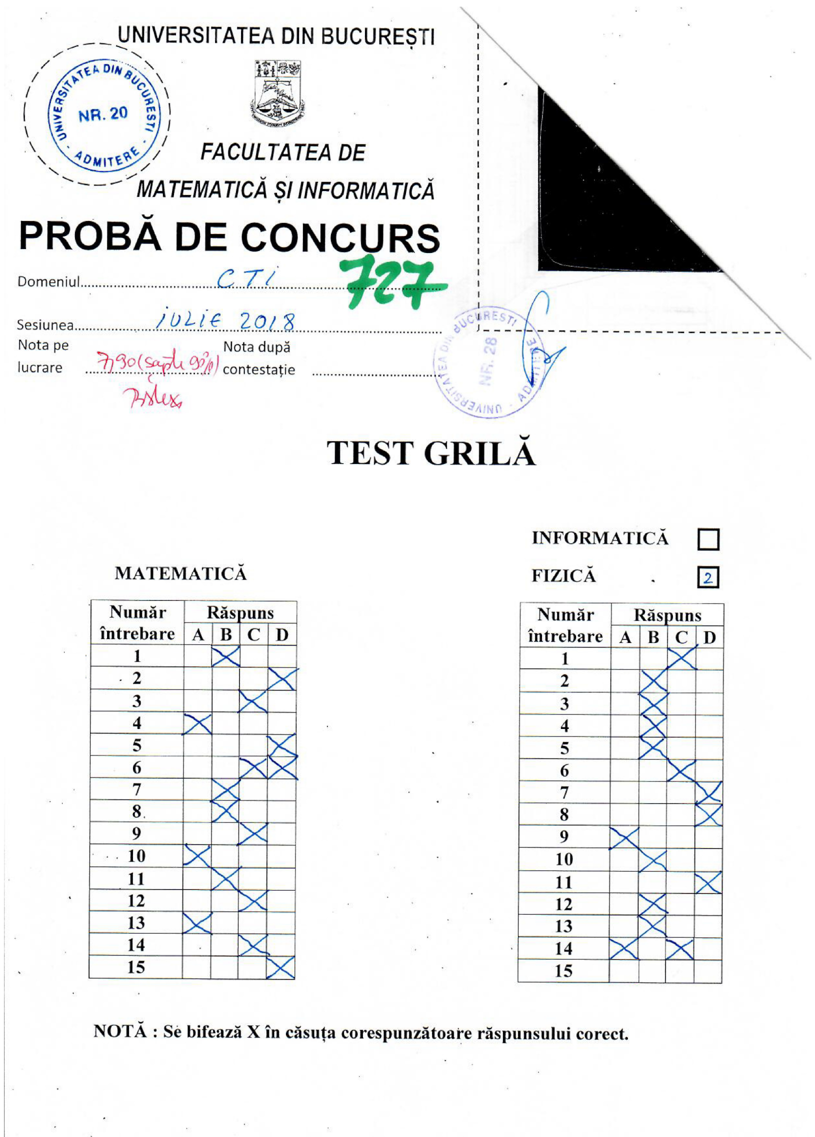


2D Geometric transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

BONUS: handwritten grade recognition



UNIVERSITATEA DIN BUCUREȘTI

UNIVERSITATEA DIN BUCUREȘTI NR. 20 ADMITERE

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

PROBĂ DE CONCURS

727

Domeniul ..... CTI

Sesiunea ..... IULIE 2018

Nota pe lucrare ..... 7,90 (șapte 90/100) Nota după contestație .....

UNIVERSITATEA DIN BUCUREȘTI NR. 28 ADMITERE

TEST GRILĂ

INFORMATICĂ ☐

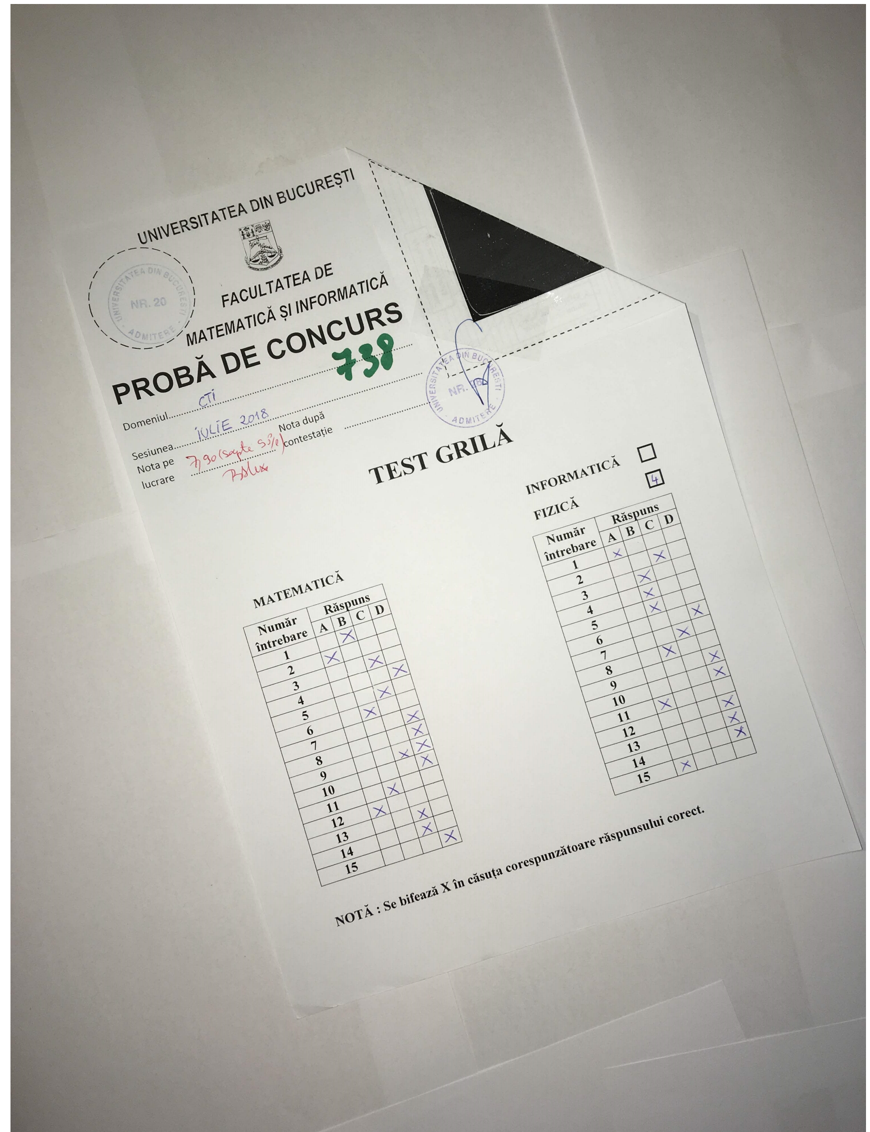
FIZICĂ ☒ 2

MATEMATICĂ

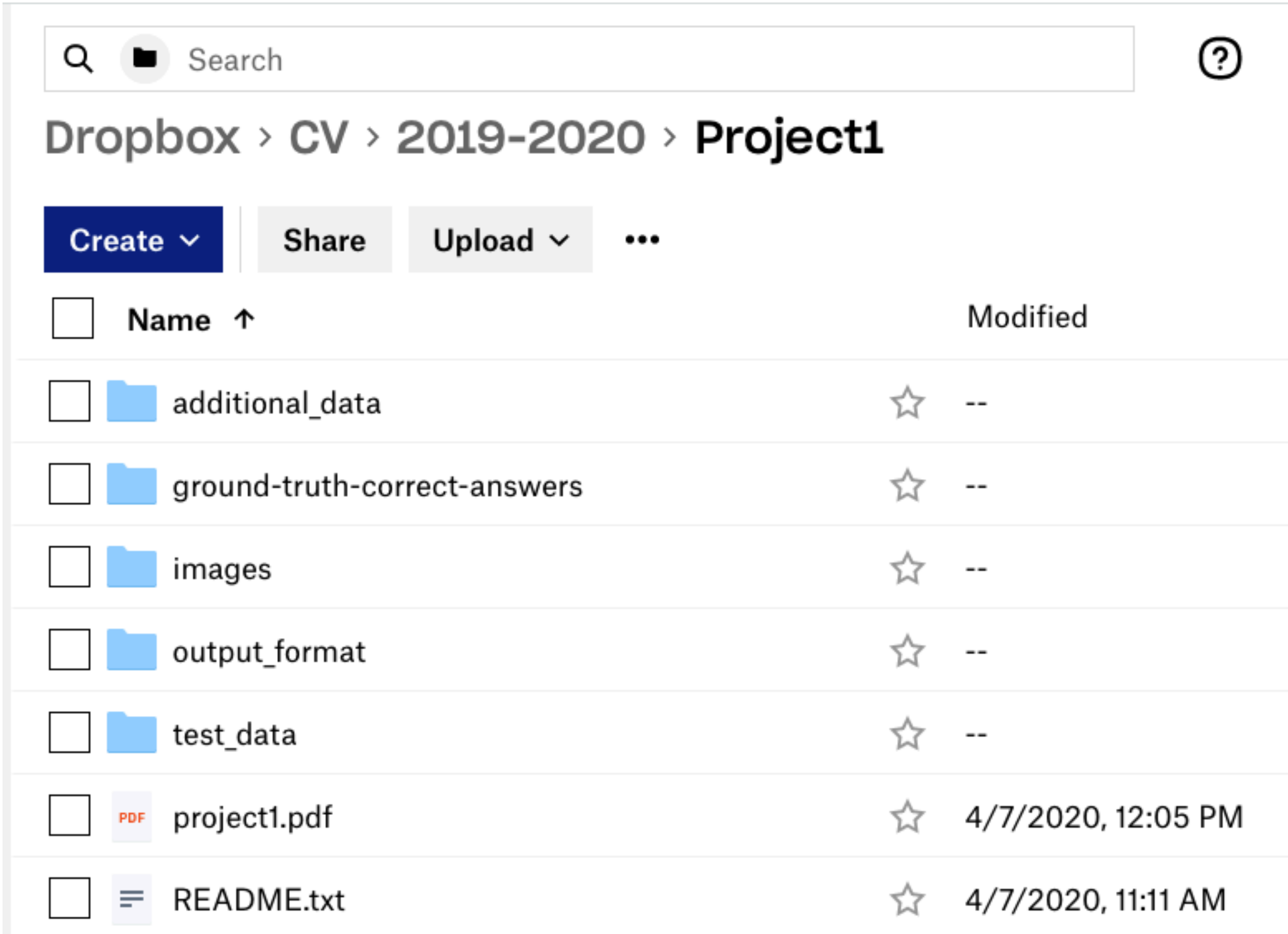
Număr întrebare	A	B	C	D
1		X		
2				X
3			X	
4	X			
5				X
6			X	X
7		X		
8		X		
9			X	
10	X			
11		X		
12			X	
13	X			
14			X	
15				X

Număr întrebare	A	B	C	D
1			X	
2		X		
3		X		
4		X		
5		X		
6			X	
7				X
8				X
9	X			
10		X		
11				X
12		X		
13		X		
14	X		X	
15				

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.



Project 1: tinyurl.com/CV-2020-project1



Project 2
Video analysis of a snooker footage

This project worths 3.5 points but you can gain a bonus for a total of 4.5 points.

- **Task 1** - you receive an image of the snooker table seen from above. The task is to count the number of balls on the table and specify their color;
- **Task 2** - you receive a video containing a player making a shot. In this video the snooker table is seen from above. The task is to decide whether a ball was poted into a pocket and if so recognize the color of the poted ball and the pocket (one of the six pockets) where the ball was poted;
- **Task 3** - you receive a video containing a player making a shot. In this video the snooker table is seen from above. The task is to track the cue ball and another specified ball.
- (bonus) **Task 4** - you receive a much longer video than the previous ones (from Task 2 and Task 3). In this scenario, it may happen that several frames contain the snooker table seen from other viewpoints than above. The task here is output the score for the current footage based on the balls poted by each player. **1 point**

Task 1 – training data

Image 1

annotation

1.txt

```
10
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
3 red
```

Task 1 – training data

Image 8

annotation

8.txt

```
10
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
3 red
```

Task 1 – training data

Image 18

annotation

18.txt

```
21
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
14 red
```

Task 1 – training data

Image 23

annotation

23.txt

```
16
1 white
0 black
1 pink
1 blue
1 green
1 brown
1 yellow
10 red
```

Task 1 – training data

Image 49

annotation

49.txt

```
21
1 white
1 black
1 pink
1 blue
1 green
1 brown
1 yellow
14 red
```

Project 2 – Task 2

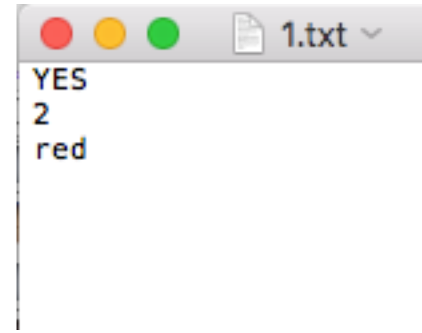
- **Task 2** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to decide whether

a ball was potted into a pocket and if so recognize the color of the potted ball and the pocket (one of the six pockets) where the ball was potted. By solving correctly this task for a video you will earn 0.04 points as follows: if the correct answer is NO (no ball was poted) you will receive 0.04 points if your output is NO, else you will receive 0.02 points if your output is YES, 0.01 points for specifying the correct pocket (1 - top left corner, 2 - top right corner, 3 - bottom left corner, 4 - bottom right corner, 5 - middle left corner, 6 - middle right corner) and 0.01 points for specifying the correct color of the potted ball. In each of the 25 videos there will be a maximum of one ball being potted. For a test video your output should be similar with the one provided in the directory `training_data/Task2/`. **(1 point)**

Task 2 – training data

Video 1

annotation



Task 2 – training data

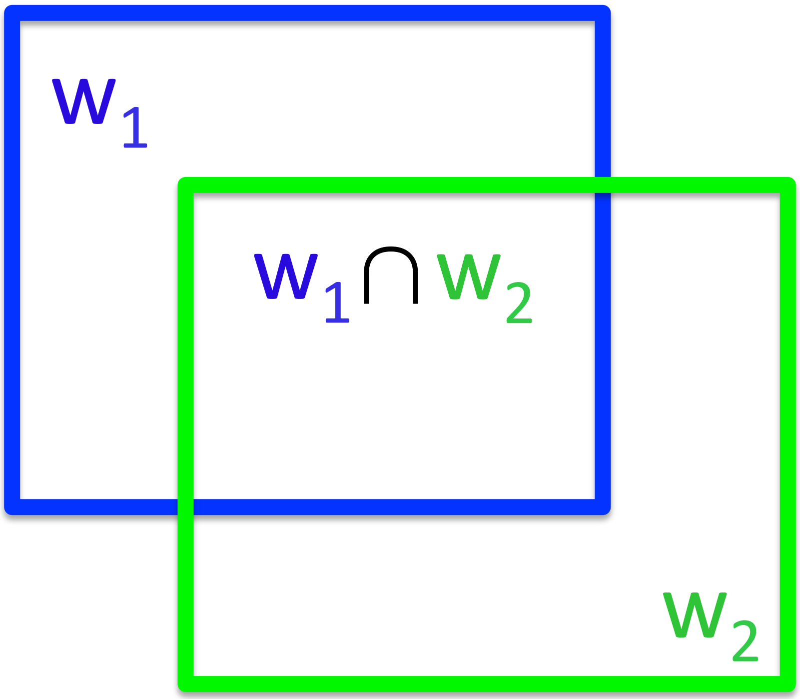
Video 2

annotation

Project 2 – Task 3

- **Task 3** - you receive a test set of 25 videos containing a player making a shot. In each of the 25 videos the snooker table is seen from above. The task is to track the cue ball (the white ball) and another specified ball. The initial bounding boxes of the two balls to be tracked are provided for the first frame. By correctly solving the tracking problem for a video you will earn 0.04 points (0.02 points for each correctly tracked ball). In each video we will consider that your algorithm *correctly tracks a ball* if in more (greater or equal) than 80% of the frames your algorithm *correctly localizes the ball to be tracked*. We consider that your algorithm correctly localizes the ball to be tracked in a specific frame if the value of the IOU (intersection over union) beetween the window provided by your algorithm and the ground-truth window is more than 20%. For a test video your output should be similar with the one provided in the folder `training_data/Task3/`. **(1 point)**

Intersection over union



$$IOU(w_1, w_2) = \frac{area(w_1 \cap w_2)}{area(w_1 \cup w_2)}$$

$$IOU(w_1, w_2) = \frac{area(w_1 \cap w_2)}{area(w_1) + area(w_2) - area(w_1 \cap w_2)}$$

Task 3 – training data

Video 1

annotation

1_white.txt

```
196 -1 -1 -1 -1
0 859 519 890 550
1 863 522 888 550
2 859 520 889 544
3 861 522 890 549
4 860 521 886 547
5 860 521 888 547
6 858 518 893 549
7 862 520 888 546
8 860 520 887 547
9 861 521 889 548
10 860 521 887 547
```

```
178 852 98 873 120
179 853 99 874 121
180 853 99 874 121
181 854 99 875 121
182 854 99 875 121
183 855 99 876 121
184 855 99 876 121
185 855 99 876 121
186 856 99 877 121
187 856 99 877 121
188 855 99 876 121
189 856 99 877 121
190 856 99 877 121
```

Task 3 – training data

Video 2

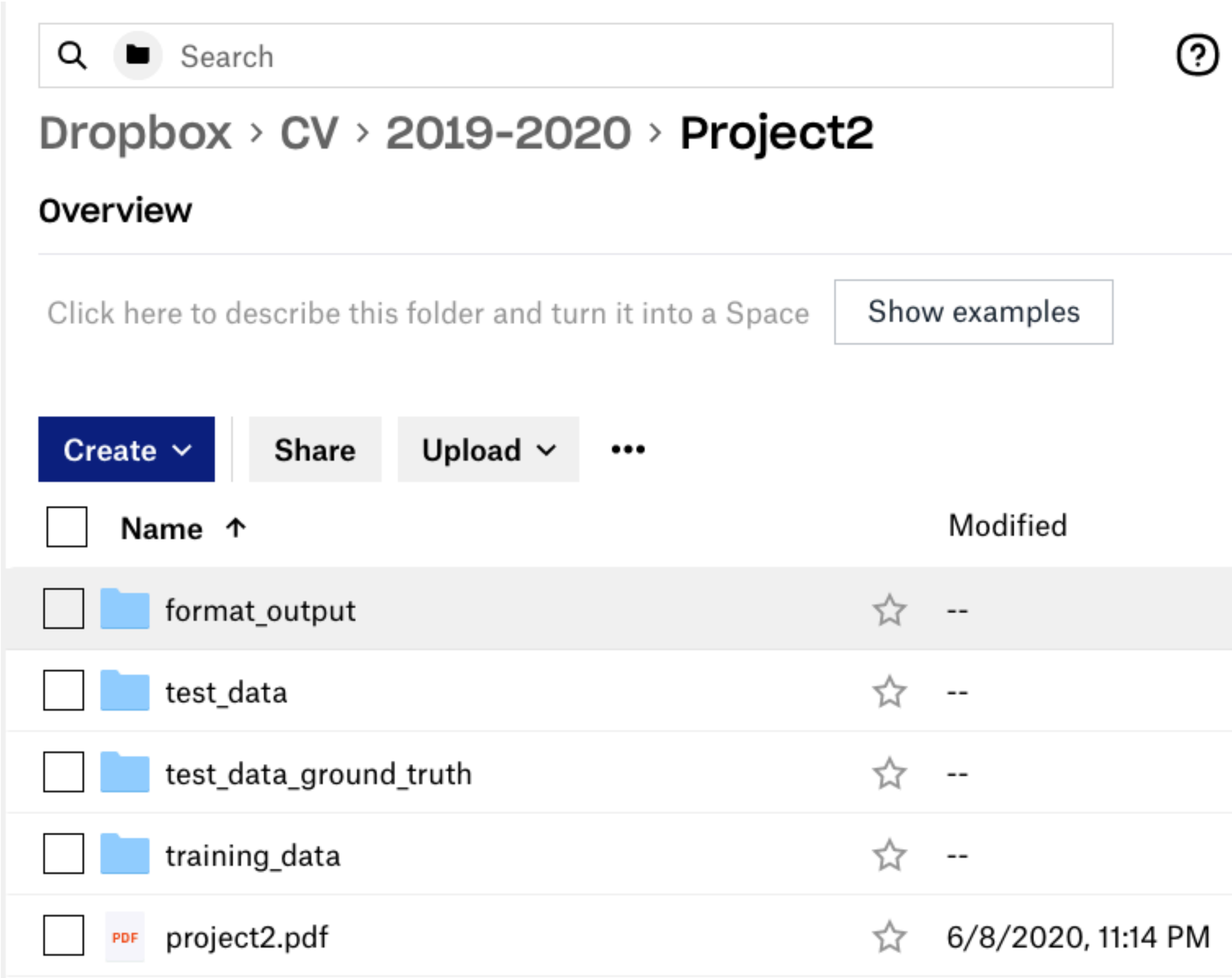
Project 2 – Task 4

- (bonus) **Task 4** - - you receive a test set of 25 videos containing a player making a shot. Different than the previous tasks, in this videos the snooker table can be filmed from different viewpoints (not only from above). The task is to track the cue ball. The task here is much harder as you have diferent scales of the cue ball, changes in camera viewpoint, the initial bounding box of the white ball is not provided. By correctly solving the tracking problem for a video you will earn 0.04 points. The rules described at Task 3 apply here, meaning that your algorithm should correctly localizes the cue ball in more than 80% of the frames, at each frame correctly localization means that the window provided by your algorithm should have IOU more than 20% with respect to the ground-truth window. For a test video your output should be similar with the one provided in the folder `training_data/Task4/` **(1 point)**

Task 4 – training data

Task 4 – training data

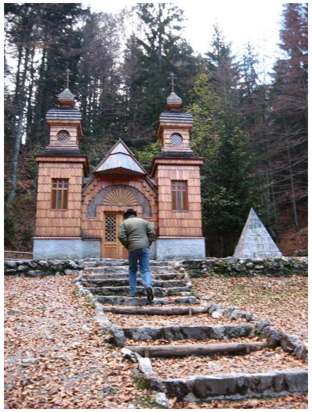
Project 2: tinyurl.com/CV-2020-project2

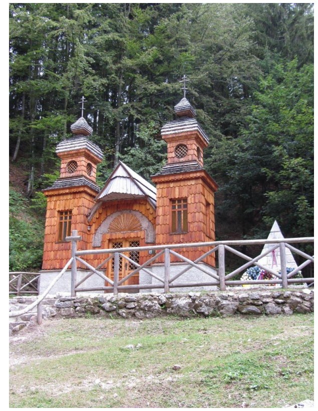
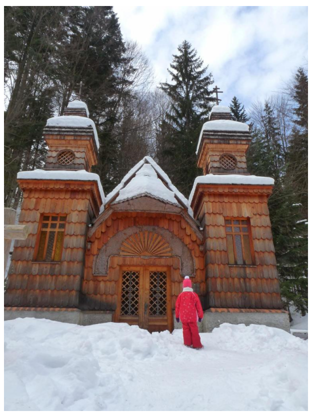


Project 3 – Image retrieval using bag of visual words model

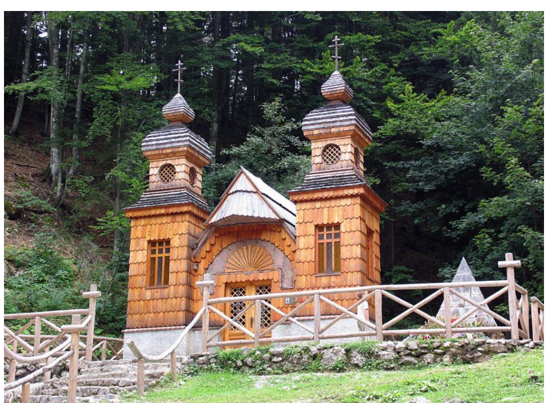
- build a database with 50 classes
- each class contains a small number (e. g. 10) images with the same scene (landmark)
- the goal: design an image retrieval system that takes as input a query image and outputs most N (N > 10) similar in content images from the database
- ingredients:
  - bag-of-visual-word model
  - SIFT descriptors
  - geometric verification with RANSAC
  - inverted file index – use TF-IDF

Class 1







Class 2










Class 45


F.C. ST. PAULI
1910


F.C. ST. PAULI


F.C. ST. PAULI
119
118
26


F.C. ST. PAULI
1910

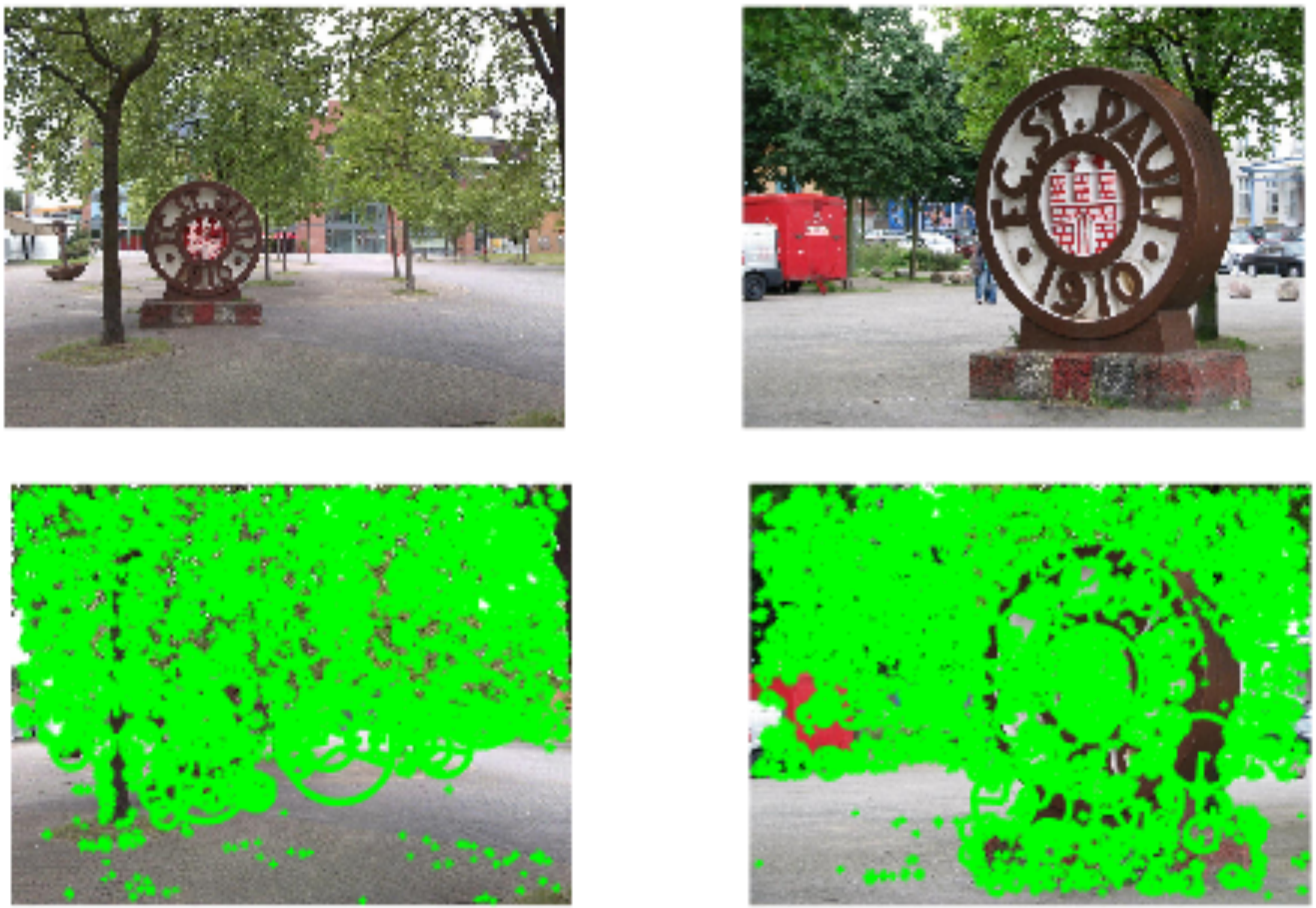

F.C. ST. PAULI
1910


F.C. ST. PAULI
1910

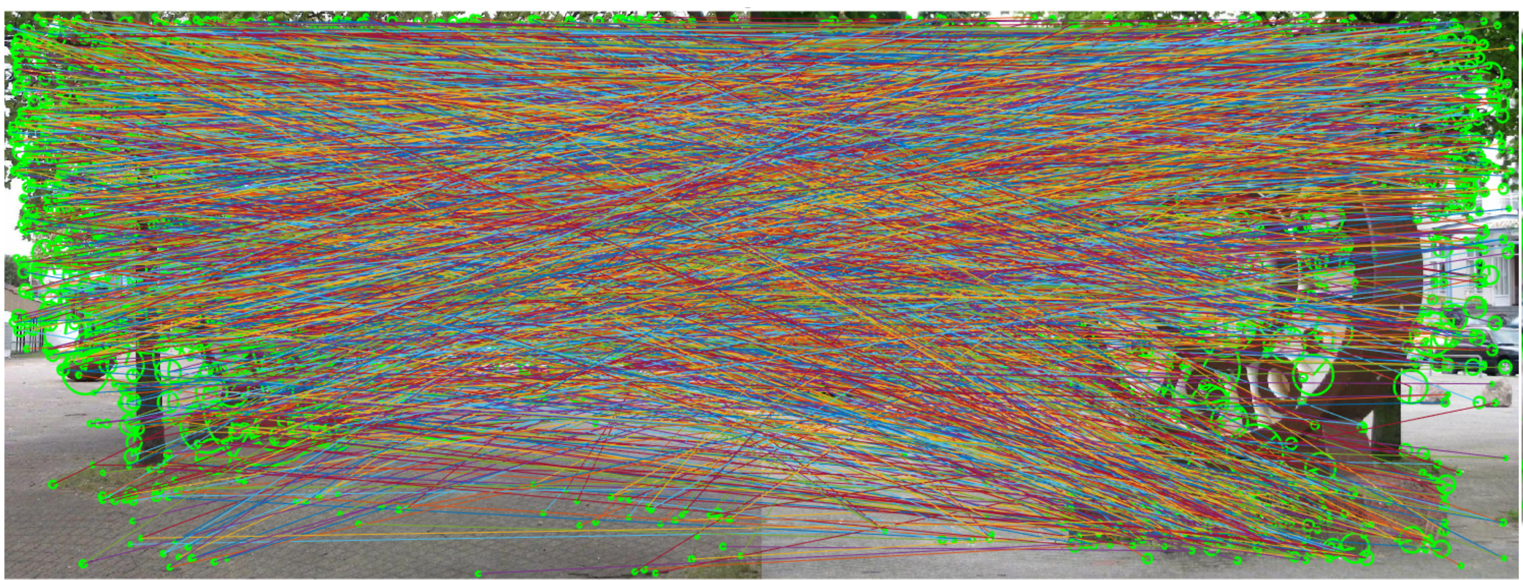

F.C. ST. PAULI
1910


F.C. ST. PAULI
1910

SIFT extraction



SIFT matching using 1^{st}NN



SIFT matching using the ratio $1^{st}NN/2^{nd}NN$

SIFT matching using the ratio 1^{st}NN/2^{nd}NN + geometric verification with RANSAC

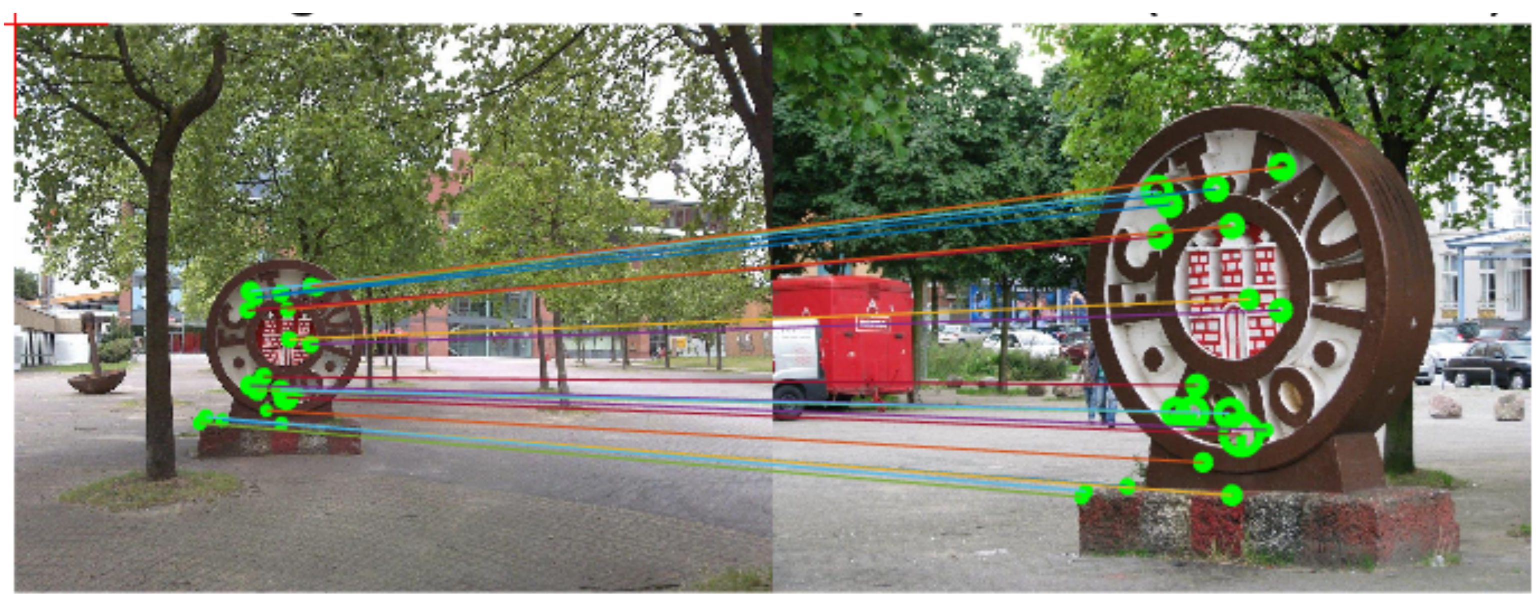


Image retrieval based on inverted file index

Query image

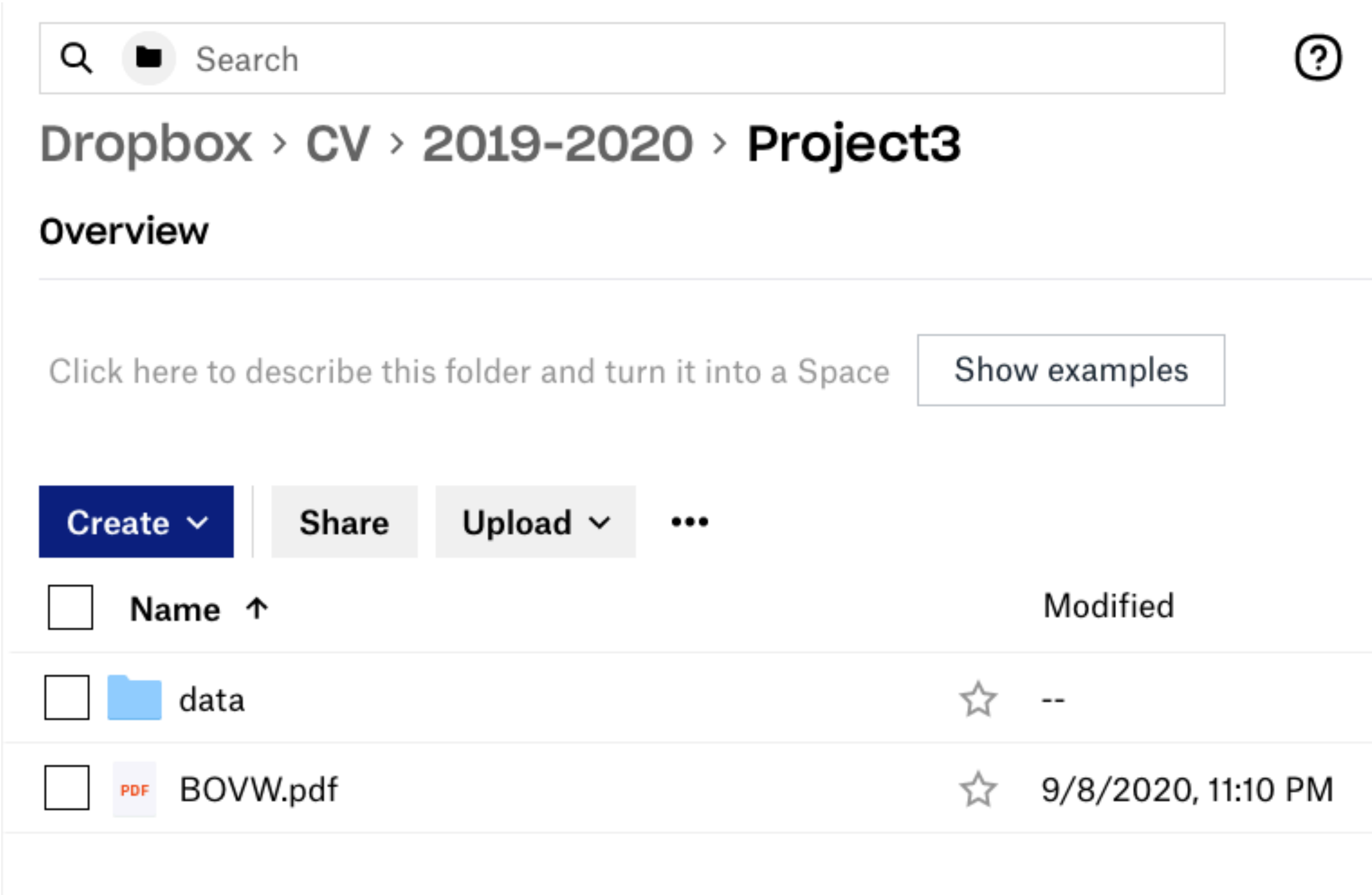
Query image

Image retrieval based on inverted file index + geometric verification

Image retrieval based on inverted file index + geometric verification

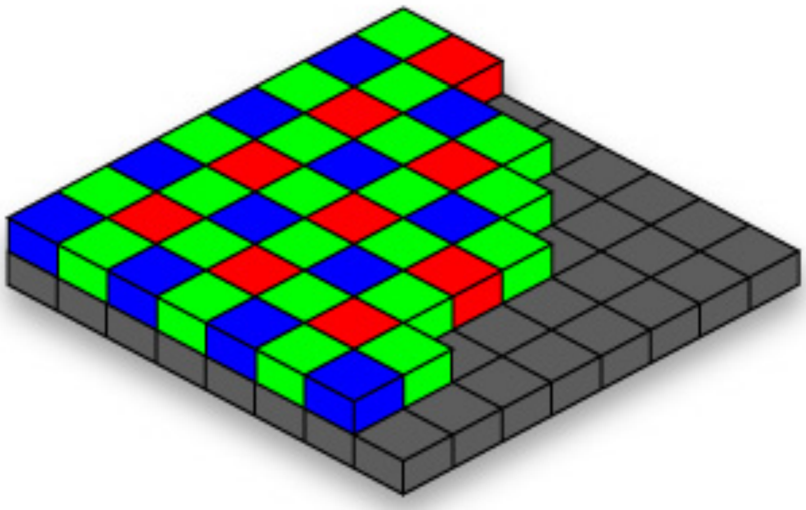
Query image

Project 3: tinyurl.com/CV-2020-project3

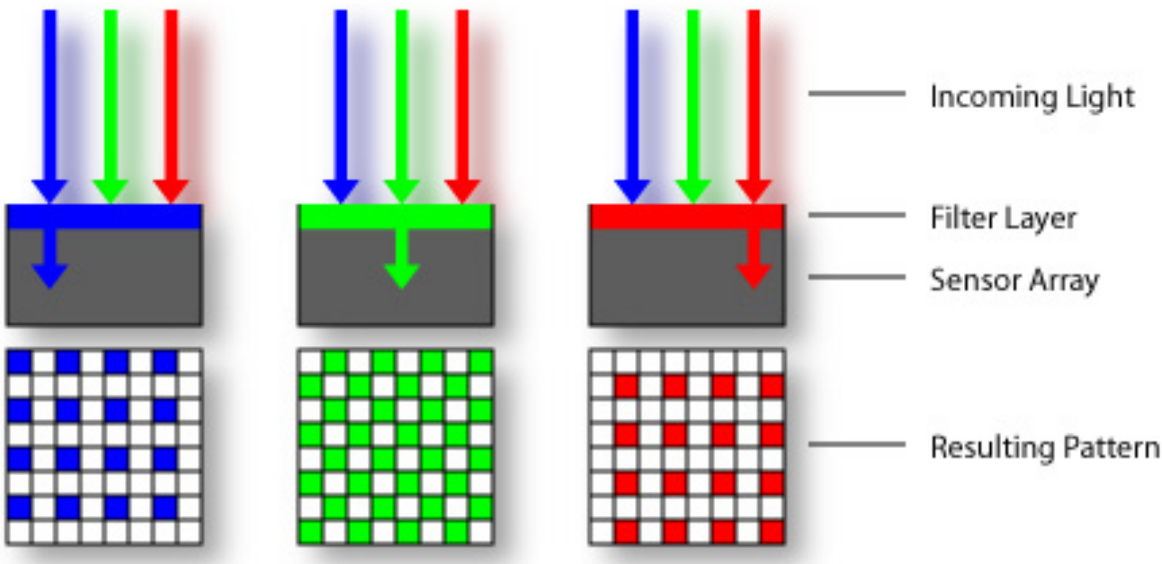


First laboratory class

Color Sensing: Bayer Grid



Estimate RGB at each cell from neighboring values

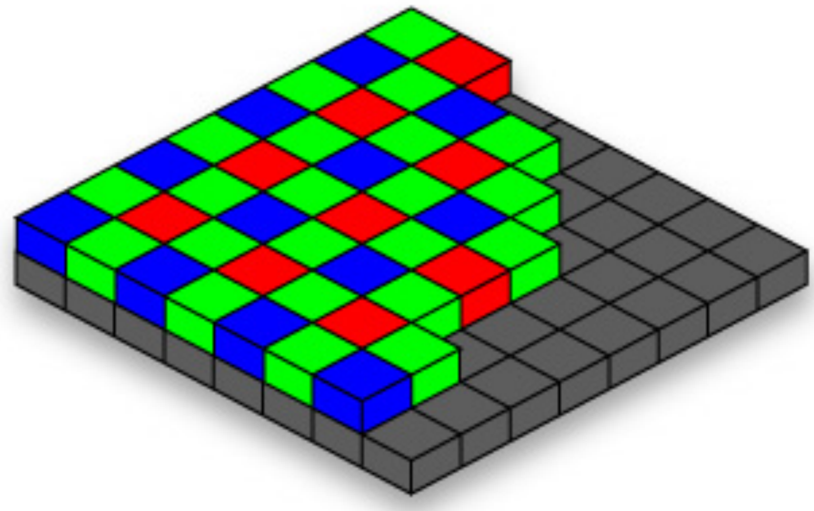


http://en.wikipedia.org/wiki/Bayer_filter

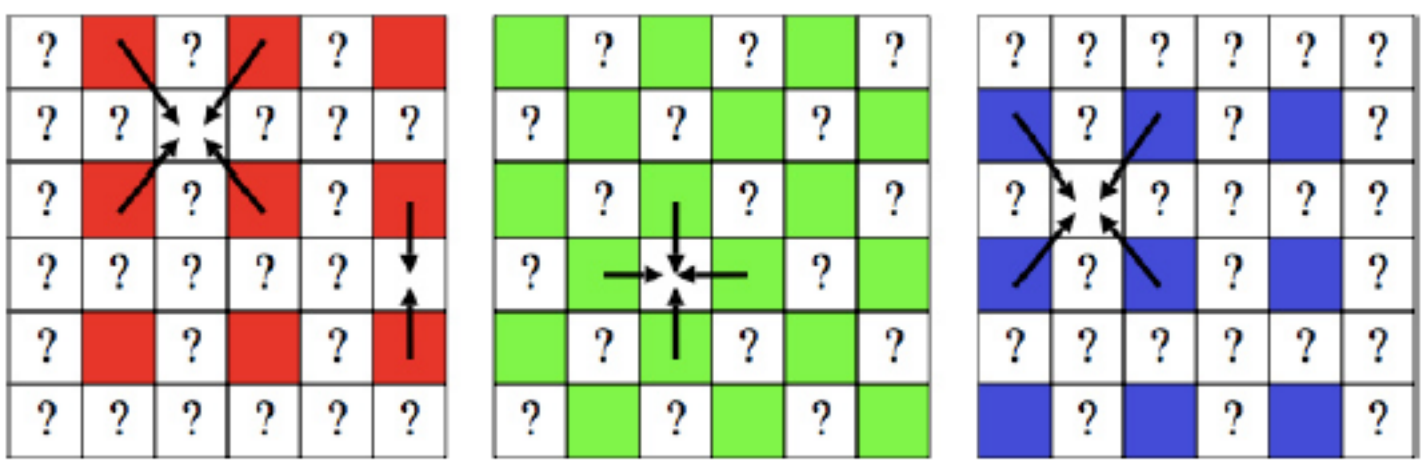
Slide credit Steve Seitz

First laboratory class

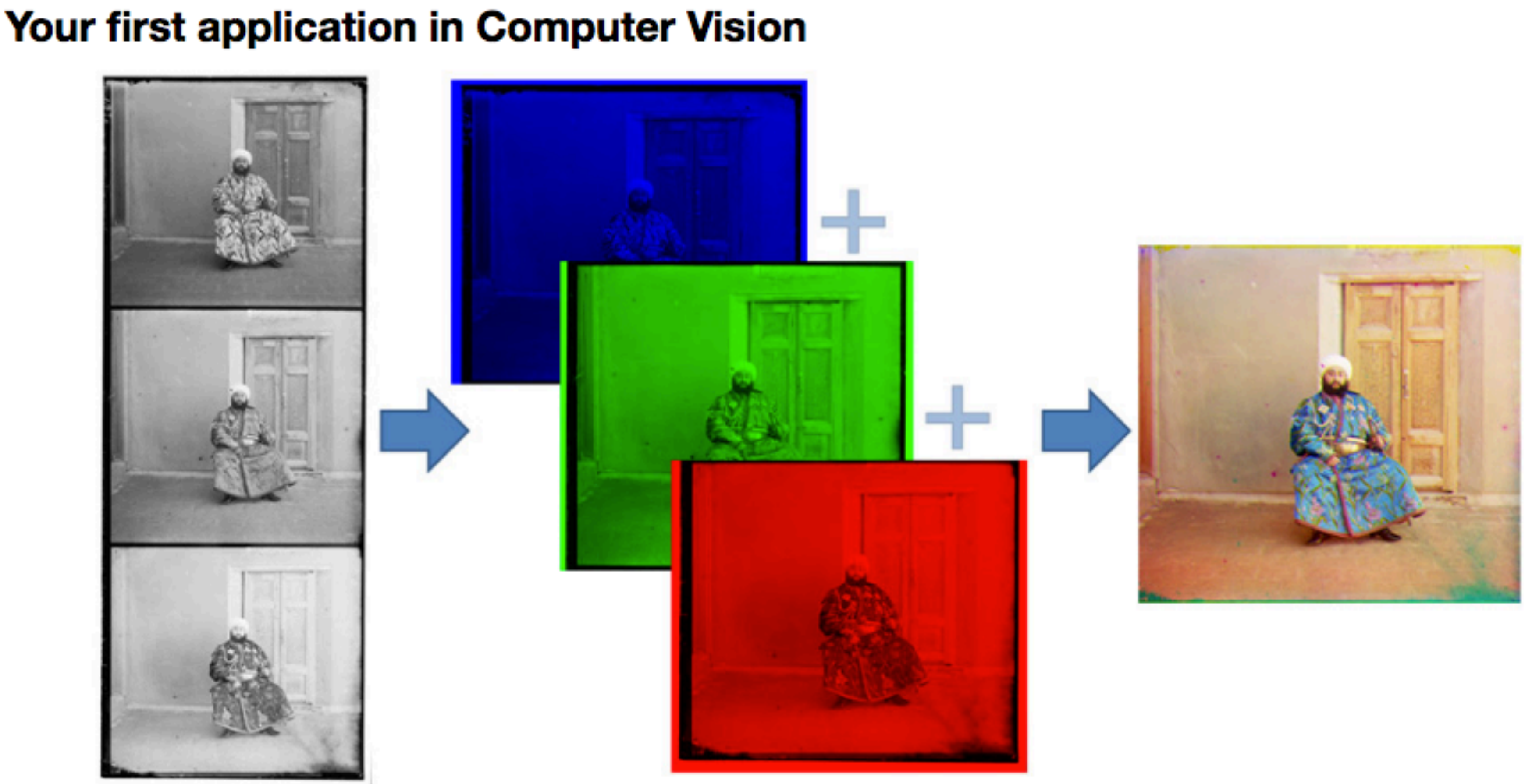
Color Sensing: Bayer Grid



Estimate RGB at each cell from neighboring values

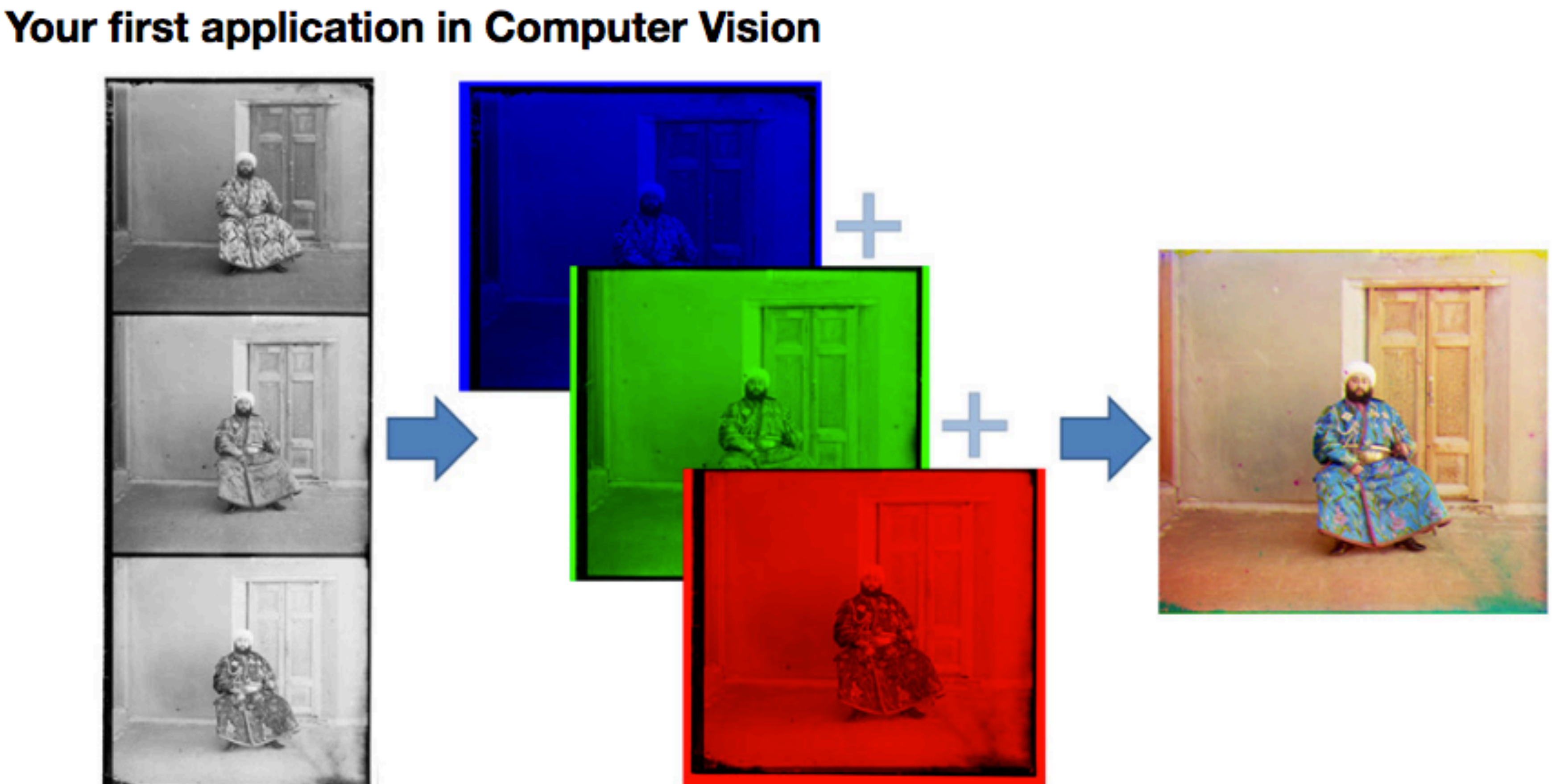


First laboratory class



Record three exposures of every scene onto a glass plate using a red, a green, and a blue filter and then project the monochrome pictures with correctly colored light to reproduce the color image

First laboratory class



Your program will take a **glass plate image** as input and produce a **single color image** as output. The program should *divide the image into three equal parts* and **align the second and the third parts (G and R) to the first (B)**.

Not aligned

Aligned