

1							
2							
3							
4							
5							
6							
7							
8							
9							
0	X						

Registration number

Signature

**Note:**

- Cross your Registration number(with leading zero). It will be evaluated automatically.
- Do not sign the above signature field. A signature field will be provided as part of the first problem.

## Computer Vision II: Multiple View Geometry

**Exam:** IN2228 / Midterm

**Date:** Thursday 16<sup>th</sup> July, 2020

**Examiner:** Florian Bernard

**Time:** 14:00 – 14:20

### Working instructions

- This exam consists of **8 pages** with a total of **6 problems**. Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 25 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - this is an open book graded exercise
- This graded exercise uses the TUMexam platform which offers a student manual on their webpage<sup>1</sup>.
- The boxes on the sides of the subproblems (those with numbers) are used for correction and ticking them is prohibited. A ticked box that is not part of a multiple choice question may result in zero points for the problem.
- In the multiple choice questions there is always exactly one correct answer. A correct answer (comprising of exactly one tick at the correct position) will give the indicated credits, whereas a wrong answer will result in 0 credits.

*Mark correct answers with a cross*



*To undo a cross, completely fill out the answer option*



*To re-mark an option, use a human-readable marking*




- Remark: We aimed to provide a large variety of the type of problems that may occur in the final/re-take exam. **To obtain the grade bonus it is not necessary to attain all credits.** Do not get discouraged if the time is not sufficient to solve all problems. Start with the problems that you can solve easily and then progress to the harder ones.


Left room from \_\_\_\_\_ to \_\_\_\_\_ / Early submission at \_\_\_\_\_

<sup>1</sup>[https://tumexam.de/static/handreichung\\_submissions\\_students.pdf](https://tumexam.de/static/handreichung_submissions_students.pdf)

## Problem 1 Personal Information (0 credits)

- 0  a) Please sign the following by entering your full name: "I hereby assure that I solve and submit this exam myself under my own name by only using the allowed tools listed on the first page".

This field should contain your name.

- 0  b) Please enter your matriculation number with leading zero.

This field should contain your matriculation number.

Sample Solution

## Problem 2 Linear Algebra (5 credits)

The solution to problem a) should be given in MATLAB syntax, e.g.,

$$A = [a11, a12, a13, a14; a21, a22, a23, a24; a31, a32, a33, a34]$$

a) Consider the linear spaces

$$U = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\},$$
$$V = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \\ -1 \end{pmatrix} \right\}.$$

0  
1  
2  
3

Find a matrix  $A \in \mathbb{R}^{3 \times 4}$  such that  $U$  and  $V$  are equal to the range and kernel of  $A$ .  $A =$

There are multiple possible solutions for  $A$ . One valid solution is  
 $A = [0, 1, 0, 2; 0, 0, 0, 0; 1, 0, 1, 0]$ .

b) Let  $B \in \mathbb{R}^{n \times k}$ ,  $C \in \mathbb{R}^{m \times k}$ ,  $D \in \mathbb{R}^{m \times n}$  and consider the matrix

$$X = (\text{rank}(C^T DB) + 1)^{-1} BC^T D.$$

0  
1  
2

For **given**  $n, k, m$ , write down an expression for the minimal rank of  $X$  and the maximal rank of  $X$ , as well as a **brief** justification.

Minimal rank of  $X$ :

The minimal rank of  $X$  is 0. It is attained (among other possibilities) if any of the matrices  $B, C, D$  is 0.

Maximal rank of  $X$ :

For **given**  $n, k, m$  the maximal rank of  $X$  is  $\min(n, k, m)$ .  
The factor  $(\text{rank}(C^T DB) + 1)^{-1} > 0$  and thus does not change the rank. The rank of a product of matrices is smaller or equal than the minimal rank of the factors. The rank of any matrix  $M \in \mathbb{R}^{r \times c}$  is smaller or equal to  $\min(r, c)$ . It is not necessary to show that the rank can be attained. **Note:** To obtain the points a less detailed explanation is sufficient.

### Problem 3 MATLAB (7 credits)

0  
1  
2

a) Consider the matrix  $A$  and vector  $b$  given by

$$A = \begin{pmatrix} 4 & 6 \\ 9 & 2 \\ 7 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 8 \\ 6 \end{pmatrix}.$$

Write a MATLAB one-line expression for  $x$  that **for the specific**  $A, b$  gives a near-identical result to  $x_{\text{ref}} = A \setminus b$ , i.e.

`assert(norm(x-x_ref) < 1e-12)`

does not throw an error. Your expression may use only the following MATLAB symbols/functions  $A, b, *, +, -, \text{inv}, ', \text{rank}, \text{zeros}, \text{eye}, \text{ones}, \text{magic}$  and parenthesis.  $x =$

The rank of  $A$  is two which is equal to the number of columns. Therefore, the backslash operator behaves similar to `pinv(A)*b`. The solution is thus to compute the pseudo inverse: `x = inv(A' * A) * A' * b`. Note: In a real program use `A \ b` instead as it is faster and more stable.

0  
1  
2  
3  
4  
5

b) Let  $w$  be a vector in  $\mathbb{R}^3$  which is represented in MATLAB as a column vector  $w$  s.t. `size(w)` gives `[3, 1]`. Let  $R$  be a rotation matrix in  $SO(3)$  which is represented in MATLAB by  $R$ . Given are a large number of vectors as the rows of the matrix  $X$  in  $\mathbb{R}^{100\,000 \times 3}$  which is represented in MATLAB by  $X$ . Given are programA

```
function Y = programA(w, R, X)
    Y = zeros(size(X));
    for i = 1:size(X, 1)
        Y(i, :) = cross(w, (R * X(i,:)'))';
    end
end
```

and programB

```
function Y = programB(w, R, X)
    Y = cross(repmat(w', size(X, 1), 1), X * R');
end
```

which are called like `Y_A = programA(w, R, X); Y_B = programB(w, R, X)`.

Give an explanation of what is computed, i.e. what is the mathematical relationship between  $Y_A, Y_B$  and the inputs  $w, R, X$ . What is the relationship between  $Y_A$  and  $Y_B$ ? What do the rows of  $Y_A$  and  $Y_B$  contain?

$Y_A$  and  $Y_B$  are equal up to numerical error. Let  $y_i$  be the  $i$ -th row as a column vector, then there holds:  $y_i = w \times R x_i$  where  $x_i$  is the  $i$ -th row of  $X$  as a column vector.

Which of the two programs is significantly faster? Explain your choice.<sup>2</sup>

`programB` is significantly faster. The speedup is obtained by vectorization which eliminates the slow `for` loop in MATLAB, which is an interpreted language.

Write a one-line expression which computes the same result even faster, i.e. at least a factor of three in median run-time over 10 runs.<sup>3</sup> You are given  $w_{\text{hat}}$  which is the  $3 \times 3$  matrix corresponding to  $\hat{w}$ . Your expression may use only the following MATLAB symbols/functions  $R, X, w, w_{\text{hat}}, *, +, -, \text{inv}, ', \text{rank}, \text{zeros}, \text{eye}, \text{ones}, \text{magic}$  and parenthesis.  $Y_C =$

The solution is `X * (w_hat * R)'`. A possible other solution would be `(X * R') * w_hat'` which is ca. twice as slow as the first solution but still sufficiently fast for the problem statement.

<sup>2</sup>Assume execution on a standard desktop PC with an i5 CPU and without GPU. You do not have to use this fact in your explanation.

<sup>3</sup>This subproblem gives few points in comparison to the rest of the graded exercise.

#### Problem 4 Image Formation (4 credits)

A 3D point  $P = (2, 0, 4)^\top$  is observed by a camera, which has its optical center at  $C = (0, -1, 0)^\top$  and no rotation ( $R = \text{Id}_{3 \times 3}$ ). The intrinsic parameter matrix  $K$  is given by

$$K = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 400 & 240 \\ 0 & 0 & 1 \end{pmatrix}.$$

Calculate the pixel-position of the projected point in the image and tick the correct answer.

a)

- ☐  $u = 430$
- ☐  $u = 260$
- ☐  $u = 620$
- ☐  $u = 750$
- ☐  $u = 770$
- ☐  $u = 950$
- ☐  $u = 200$
- ☒  $u = 570$

b)

- ☒  $v = 340$
- ☐  $v = 950$
- ☐  $v = 650$
- ☐  $v = 860$
- ☐  $v = 350$
- ☐  $v = 810$
- ☐  $v = 490$
- ☐  $v = 670$

c) Given that the camera is nearly perfect what is the size of the image?

- ☐ height = 800 px, width = 1000 px
- ☒ height = 480 px, width = 640 px
- ☐ height = 240 px, width = 320 px
- ☐ height = 400 px, width = 500 px

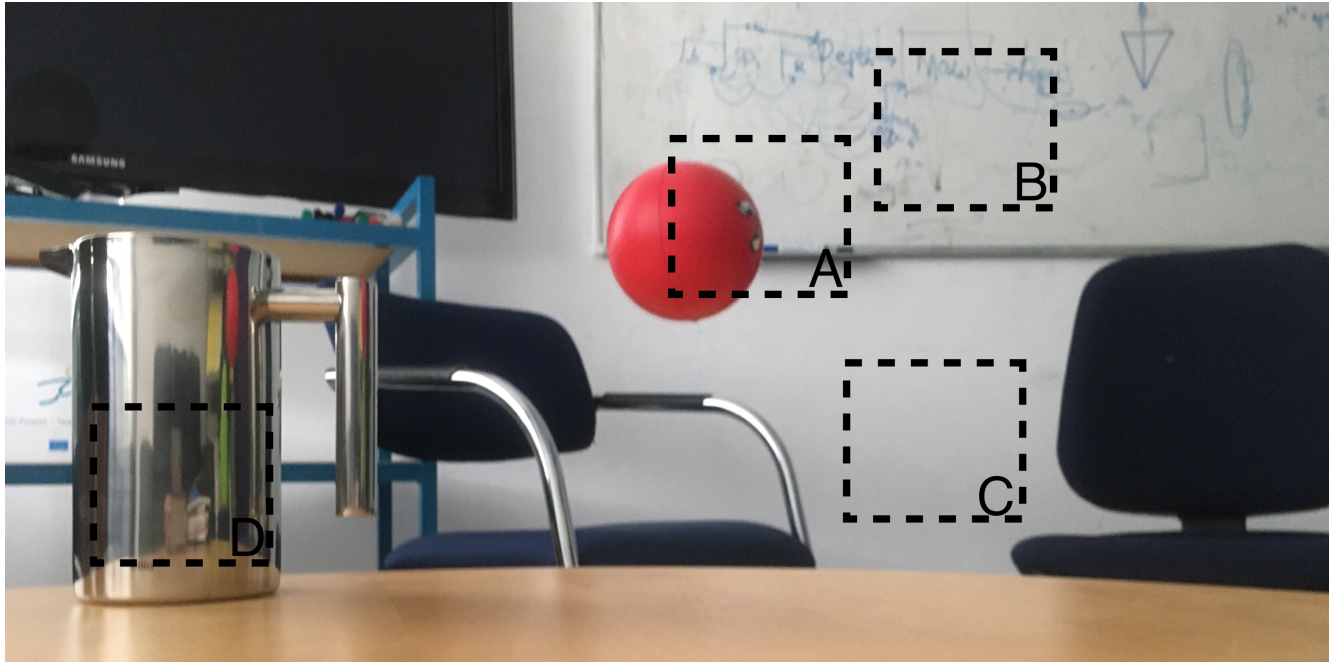
d) **Briefly** justify your answer from c).

For an ideal camera the principal point is in the middle of the image. This means that the dimensions of the image are ca. twice the principal point (up to pixel-center convention). The second sentence is true up to an error of a view pixels and none of the answers in c) are close enough to make the situation ambiguous.

☐ 0  
☐ 1

## Problem 5 The Lucas-Kanade Method (4 credits)

Given is the following image where the black, dashed boxes are annotations used within this exercise.



The image is the first image of an image pair used within the Lucas-Kanade algorithm. Assume that the second image is taken 0.2 seconds later. Furthermore assume that the camera motion is slow and that the translation and rotation can be in any direction. The red ball in A is falling. The Lucas-Kanade algorithm is used to estimate the optical flow for the two images. For the marked image regions A – D decide and explain if the algorithm will yield a good result. The neighborhood  $W(x)$  is set to be the same as the annotation boxes. Use **precise, technical** terms from the lecture for your explanations.

0  
1

a) Image region A:

It will not yield a good result. The ball is moving and the background is not, thus the **constant motion in a neighborhood assumption** is violated.

0  
1

b) Image region B:

It will yield a good result. The **brightness constancy assumption** and the **constant motion in a neighborhood assumption** are fulfilled. The neighborhood contains sufficient structure to avoid the **aperture problem**.

0  
1

c) Image region C:

It will not yield a good result. The wall is uniformly white and thus the **aperture problem** arises.

0  
1

d) Image region D:

It will not yield a good result. The carafe has a very specular (mirror-like) surface and thus the **brightness constancy assumption** is violated because the view point changes (this is given in the problem statement).

## Problem 6 Fundamental Matrix (5 credits)

Let  $F \in \mathbb{R}^{3 \times 3}$  be the fundamental matrix for the cameras  $C_1$  and  $C_2$ . Let  $K$  be the intrinsic camera matrix for  $C_1$ ,  $C_2$  and let  $T$  be the translation between the two camera centers as seen in the coordinate system of camera  $C_2$ . Let  $e_2$  be the epipole in the second image.

This exercise will prove that  $e_2^\top F = 0$  holds. **Below** each step there will be a solution box in which you should give a **brief** explanation of why the step is correct. Please **briefly** reference the appropriate material from the lecture or the tutorials. Basic mathematical facts do not need to be referenced.

a)

$$e_2^\top F = \left(\frac{1}{\lambda_1} KT\right)^\top (K^{-\top} \hat{T} R K^{-1})$$

 0  
1

The position of camera  $C_1$  in the coordinate system of camera  $C_2$  is given by  $T$ . Let  $\lambda_1$  be the depth of camera  $C_1$  in the coordinate system of camera  $C_2$  then the pixel coordinates of  $e_2$  are  $\frac{1}{\lambda_1} KT$ . The second parenthesis is the definition of the fundamental matrix.

b)

$$= \frac{1}{\lambda_1} T^\top K^\top K^{-\top} \hat{T} R K^{-1}$$

 0  
1

The transpose of a product of matrices is the product of the transposes matrices in reverse order.

c)

$$= \frac{1}{\lambda_1} T^\top \hat{T} R K^{-1}$$

 0  
1

The product of a matrix with its inverse matrix is the identity, which can be omitted in a product.

d)

$$= -\frac{1}{\lambda_1} (\hat{T} T)^\top R K^{-1}$$

 0  
1

The following holds  $(\hat{T} T)^\top = T^\top \hat{T}^\top = -T^\top \hat{T}$  due to the definition of skew-symmetric matrices.

e)

$$= 0$$

 0  
1

We have that  $\hat{T} T = T \times T = 0$ . It is also acceptable to reference  $\hat{T} T = 0$  from the tutorial with a concrete reference.

Sample Solution