EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-Naturwissenschaftliche Fakultät

Technische Informatik

Enkelejda Kasneci,
Thiago Santini
enkelejda.kasneci@uni-tuebingen.de,
thiago.santini@uni-tuebingen.de

Eye Movements and Visual Perception

Practical Session 3

Winter term 2017/18 Tübingen, 08. January 2018

Introduction

The aim of this practical session is to implement an algorithm for scanpath comparison.

Submit your answers to ILAS on a single folder using the **Upload Multiple Files as Zip-Archive** option. This folder should contain all items marked in this document with (Answer in: example.txt) . Name the file **exactly** as specified in this document with your answer inside. No file upload is required for items not marked as such. **Upload a single submission per group.**

Exercise 1: Group Identification

Add the matriculation number of all group members, **one per line**. (Answer in: group.txt)

Exercise 2: Read in fixation lists

1. Use the Matlab file "read_fixation_list.m" to read in the fixation lists.
2. Fixation lists are: "ex1.txt", "ex2.txt" and "ex3.txt".

Exercise 3: Convert fixation lists to strings

1. Lay a grid of size 5x5 over the image (500x500 pixels) and assign a letter to each field as shown in Figure 1.
2. Each fixation is marked by the letter from the grid field it hits.
3. To encode fixation duration, repeat the letter assigned to the fixation proportionally to its duration (use bins of 150 ms).

4. Calculate all strings without and with duration dependence.
   (Answer in: ex1-without-dependece.txt,
   ex2-without-dependece.txt,
   ex3-without-dependece.txt,
   ex1-with-dependece.txt,
   ex2-with-dependece.txt,
   ex3-with-dependece.txt)

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Figure 1: Letter grid to convert a fixation list into a string.

Exercise 4: Implement ScanMatch without gap

1. Use the string with duration dependence here.
2. Calculate the letter difference costs for each letter as shown in Figure 2.
3. Calculate the string difference using the letter costs for "ex1,ex2", "ex1,ex3" and "ex2,ex3".
4. Norm the results to 1 using the highest value possible (Maximum of all gray boxes multiplied by the length).
5. Compare the results.

Exercise 5: Add gaps to ScanMatch (Needleman-Wunsch algorithm)

1. Use the string with duration dependence here.
2. Instead of calculating the direct difference of each letter it should be possible to add a gap where the strings don't match. Why? (Answer in: add-gap.txt)
3. Needleman-Wunsch algorithm.
   a) use the mean of all negative cost entries in the cost tables as gap cost (-3.2).
   b) Create matrix with size(m+1,n+1) where m is the length of the first string and n the length of the second string.
   c) Initialize matrix as shown in Figure 3.
   d) Start in the top left empty cell.
   e) Calculate:
      i. Matrix[i-1,j] + gap
      ii. Matrix[i,j-1] + gap

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

-4	-3	-2	-3	-4
-3	-2	-1	-2	-3
-2	-1	60	-1	-2
-3	-2	-1	-2	-3
-4	-3	-2	-3	-4

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

100	-1	-2	-3	-4
-1	-2	-3	-4	-5
-2	-3	-4	-5	-6
-3	-4	-5	-6	-7
-4	-5	-6	-7	-8

Figure 2: Gray marks the box of the letter to which the table belongs. For each grid position difference from the letter box in x or y direction the costs increase by -1. The gray letter box gets the absolute value of all costs from its tabel.

iii. Matrix[i-1,j-1] + difference costs of letters from first string letter cost table

f) Store the highest value in the field and also store all directions of the calculation that lead to this value.

g) Fill the first row than start wit the next column.

h) If the matrix is filled follow all paths from the right bottom of the matrix to their end. Those paths are the calculated string alignments and the right bottom matrix entry is the score.

i) Calculate the scores for "ex1,ex2", "ex1,ex3" and "ex2,ex3".

j) Compare the results. Which are the two most similar sequences? (Answer in: similar.txt)

k) Calculate the string alignments(one for each comparison).
(Answer in: ex1-ex2.txt,
ex1-ex3.txt,
ex2-ex3.txt)

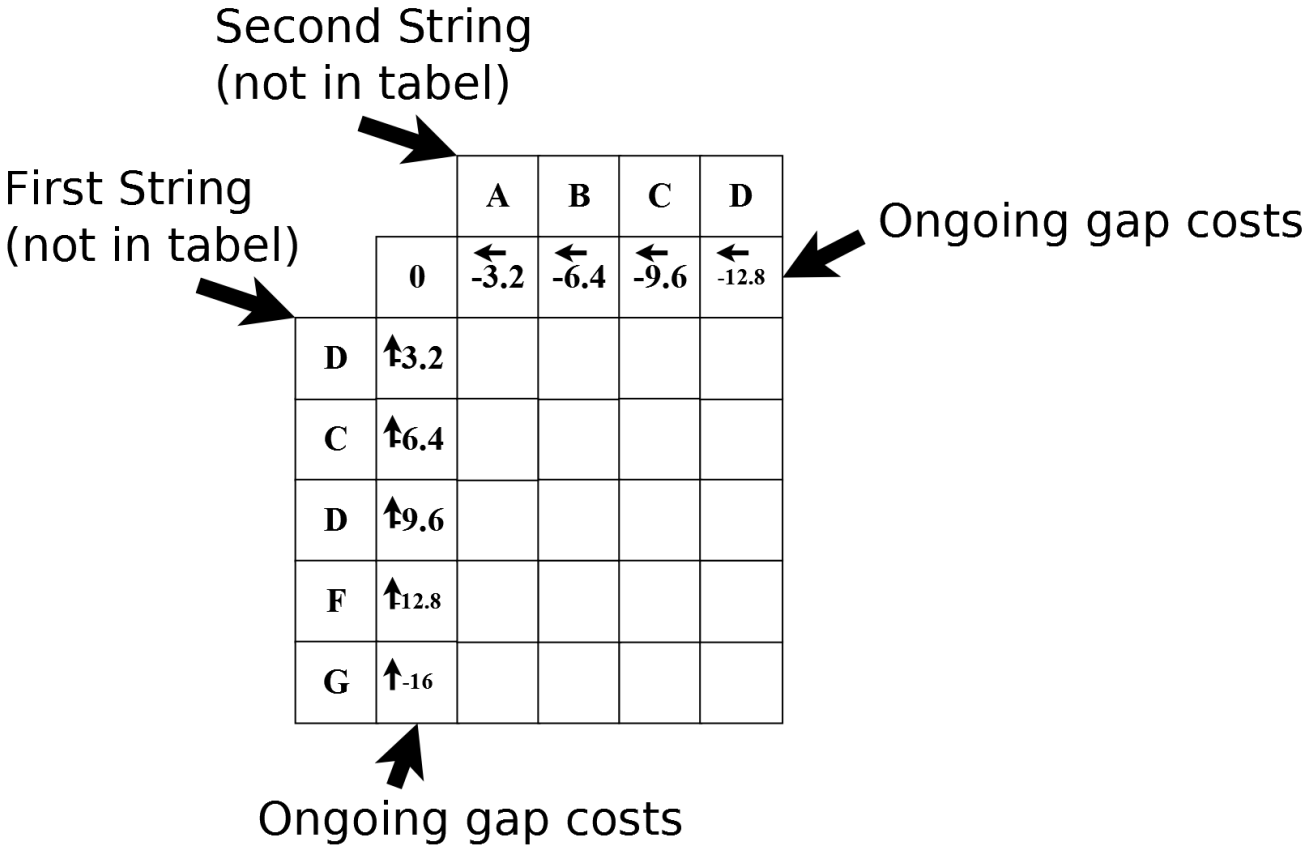

Figure 3: Initialisation of the matrix

		A	B	C	D
	0	-3.2	-6.4	-9.6	-12.8
D	3.2	-3	-5.2	-7.4	75.4
C	6.4	-5.2	-4	74.8	72.2
D	9.6	-8.4	-7.2	71.6	159.8
F	12.8	-10.6	-10.4	68.4	156.6
G	-16	-13.8	-11.6	65.2	153.4

A B C D _ _
_ D C D F G

Cost table D:

-3	-2	-1	85	-1
-4	-3	-2	-1	-2
-5	-4	-3	-2	-3
-6	-5	-4	-3	-4
-7	-6	-5	-4	-5

Cost table F:

-1	-2	-3	-4	-5
85	-1	-2	-3	-4
-1	-2	-3	-4	-5
-2	-3	-4	-5	-6
-3	-4	-5	-6	-7

Cost table C:

-2	-1	80	-1	-2
-3	-2	-1	-2	-3
-4	-3	-2	-3	-4
-5	-4	-3	-4	-5
-6	-5	-4	-5	-6

Cost table G:

-2	-1	-2	-3	-4
-1	70	-1	-2	-3
-2	-1	-2	-3	-4
-3	-2	-3	-4	-5
-4	-3	-4	-5	-6

Figure 4: The filled matrix with the resulting string alignment and the score 153.4.