7
Recognizer test

To evaluate the recognizer and try to find the combination of parameters that results in the best recognition experience, we have run a few tests. A dataset composed of 1357 shapes was created, containing all currently recognizable shapes and organized in "test cases": 335 radios (circles), 76 horizontal lines, 171 vertical lines, 81 horizontal zig-zags, 139 vertical zig-zags, 144 triangles, and 411 rectangles (distributed in three test cases with various size: 207 checkboxes, 160 buttons, and 44 frames). The dataset was then tested with different configurations of the recognizer, changing a number of parameters described below:

- Douglas-Peucker simplification algorithm usage → the recognizer may or may not use this simplification algorithm.

- Douglas-Peucker tolerance (DP Tol.) → if the recognizer uses the Douglas-Peucker algorithm, the tolerance used in the algorithm can vary. We tested values ranging from 1 to 14.

- Direction tolerance (Dir. Tol.) → when the line segment is converted into a string, only the directions determined by points distant from each other by a certain distance should be converted into a character direction. We tested this threshold varying it from 0 to 12.

- Levenshtein cost → the Levenshtein edit distance algorithm calculates the edit distance between string A and string B by performing the operations of insertion, deletion and substitution. Each operation has its own cost and the best result is given by the sequence of operations that sums up to the lesser cost. The algorithm creates a dynamic table where each cell (i,j) contains the best solution to the distance between substrings A[i] and B[j]. The last cell (where `i` is the length of string A and `j` is the length of string B) then gives the best solution for the whole string. Since the algorithm is based on filling up a table, the operations can be interpreted as the directions we travel in the table: the insertion operation is when we

go up, the delete operation when we go left and the substitution operation when we go diagonally. The operations' costs can be determined in different ways and we tested the following methods:

- o Costs equal to the diagonal difference → all the costs are the same and equal to the difference between character A[i] and B[j]:

```
diagCost = Math.Abs(A[aInx] - B[bInx]);
leftCost = Math.Abs(A[aInx] - B[bInx]);
upCost   = Math.Abs(A[aInx] - B[bInx]);
```

- o Diagonal cost 1 or 0, left and up 1 → the cost of going diagonally depends on if the characters are equal, while the costs of going left or up is always equal to 1:

```
diagCost = A[aInx] == B[bInx] ? 0 : 1;
leftCost = 1;
upCost   = 1;
```

- o Cost based on each direction and base case 0 → in this case we compare the cost to the characters in the string: if we go diagonally, we compare between the strings, if we go left, we compare the current character in string A to the previous on in the same string, if we go up, we compare the current character in string B with the previous in string B. If there is not a previous character, the cost is 0:

```
diagCost = Math.Abs(A[aInx] - B[bInx]);
leftCost = Math.Abs(A[aInx] - A[aInx-1]) : 0;
upCost   = Math.Abs(B[bInx] - B[bInx-1]) : 0;
```

- o Cost based on each direction and base case 1 → this case is similar to the previous one, but if there is not a previous character, the cost is 1:

```
diagCost = Math.Abs(A[aInx] - B[bInx]);
leftCost = Math.Abs(A[aInx] - A[aInx-1]) : 1;
upCost   = Math.Abs(B[bInx] - B[bInx-1]) : 1;
```

The distance we use, as described earlier in Section 4.4, is proportional to the segment's length, i.e. the distance is the distance calculated by the Levenshtein algorithm divided by the segment's length. With this we aim to reduce the effect of longer segments having more noise and being more error

prone. With the different costs tested, the distance does not necessarily fall within the range [0.0,1.0], since there are costs that may be greater than one. For example, when we use the `Math.Abs(A[aInx] - B[bInx])`, the resulting value can be 0, 1, 2, 3 or 4 (considering the 8 possible directions).

This test focused on associating each drawing of the dataset to a shape, i.e., the shape with the lesser distance from the drawing. To each recognizer configuration, we obtained the number of successful associations and the percentage from the total it represents. Another statistic obtained is the average shape success percentage and standard deviation, calculated by summing up each test case success percentage and dividing by the number of test cases. This value will indicate how well the configuration performed for the different shapes and sizes (the test cases) and if it was uniform between test cases, since a recognizer configuration can perform well with checkboxes rectangles but poorly with frame rectangles, for example.

Also we calculated an average shape success distance, by summing up each test case success distance and dividing by the number of test cases. This value cannot be compared between different configurations, since depending on how the Levenshtein's costs are calculated, the distance may be greater than when compared to other costs. This value is only an indicative of which value we should use to consider that the association to a shape was in fact a successful recognition.

We tested 780 different recognizer configurations in total. We will only present the top 10 results, first ordered by total success percentage (Table 4) and later by average shape success percentage (Table 5).

Table 4:   Top 10 recognizer configuration results organized by success percentage.

Dir. Tol.	Levenshtein Costs	DP Tol.	Success #	Success %	Average Shape Success %	Std Dev Shape Success %	Ave. Shape Dist.
9	equal to diag. diff.	1	1243	91.60%	93.29%	0.051	25.83
8	equal to diag. diff.	1	1240	91.38%	91.42%	0.061	27.11
9	equal to diag. diff.	2	1234	90.94%	93.55%	0.063	27.11
10	equal to diag. diff.	1	1231	90.71%	93.74%	0.079	23.89
9	equal to diag. diff.	3	1226	90.35%	93.14%	0.070	26.93
9	equal to diag. diff.	4	1225	90.27%	93.11%	0.072	27.27
8	equal to diag. diff.	2	1222	90.05%	91.25%	0.071	27.83
10	equal to diag. diff.	2	1220	89.90%	93.96%	0.096	25.08
8	equal to diag. diff.	3	1219	89.83%	91.06%	0.073	27.97
8	equal to diag. diff.	4	1219	89.83%	91.09%	0.074	28.22

Table 5:   Top 10 recognizer configuration results organized by average shape success percentage.

Dir. Tol.	Levenshtein Costs	DP Tol.	Success #	Success %	Average Shape Success %	Std Dev Shape Success %	Ave. Shape Dist.
10	equal to diag. diff.	2	1220	89.90%	93.96%	0.096	25.08
11	equal to diag. diff.	1	1214	89.46%	93.88%	0.110	22.61
10	equal to diag. diff.	1	1231	90.71%	93.74%	0.079	23.89
9	equal to diag. diff.	2	1234	90.94%	93.55%	0.063	27.11
10	equal to diag. diff.	3	1218	89.76%	93.47%	0.096	25.53
11	equal to diag. diff.	3	1182	87.10%	93.35%	0.143	22.63
11	equal to diag. diff.	2	1185	87.32%	93.33%	0.142	22.15
9	equal to diag. diff.	1	1243	91.60%	93.29%	0.051	25.83
9	equal to diag. diff.	3	1226	90.35%	93.14%	0.070	26.93
9	equal to diag. diff.	4	1225	90.27%	93.11%	0.072	27.27

The first observable conclusion is that the Levenshtein costs with best results was the first one described, with the cost equal to the diagonal difference. Another observation is that, although the use of the Douglas-Peucker algorithm provided better results, the tolerance used is small. This conclusion needs further investigation, since the recognizer with the same configuration of the best result in Table 4, except for the use of Douglas-Peucker, obtained a 83.27% success percentage.

A close-up investigation of the two best configurations can be seen in the following tables. In them, each column represents a test case. The first block of shadowed rows is a summary of the results: the first row shows the number of shapes, the second row shows the number of shapes associated correctly and the following rows show the success percentage, the successful associations' average distance and the standard deviation of such distances. The next block of rows provide a detailed investigation of the associations made: each block of rows represents a shape (displayed in the first column) and to each shape we show the number of associations made, the average distance and the standard deviation. The cells in boldface highlight the successful associations.

For example, if we take Table 6 and analyze the Radio test case (by observing the column marked with "Radio"), we can see that there were 335 shapes in the dataset that were drawn as radio buttons (i.e. drawn as small circles). 280 of these were successfully associated to the shape "Circle", which represents a success rate of 83.58%. Amongst the successes, the average distance was 0.40 with a standard deviation of 0.11. Looking closely at the results, we can see that

the 55 circles that were not associated correctly (335 shapes - 280 successes) were mostly recognized as rectangles. Actually, 49 circles were incorrectly associated to rectangles, with an average distance of 0.36 and a 0.10 standard deviation. 5 other circles were considered triangles and 1 circle was considered a horizontal zig-zag.

Table 6:   Shape analysis for best success percentage configuration.

		Radio	Label	HorizontalZZ	Checkbox	Button	Frame	Triangle	Vertical	VerticalZZ
	Shapes #	335	76	81	207	160	44	144	171	139
	Success #	280	73	80	197	147	42	133	171	120
	Success %	83.58%	96.05%	98.77%	95.17%	91.88%	95.45%	92.36%	100.00%	86.33%
	Success Ave. Dist.	0.40	0.12	0.09	0.08	0.17	0.08	0.15	0.00	0.35
	Success Std. Dev.	0.11	0.24	0.16	0.13	0.19	0.10	0.22	0.00	0.20
Circle	#	**280**	0	0	6	8	2	0	0	0
Circle	Ave. Dist.	**0.40**	0.00	0.00	0.37	0.33	0.23	0.00	0.00	0.00
Circle	Std. Dev.	**0.11**	0.00	0.00	0.16	0.14	0.11	0.00	0.00	0.00
Horizontal	#	0	**73**	0	1	0	0	0	0	0
Horizontal	Ave. Dist.	0.00	**0.12**	0.00	0.71	0.00	0.00	0.00	0.00	0.00
Horizontal	Std. Dev.	0.00	**0.24**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Horizontal Zig-Zag	#	1	0	**80**	1	0	0	0	0	0
Horizontal Zig-Zag	Ave. Dist.	0.71	0.00	**0.09**	0.57	0.00	0.00	0.00	0.00	0.00
Horizontal Zig-Zag	Std. Dev.	0.00	0.00	**0.16**	0.00	0.00	0.00	0.00	0.00	0.00
Rectangle	#	49	0	0	**197**	**147**	**42**	11	0	6
Rectangle	Ave. Dist.	0.36	0.00	0.00	**0.08**	**0.17**	**0.08**	0.29	0.00	0.77
Rectangle	Std. Dev.	0.10	0.00	0.00	**0.13**	**0.19**	**0.10**	0.07	0.00	0.08
Triangle	#	5	3	1	2	5	0	**133**	0	2
Triangle	Ave. Dist.	0.58	0.47	0.60	0.25	0.60	0.00	**0.15**	0.00	0.79
Triangle	Std. Dev.	0.07	0.05	0.00	0.08	0.12	0.00	**0.22**	0.00	0.04
Vertical	#	0	0	0	0	0	0	0	**171**	11
Vertical	Ave. Dist.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.00**	0.77
Vertical	Std. Dev.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.00**	0.02
Vertical Zig-Zag	#	0	0	0	0	0	0	0	0	**120**
Vertical Zig-Zag	Ave. Dist.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.35**
Vertical Zig-Zag	Std. Dev.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.20**

Table 7: Shape analysis for best average success percentage configuration.

		Radio	Label	HorizontalZZ	Checkbox	Button	Frame	Triangle	Vertical	VerticalZZ
	Shapes #	335	76	81	207	160	44	144	171	139
	Success #	225	75	80	201	154	42	138	171	134
	Success %	67.16%	98.68%	98.77%	97.10%	96.25%	95.45%	95.83%	100.00%	96.40%
	Success Ave. Dist.	0.47	0.10	0.08	0.07	0.16	0.08	0.14	0.00	0.34
	Success Std. Dev.	0.11	0.23	0.17	0.12	0.22	0.11	0.22	0.00	0.20
Circle	#	**225**	0	0	4	3	1	0	0	0
Circle	Ave. Dist.	**0.47**	0.00	0.00	0.40	0.44	0.15	0.00	0.00	0.00
Circle	Std. Dev.	**0.11**	0.00	0.00	0.19	0.16	0.00	0.00	0.00	0.00
Horizontal	#	0	**75**	0	0	0	0	0	0	0
Horizontal	Ave. Dist.	0.00	**0.10**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Horizontal	Std. Dev.	0.00	**0.23**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Horizontal Zig-Zag	#	5	1	**80**	0	0	0	0	0	0
Horizontal Zig-Zag	Ave. Dist.	0.63	0.50	**0.08**	0.00	0.00	0.00	0.00	0.00	0.00
Horizontal Zig-Zag	Std. Dev.	0.07	0.00	**0.17**	0.00	0.00	0.00	0.00	0.00	0.00
Rectangle	#	90	0	0	**201**	**154**	**42**	6	0	0
Rectangle	Ave. Dist.	0.39	0.00	0.00	**0.07**	**0.16**	**0.08**	0.31	0.00	0.00
Rectangle	Std. Dev.	0.15	0.00	0.00	**0.12**	**0.22**	**0.11**	0.07	0.00	0.00
Triangle	#	15	0	1	2	3	1	**138**	0	0
Triangle	Ave. Dist.	0.57	0.00	0.50	0.30	0.68	0.55	**0.14**	0.00	0.00
Triangle	Std. Dev.	0.10	0.00	0.00	0.10	0.05	0.00	**0.22**	0.00	0.00
Vertical	#	0	0	0	0	0	0	0	**171**	5
Vertical	Ave. Dist.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.00**	0.77
Vertical	Std. Dev.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.00**	0.02
Vertical Zig-Zag	#	0	0	0	0	0	0	0	0	**134**
Vertical Zig-Zag	Ave. Dist.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.34**
Vertical Zig-Zag	Std. Dev.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.20**

Comparing the overall results, it is possible to see that the results shown in Table 7 were better than those shown in Table 6, except for the "Radio" case. This shows that the circle case, shape that has no vertices, demand a more thoughtful approach. Due to this large difference, we chose to keep the configuration of best success percentage (the one from Table 6).

Further investigation of the different configurations may point to a better solution than the one chosen. A larger dataset may be needed and the results obtained should be compared to other approaches, such as those that will be described in Section 9.2.