Texture classification with thousands of features

A Kadyrov, A Talepbour and M Petrou,
School of Electronics, Computing and Mathematics,
University of Surrey, Guildford GU2 7XH, United Kingdom

Abstract

The Trace transform is a generalisation of the Radon transform that allows one to construct image features that do not necessarily have meaning in terms of human perception, but they measure different image characteristics. The ability of producing thousands of features from an image allows one to be selective as to which are appropriate for a particular task. In this paper we propose the use of such an approach to the problem of texture discrimination and compare its results with the classical co-occurrence matrix approach where usually the features used are fewer than ten.

1 Introduction

The wealth of objects around us requires a wealth of descriptors. It is very unlikely that a few characteristics measured from the images of these objects will suffice to allow us to discriminate all objects we see. And yet our brain recognises thousands of objects, working mostly at the sub-conscious level. That is the reason knowledge engineering is very difficult: it is very hard to identify the characteristics which allow us to identify easily so many different faces, objects, materials, textures etc. Restricting ourselves, therefore, in computer vision to features that we can *consciously* identify as characterising our cognition excludes the vast number of features that our *sub-conscious* uses and which we cannot usually identify. We may, however, replace the mechanism of our sub-conscious with a Mathematical tool that allows us to construct thousands of features that do not have physical or other meaning: we may use the Trace transform which is an alternative image representation and from which we can construct the so called triple features [2]. In [2] it was shown how one can construct such features invariant to rotation, translation and scaling, while in [3] it was shown how to construct object signatures invariant to affine transforms. In this paper we are not trying to construct invariant features. Instead we are simply trying to construct many features, each of which captures some aspect of the image. We then use a simple form of training that allows us to give relative importance to each feature with respect to the task we have. In the specific example presented, the task in question is texture discrimination.

One of the most well established methods for texture discrimination is based on the use of co-occurrence matrices, and in particular on the use of features extracted from them [1]. The co-occurrence matrix captures the second order statistics of a stationary texture and computes some quantities from them that have perceptual meaning: contrast, homogeneity, etc. As co-occurrence matrices are considered a bench mark for texture analysis, we are going to use them here to discriminate textures from the Brodatz album and compare their performance with the results produced by the Trace transform method.

This paper is organised as follows: In section 2 we present a brief overview of the Trace transform and the way it is used to extract features from an image. In section 3 we present the training method we use to establish the usefulness of each produced feature. In section 4 we present the results of the texture discrimination experiments and in section 5 we present our conclusions.

2   Triple-feature construction from the Trace Transform

Let us imagine an image criss-crossed by all possible lines one can draw on it. Each line is characterised by parameters ϕ and p defined in figure 1. The Trace transform calculates a functional T over parameter t along line (ϕ, p). One then calculates another functional, P, along the columns of the Trace transform, ie over parameter p, and finally a functional Φ over the string of numbers created this way, ie over parameter ϕ. The result is a single number. With an appropriate choice of the three functionals T, P and Φ, one can make this number to be invariant to a certain group of transformations [2]. On the other hand, one may use different combinations of functionals to produce different features. For example, 10 different functionals T, P and Φ will produce 10^3 features.

The way these features are constructed implies that each of them measures something along batches of parallel lines (functional p applied to constant ϕ but over all values of p) and finally something across different batches of such parallel lines (functional Φ applied over all different directions ϕ). The use of different functionals implies measuring different footprints of the image along these constructs.
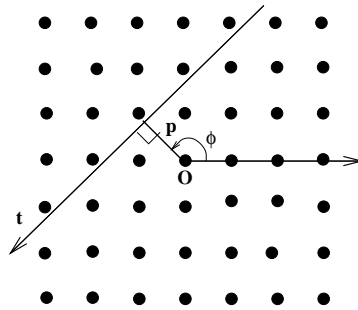

Figure 1: Definition of the parameters of an image tracing line

Table 1 shows the functionals used as Trace functional T in these experiments. Figure 2 shows an original texture image and its trace transform constructed by using the first functional from Table 1.

Table 2 shows the so called *diametric functionals* P and table 3 shows the so called *circus functionals* Φ. As we have 31 Trace functionals, 10 diametric functionals and 18 circus functionals, we have a total of 5880 features, by combining them in all possible ways. Each feature is denoted by a three part number, the first one identifying which T functional, the second which P functional, and the third which Φ functional was used to produce it.

	Functional
1	$\sum_{i=0}^{N} x_i$
2	$\sum_{i=0}^{N} i x_i$
3	2nd_Central_Moment/Sum_of_all_values
4	$\sqrt{\sum_{i=0}^{N} x_i^2}$
5	$Max_{i=0}^{N} x_i$
6	$\sum_{i=0}^{N-1} |x_{i+1} - x_i|$
7	$\sum_{i=0}^{N-1} |x_{i+1} - x_i|^2$
8	$\sum_{i=3}^{N-3} |x_{i-3} + x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2} - x_{i+3}|$
9	$\sum_{i=0}^{N-2} |x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2}|$
10	$\sum_{i=4}^{N-4} |x_{i-4} + x_{i-3} + \ldots + x_{i-1} - x_{i+1} - \ldots - x_{i+3} - x_{i+4}|$
11	$\sum_{i=5}^{N-5} |x_{i-5} + x_{i-4} + \ldots + x_{i-1} - x_{i+1} - \ldots - x_{i+4} - x_{i+5}|$
12	$\sum_{i=6}^{N-6} |x_{i-6} + x_{i-5} + \ldots + x_{i-1} - x_{i+1} - \ldots - x_{i+5} - x_{i+6}|$
13	$\sum_{i=7}^{N-7} |x_{i-7} + x_{i-6} + \ldots + x_{i-1} - x_{i+1} - \ldots - x_{i+6} - x_{i+7}|$
14	$\sum_{i=4}^{N-4} \sum_{k=0}^{4} |x_{i-k} - x_{i+k}|$
15	$\sum_{i=5}^{N-5} \sum_{k=0}^{5} |x_{i-k} - x_{i+k}|$
16	$\sum_{i=6}^{N-6} \sum_{k=0}^{6} |x_{i-k} - x_{i+k}|$
17	$\sum_{i=7}^{N-7} \sum_{k=0}^{7} |x_{i-k} - x_{i+k}|$
18	$\sum_{i=10}^{N-10} \sum_{k=0}^{10} |x_{i-k} - x_{i+k}|$
19	$\sum_{i=15}^{N-15} \sum_{k=0}^{15} |x_{i-k} - x_{i+k}|$
20	$\sum_{i=20}^{N-20} \sum_{k=0}^{20} |x_{i-k} - x_{i+k}|$
21	$\sum_{i=25}^{N-25} \sum_{k=0}^{25} |x_{i-k} - x_{i+k}|$
22	$\sum_{i=10}^{N-10} ((1 + \sum_{k=0}^{10} |x_{i-k} - x_{i+k}|) / (1 + \sum_{k=-10}^{9} |x_{i-k} - x_{i+k}|))$
23	$\sum_{i=10}^{N-10} \sqrt{(\sum_{k=0}^{10} |x_{i-k} - x_{i+k}|)^2 / (1 + \sum_{k=-10}^{9} |x_{i-k} - x_{i+k}|)}$
24	$\sum_{i=0}^{N-2} |x_i - 2x_{i+1} + x_{i+2}|$
25	$\sum_{i=0}^{N-3} |x_i - 3x_{i+1} + 3x_{i+2} - x_{i+3}|$
26	$\sum_{i=0}^{N-4} |x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4}|$
27	$\sum_{i=0}^{N-5} |x_i - 5x_{i+1} + 10x_{i+2} - 10x_{i+3} + 5x_{i+4} - x_{i+5}|$
28	$\sum_{i=0}^{N-2} |x_i - 2x_{i+1} + x_{i+2}| x_{i+1}$
29	$\sum_{i=0}^{N-3} |x_i - 3x_{i+1} + 3x_{i+2} - x_{i+3}| x_{i+1}$
30	$\sum_{i=0}^{N-4} |x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4}| x_{i+2}$
31	$\sum_{i=0}^{N-5} |x_i - 5x_{i+1} + 10x_{i+2} - 10x_{i+3} + 5x_{i+4} - x_{i+5}| x_{i+2}$

Table 1: The trace functionals T used in the experiments. N is the total number of points along the line and x_i is the ith sample along the tracing line.

	Functional
1	$Max_{i=0}^{N} x_i$
2	$Min_{i=0}^{N} x_i$
3	$\sqrt{\sum_{i=0}^{N} x_i^2}$
4	$\frac{\sum_{i=0}^{N} i x_i}{\sum_{i=0}^{N} x_i}$
5	$\sum_{i=0}^{N} i x_i$
6	$\frac{1}{N}\sum_{i=0}^{N}(x_i - \hat{x})^2$
7	c so that: $\sum_{i=0}^{c} x_i = \sum_{i=c}^{N} x_i$
8	$\sum_{i=0}^{N-1}|x_{i+1} - x_i|$
9	c so that: $\sum_{i=0}^{c}|x_{i+1} - x_i| = \sum_{i=c}^{N-1}|x_{i+1} - x_i|$
10	$\sum_{i=0}^{N-4}|x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4}|$

Table 2: The diametric functionals P used in the experiments

	Functional
1	$\sum_{i=0}^{N-1}|x_{i+1} - x_i|^2$
2	$\sum_{i=0}^{N-1}|x_{i+1} - x_i|$
3	$\sqrt{\sum_{i=0}^{N} x_i^2}$
4	$\sum_{i=0}^{N} x_i$
5	$Max_{i=0}^{N} x_i$
6	$Max_{i=0}^{N} x_i - Min_{i=0}^{N} x_i$
7	i so that $x_i = Min_{i=0}^{N} x_i$
8	i so that $x_i = Max_{i=0}^{N} x_i$
9	i so that $x_i = Min_{i=0}^{N} x_i$ without first harmonic
10	i so that $x_i = Max_{i=0}^{N} x_i$ without first harmonic
11	Amplitude of the first harmonic
12	Phase of the first harmonic
13	Amplitude of the second harmonic
14	Phase of the second harmonic
15	Amplitude of the third harmonic
16	Phase of the third harmonic
17	Amplitude of the fourth harmonic
18	Phase of the fourth harmonic

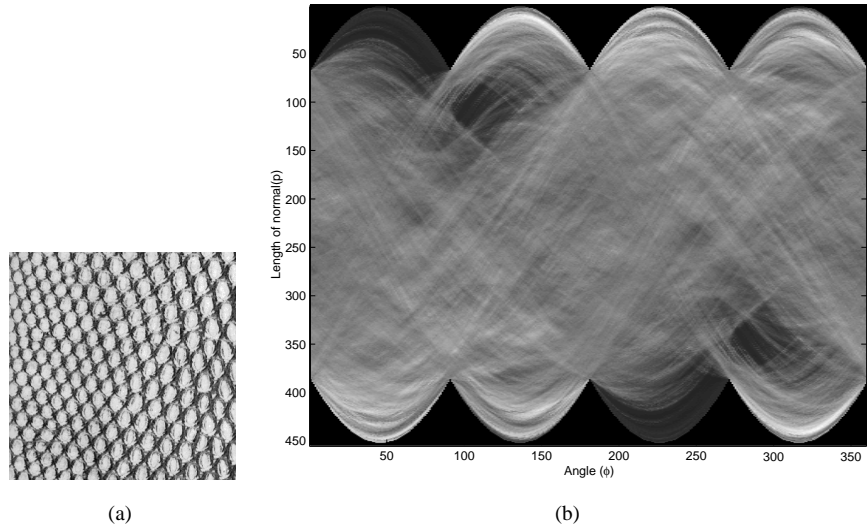Table 3: The circus functionals Φ used in the experiments

Figure 2: (a) An original texture from the Brodatz album and (b) its trace transform computed with the first functional of table 1.

3 Determining the feature relevance

To demonstrate our ideas we consider a database consisting of 112 images of texture from the Brodatz album, obtained from [4]. These images are 640×640 in size. From each image five sub-images were created by dividing it into four quadrants and also extracting a central subimage the same size as the quadrants. These sub-images are 320×320 pixels and they constitute a database of 560 textures of 112 classes. Let us call the set of 112 different textures made up from the top left quadrants, set TL, that made up from the top right quadrants, set TR, that made up from the bottom left quadrants, set BL, that made up from the bottom right quadrants set BR, and that made up from the central panels set C. We choose sets TL and TR to be used for training the system, ie for computing the relevance of each feature, and the remaining three sets for testing.

Let us consider a feature f_{klm} computed by combining the kth trace functional, with the lth diametrical functional and the mth circus functional. Let us say that its value for texture class c, and instantiation of this class s is denoted by f_{klm}^{cs}. Since we have two instantiations of each class in our training set, one from set TL and one from set TB, $s = 1$ or $s = 2$. We compute the average value of this feature over all $s = 1$ substantiations of the training texture samples we have. (Note that it does not really matter which instantiation we call $s = 1$ and which we call $s = 2$.):

$$m_{klm}^1 = \frac{1}{N} \sum_{c=1}^{N} f_{klm}^{c1} \tag{1}$$

where here N is the total number of texture classes.

We also compute the standard deviation of this feature over all classes:

$$\sigma^1_{klm} = \sqrt{\frac{1}{N}\sum_{c=1}^{N}(f^{c1}_{klm} - m^1_{klm})^2} \qquad (2)$$

A feature is useful in characterising textures, if its value is stable when the instantiation of a texture changes. So we define an average stability measure for each feature and scale it by the variance of the values of the feature over the whole database:

$$q_{klm} = \frac{1}{\sigma^1_{klm}}\sqrt{\frac{1}{N}\sum_{c=1}^{N}(f^{c1}_{klm} - f^{c2}_{klm})^2} \qquad (3)$$

The smaller q_{klm} is, the better feature f_{klm} is. We may set a threshold Q which will allow us to give weights to the features:

$$w_{klm} = \begin{cases} Q - q_{klm} & \text{if } q_{klm} < Q, \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Finally, the "distance" $dist$ between a test sample $test$ and any reference sample ref can be obtained from

$$dist(test, ref) \equiv \sum_{klm} w_{klm} \frac{1}{\sigma^1_{klm}}|f^{test}_{klm} - f^{ref}_{klm}| \qquad (5)$$

Note that as the value of Q increases, more features are included in the computation of $dist$.

4   Experimental results

The tracing lines used were such that each batch of parallel lines consisted of lines 2 inter-pixel distances apart. Each line was sampled by parameter t defined in figure 1, so that the sampling points were also 2 inter-pixel distances apart. For each value of p, 20 different orientations were used, ie the orientations of the lines with the same p differed by 18°. To avoid having lines of different lengths we only considered the part of each image that was inside the maximum inscribed circle in each 320×320 pixels square.

The significance of each feature was extracted from the training samples, and subsequently each one of the test samples was associated with the reference texture with which the distance value computed by equation (5) was least. Table 4 presents the results of this approach for identifying the correct class of a texture as the most similar one, the second most similar one, the third most similar one, the fourth most similar one, and beyond, presenting the numbers under the corresponding columns. All these numbers are out of 112, as we present the results of testing separately for each set of data. In all cases the reference set was the TL set of images. Each row of results corresponds to a different choice

6 6 1

of threshold Q. Note that after a certain value of Q (which is about 0.3) all features are used in the calculation and the performance of the system stabilises.

Q	Test Set BL					Test Set BR					Test Set C				
	1	2	3	4	R	1	2	3	4	R	1	2	3	4	R
0.1	1	1	1	1	108	1	1	1	1	1	108	1	1	1	108
0.2	35	19	6	4	48	38	14	6	6	48	39	11	4	5	53
0.3	85	9	5	1	12	84	10	2	0	16	70	12	6	5	19
0.4	90	7	3	3	9	89	10	2	1	10	80	9	5	3	15
0.5	93	5	1	4	9	90	11	1	1	9	81	11	3	3	14
0.6	93	6	0	3	10	89	11	3	0	9	80	11	3	5	13
0.7	90	8	0	3	11	89	12	2	0	9	80	10	4	4	14
0.8	90	8	0	1	13	89	13	1	0	9	79	10	5	4	14
0.9	88	10	1	1	12	89	14	0	0	9	79	11	4	3	15
1.0	92	6	1	4	9	92	10	1	0	9	81	8	6	4	13
1.1	93	5	3	3	8	93	8	1	0	10	80	11	6	2	13
1.2	93	5	4	3	7	95	6	1	0	10	78	16	4	2	12
1.3	93	5	6	1	7	95	6	0	1	10	82	11	3	4	12
1.4	94	5	2	3	8	93	7	1	1	10	82	11	2	2	15
1.5	94	5	3	2	8	90	10	1	1	10	80	13	2	2	15
1.7	94	2	5	4	7	88	10	2	2	10	78	14	3	2	15
2.3	93	3	5	3	8	85	11	4	2	10	75	12	4	5	16
2.6	92	5	4	2	9	84	12	3	2	11	74	11	6	4	17
3.0	89	8	2	4	9	83	12	4	2	11	74	8	9	2	19

Table 4: Each texture of the set indicated was used to enquiry the database of reference textures. Under the headings 1, 2, 3 and 4 we show how many times the correct texture appeared in the first, second, third and fourth position of the returned answer, respectively. Under heading R we show how many times it appeared in one of the remaining positions. The results are shown for different values of threshold Q.

For comparison we also used the co-occurrence matrix approach. Each co-occurrence matrix was constructed to be rotationally symmetric, ie the pairs of pixels considered were at a certain fixed distance d from each other, but not necessarily at a fixed orientation with respect to the image coordinates. From each co-occurrence matrix we computed the following 6 features:

- Energy:

$$\sum_{i=0}^{255} \sum_{j=0}^{255} p(i,j)^2 \tag{6}$$

- Entropy:

$$\sum_{i=0}^{255} \sum_{j=0}^{255} p(i,j) \log p(i,j) \tag{7}$$

- Contrast:

$$\sum_{i=0}^{255} \sum_{j=0}^{255} (i-j)^2 p(i,j) \tag{8}$$

- Correlation:

$$\frac{\sum_{i=0}^{255} \sum_{j=0}^{255} ij p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{9}$$

where

$$\mu_x = \sum_{i=0}^{255} i \sum_{j=0}^{255} p(i,j) \qquad \mu_y = \sum_{j=0}^{255} j \sum_{i=0}^{255} p(i,j) \tag{10}$$

$$\sigma_x = \sum_{i=0}^{255} (i-\mu_x)^2 \sum_{j=0}^{255} p(i,j) \qquad \sigma_y = \sum_{j=0}^{255} (j-\mu_y)^2 \sum_{i=0}^{255} p(i,j) \tag{11}$$

- Homogeneity:

$$\sum_{i=0}^{255} \sum_{j=0}^{255} \frac{p(i,j)}{1+|i-j|} \tag{12}$$

where $p(i,j)$ is the fraction of pairs of pixels that are at the particular distance d from each other and one of them has grey value i while the other has grey value j.

These features are then treated the same way as the trace transform based features, ie their relevance to the problem is computed from the training samples. Texture classification is then performed in the same way as for the proposed approach. Tables 5 and 6 show the results obtained for $d = 5$ and $d = 2$ respectively.

5 Conclusions

We advocate here the use of thousands of features for the problem of texture classification. These features do not need to make sense to the human perception, and therefore their number can be very large. The relevance of these features to the task we wish to solve can be assessed in a training phase, and then these features can be combined with their appropriate weights to form a similarity measure between two images. The proposed method tested with hundreds of textures from the Brodatz album was shown to be much more powerful than the commonly used method based on co-occurrence matrix features.

The training of our method does not have to be done with representations of all the textures we wish to identify. In another series of experiments performed, the results of which cannot be reported here due to lack of space, we used only 30 different textures from the training database to train the system, and then we used the features to recognise all texture classes, even texture classes that were not represented in the training set used to decide the relative importance of the features. The results were only slightly worse than the results reported here. For example, in the reported experiments, the best recognition rate was about 93 out of 112. In the experiemnts with the limited training, the best results were about 85 out of 112, still significantly better than those of the co-occurrence matrices.

Q	Test Set BL					Test Set BR					Test Set C				
	1	2	3	4	R	1	2	3	4	R	1	2	3	4	R
0.1	1	1	1	1	108	1	1	1	1	108	1	1	1	1	108
0.2	1	1	1	1	108	1	1	1	1	108	1	1	1	1	108
0.3	37	20	11	8	36	43	21	13	6	29	33	17	8	9	45
0.4	49	17	5	8	33	58	15	10	4	25	39	15	7	9	42
0.5	52	15	10	7	28	59	18	10	4	21	39	19	7	3	44
0.6	51	18	10	5	28	59	20	9	4	20	39	21	6	4	42
0.7	55	14	13	3	27	60	18	10	5	19	42	18	4	7	41
0.8	55	15	13	2	27	60	18	10	4	20	42	17	9	6	38
0.9	54	17	10	4	27	60	17	10	5	20	42	18	8	6	38
1.0	54	17	10	5	26	60	17	9	5	21	42	17	8	7	38
1.1	55	17	9	5	26	60	18	8	5	21	42	17	8	8	37
1.2	55	18	8	5	26	61	17	8	5	21	42	17	8	7	38
1.3	55	19	7	5	26	62	16	8	6	20	42	17	8	7	38
1.4	55	20	6	5	26	62	16	8	6	20	42	17	8	7	38
1.5	55	20	6	5	26	63	15	8	6	20	42	16	8	8	38
1.6	55	20	6	5	26	64	14	8	6	20	43	15	8	8	38
1.7	55	20	6	5	26	64	14	8	6	20	43	15	8	8	38
1.8	55	20	6	5	26	64	14	8	6	20	44	14	8	8	38
1.9	55	20	6	5	26	64	14	8	6	20	44	14	8	8	38
2.0	55	20	6	5	26	65	13	8	7	19	44	14	8	8	38
2.1	55	20	6	5	26	65	13	8	7	19	44	14	8	8	38
2.2	55	20	6	5	26	65	13	8	7	19	44	14	8	8	38
2.3	55	20	6	5	26	65	13	8	7	19	44	14	8	8	38

Table 5: The results from the co-occurrence matrix approach for $d = 5$. The arrangement is the same as for table 4.

References

[1] R M Haralick, 1985. "Statistical Image Texture Analysis", in Handbook of Pattern Recognition and Image Processing, T Y Young and K S Fu (eds), Academic Press, pp 247–279.

[2] A Kadyrov and M Petrou, 2001. "The Trace transform and its applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, Vol 23, pp 811–828.

[3] A Kadyrov and M Petrou, 2001. "Object descriptors invariant to affine distortions". British Machine Vision Conference, BMVC2001, Sept 10–13, Manchester, UK, T Cootes and C Taylor (eds), Vol 2, pp 391–400.

[4] http://www.ux.his.no/ tranden/brodatz.html

Q	Test Set BL					Test Set BR					Test Set C				
	1	2	3	4	R	1	2	3	4	R	1	2	3	4	R
0.1	1	1	1	1	108	1	1	1	1	108	1	1	1	1	108
0.2	1	1	1	1	108	1	1	1	1	108	1	1	1	1	108
0.3	43	13	16	6	34	46	21	12	3	30	3 3	18	8	6	47
0.4	52	17	10	3	30	61	17	7	5	22	3 8	19	8	5	42
0.5	56	12	12	1	31	60	19	8	2	23	4 0	17	6	6	43
0.6	54	15	11	3	29	65	14	7	2	24	3 9	17	6	10	40
0.7	54	17	12	4	25	65	14	9	3	21	3 9	17	7	10	39
0.8	54	19	9	6	24	66	13	8	3	22	4 1	14	10	9	38
0.9	55	18	10	5	24	65	18	6	2	21	4 0	17	8	9	38
1.0	56	17	10	5	24	64	19	7	1	21	4 1	16	7	10	38
1.1	57	16	10	5	24	64	19	7	1	21	4 1	16	6	10	39
1.2	57	17	9	5	24	63	20	6	2	21	4 1	16	7	9	39
1.3	57	17	9	5	24	64	19	6	3	20	4 1	16	7	8	40
1.4	57	17	9	5	24	64	19	7	2	20	4 1	15	8	8	40
1.5	57	17	9	5	24	64	19	6	3	20	4 1	15	9	7	40
1.6	57	18	8	5	24	63	20	6	2	21	4 2	14	9	6	41
1.7	58	17	7	6	24	63	20	6	2	21	4 2	14	8	7	41
1.8	58	17	7	6	24	63	20	6	2	21	4 2	14	8	7	41
1.9	58	17	7	6	24	63	20	6	2	21	4 2	14	8	7	41
2.0	58	17	7	6	24	63	20	6	1	22	4 1	15	9	6	41
2.1	58	17	7	6	24	63	20	6	1	22	4 2	14	9	7	40
2.2	58	17	7	6	24	63	20	6	1	22	4 2	14	9	7	40
2.3	58	17	7	6	24	63	20	6	1	22	4 2	14	9	7	40

Table 6: The results from the co-occurrence matrix approach for $d = 2$. The arrangement is the same as for table 4.