Run Length Matrices For Texture Analysis

Nicholas J. Tustison and James C. Gee

May 27, 2008

Penn Image Computing and Science Laboratory
University of Pennsylvania

Abstract

Texture analysis provides quantitative information describing properties in images such as coarseness and smoothness. Two common quantification schemes are based on co-occurence matrices and run-length matrices. Although the co-occurence measures are readily available in the Insight Toolkit, no such set of classes exists for run-length measures. This submission is meant to remedy this deficiency by providing a set of classes which are modeled after the ITK co-occurence measures classes.

`run-length, texture`

1 Introduction

Each element, $p(i, j, \theta)$, of a grey level run-length 2-D matrix gives the total number of occurrences of grey-level runs of length j and of intensity value i in the direction θ. A set of consecutive pixels with identical or similar intensity values constitutes a *grey level run*. The various measures calculated from the grey-level run length matrix of an image can be used as distinguishing features similar to the co-occurence matrix measures.

The five traditional run-length features originally proposed by Galloway [4] are:

- *Short Run Emphasis (SRE)*:

$$\mathrm{SRE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i, j, \theta)}{j^2} \tag{1}$$

- *Long Run Emphasis (LRE)*:

$$\mathrm{LRE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i, j, \theta) \cdot j^2 \tag{2}$$

- *Grey-Level Nonuniformity (GLN)*:

$$\mathrm{GLN}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \left(\sum_{j=1}^{N} p(i, j, \theta) \right)^2 \tag{3}$$

- *Run Length Nonuniformity (RLN)*:

$$\mathrm{RLN}(\theta) = \frac{1}{n_r} \sum_{j=1}^{N} \left(\sum_{i=1}^{M} p(i,j,\theta) \right)^2 \tag{4}$$

- *Run Percentage (RP)*:

$$\mathrm{RP} = \frac{n_r}{n_p} \tag{5}$$

where n_r is the total number of runs and n_p is the number of pixels in the image.

Additional investigation into run-length features has prompted additional measures based on the run-length measures. For example, Chu et al. [2] proposed the following two features:

- *Low Grey-Level Run Emphasis (LGRE)*:

$$\mathrm{LGRE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i,j,\theta)}{i^2} \tag{6}$$

- *High Grey-Level Run Emphasis (HGRE)*:

$$\mathrm{HGRE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i,j,\theta) \cdot i^2 \tag{7}$$

Additionally, the work of [3] resulted in four additional run-length features.

- *Short Run Low Grey-Level Emphasis (SRLGE)*:

$$\mathrm{SRLGE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i,j,\theta)}{i^2 \cdot j^2} \tag{8}$$

- *Short Run High Grey-Level Emphasis (SRHGE)*:

$$\mathrm{SRHGE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i,j,\theta) \cdot i^2}{j^2} \tag{9}$$

- *Long Run Low Grey-Level Emphasis (LRLGE)*:

$$\mathrm{LRLGE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{p(i,j,\theta) \cdot j^2}{i^2} \tag{10}$$

- *Long Run High Grey-Level Emphasis (LRHGE)*:

$$\mathrm{LRHGE}(\theta) = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i,j,\theta) \cdot i^2 \cdot j^2 \tag{11}$$

In our implementation, as is typical, the features above are averaged over the neighborhood directions (e.g. in 2-D $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$).

2 Implementation

As mentioned previously, our implementation is based on the following co-occurence feature classes that already exist in the Insight Toolkit:

- `ScalarImageToGreyLevelCooccurrenceMatrixGenerator`,
- `MaskedScalarImageToGreyLevelCooccurrenceMatrixGenerator`,
- `GreyLevelCooccurrenceMatrixTextureCoefficientsCalculator`.

The following analogous classes are included with our submission:

- `ScalarImageToGreyLevelRunLengthMatrixGenerator`,
- `MaskedScalarImageToGreyLevelRunLengthMatrixGenerator`,
- `GreyLevelRunLengthMatrixTextureCoefficientsCalculator`.

After generating the run-length matrix with the class `ScalarImageToGreyLevelRunLengthMatrixGenerator` (if a mask is available then the class `MaskedScalarImageToGreyLevelRunLengthMatrixGenerator` is used), the resulting 2-D histogram is input for the class [**?**].

Instantiation of the matrix generator is performed as demonstrated by the following code snippet:

```
53        typedef itk::Statistics::ScalarImageToGreyLevelRunLengthMatrixGenerator<ImageType>
54            RunLengthMatrixGeneratorType;
55        RunLengthMatrixGeneratorType::Pointer generator = RunLengthMatrixGeneratorType::New();
```

Different parameters are set for the run-length matrix generator

```
56        generator->SetInput( imageReader->GetOutput() );
57        generator->SetNumberOfBinsPerAxis( numberOfBins );
58        generator->SetPixelValueMinMax( 0, 255 );
59        generator->SetDistanceValueMinMax( 0, origin.EuclideanDistanceTo( antiorigin ) );
60        generator->Compute();
```

The calculator is then instantiated and the run-length feature measures are calculated

```
62        typedef itk::Statistics::GreyLevelRunLengthMatrixTextureCoefficientsCalculator
63            <RunLengthMatrixGeneratorType::HistogramType> CalculatorType;
64        CalculatorType::Pointer calculator = CalculatorType::New();
65        calculator->SetHistogram( generator->GetOutput() );
66        calculator->Compute();
67
68        RealType sre = calculator->GetShortRunEmphasis()
69            / static_cast<RealType>( numberOfDirections );
70        RealType lre = calculator->GetLongRunEmphasis()
71            / static_cast<RealType>( numberOfDirections );
72        RealType gln = calculator->GetGreyLevelNonuniformity()
73            / static_cast<RealType>( numberOfDirections );
74        RealType rln = calculator->GetRunLengthNonuniformity()
75            / static_cast<RealType>( numberOfDirections );
76        RealType rp = static_cast<RealType>( calculator->GetTotalNumberOfRuns() )
77            / static_cast<RealType>( numberOfPixelsInMask )
```

	SRE	LRE	GLN	RLN	RP
constant-0	0.00268647	985.799	767	92.1995	0.00292587
constant-255	0.00268647	985.799	767	92.1995	0.00292587
random	0.25	0.25	2689.7	259418	0.989599
1.1.01	0.249999	0.252493	3481.93	252812	0.964405
1.1.02	0.249999	0.252548	4464.78	247518	0.94422
1.1.03	0.249994	0.252554	4714.77	248870	0.949427
1.1.04	0.249999	0.252543	6226.44	247771	0.945178
1.1.05	0.249999	0.252557	6698.78	246384	0.939885
1.1.06	0.249999	0.252528	4656.11	249198	0.950617
1.1.07	0.249999	0.252594	6998.44	243052	0.927178
1.1.08	0.249998	0.252715	10245.8	232406	0.886571
1.1.09	0.249965	0.252889	8650.23	229433	0.875542
1.1.10	0.249999	0.25262	5467.76	240818	0.918661
1.1.11	0.249589	0.29686	7371.13	231024	0.884318
1.1.12	0.249998	0.25261	6674.13	241980	0.923101
1.1.13	0.249998	0.252618	4236.93	241244	0.920293

Table 1: Galloway's original five traditional run-length features [4] applied to the images in Figure 1.

```
78              / static_cast<RealType>( numberOfDirections );
79          RealType lgre = calculator->GetLowGreyLevelRunEmphasis()
80              / static_cast<RealType>( numberOfDirections );
81          RealType hgre = calculator->GetHighGreyLevelRunEmphasis()
82              / static_cast<RealType>( numberOfDirections );
83          RealType srlge = calculator->GetShortRunLowGreyLevelEmphasis()
84              / static_cast<RealType>( numberOfDirections );
85          RealType srhge = calculator->GetShortRunHighGreyLevelEmphasis()
86              / static_cast<RealType>( numberOfDirections );
87          RealType lrlge = calculator->GetLongRunLowGreyLevelEmphasis()
88              / static_cast<RealType>( numberOfDirections );
89          RealType lrhge = calculator->GetLongRunHighGreyLevelEmphasis()
90              / static_cast<RealType>( numberOfDirections );
```

3 Sample Results

Images taken from the USC-SIPI database [1] are provided in Figure 1. The features outlined in the previous section were calculated and provided in Tables 1 and 2.

References

[1] Usc-sipi databse. Texture image database, available at http://sipi.usc.edu/database/, 2008. 3, 1

[2] A. Chu, C. M. Sehgal, and J. F. Greenleaf. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11:415–420, 1990. 1, 2

[3] B. R. Dasarathy and E. B. Holder. Image characterizations based on joint gray-level run-length distributions. *Pattern Recognition Letters*, 12:497–502, 1991. 1, 2

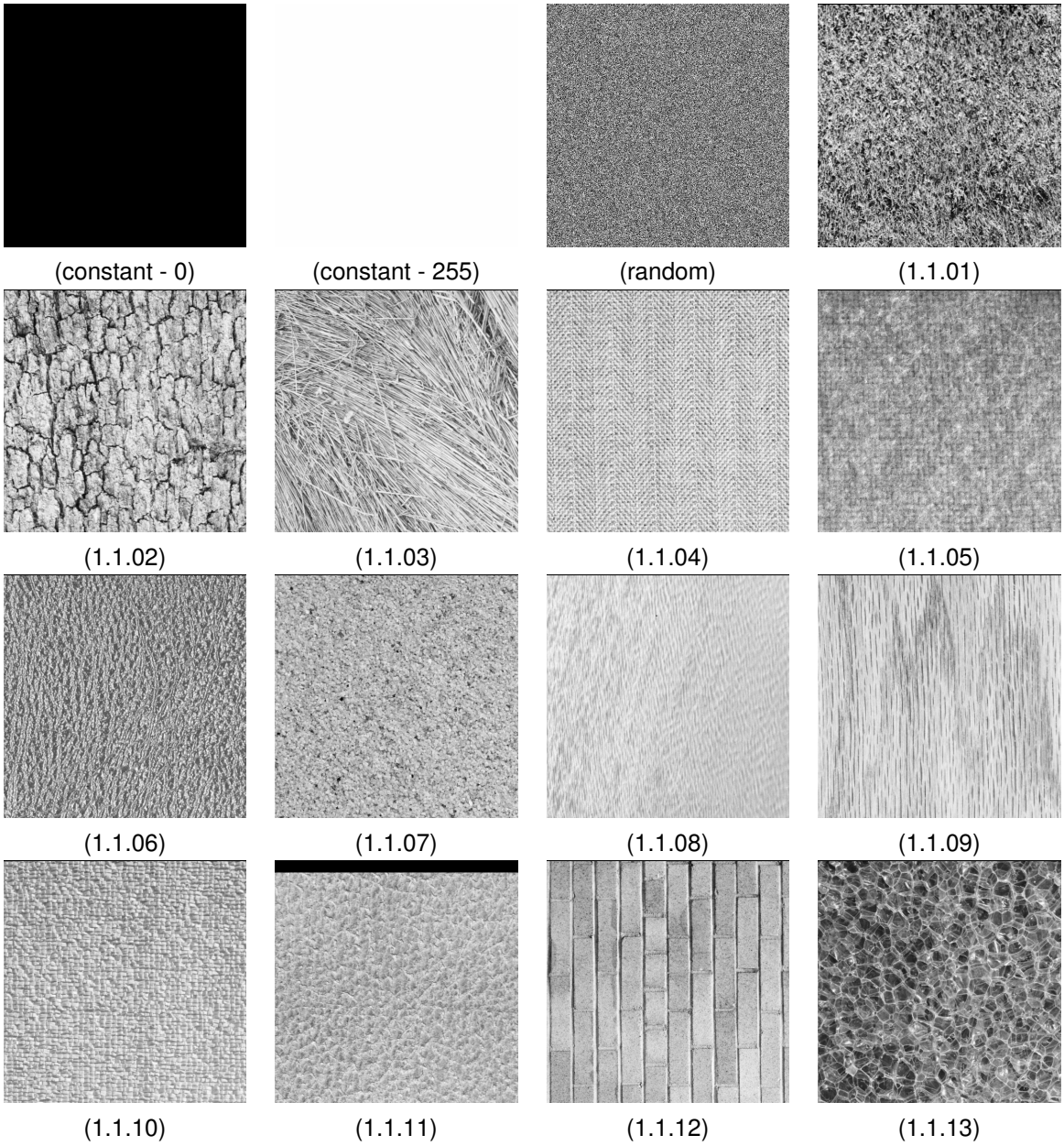

Figure 1: Sample Brodatz textures taken from the USC-SIPI texture database [1].

	LGRE	HGRE	SRLGE	SRHGE	LRLGE	LRHGE
constant-0	0.25	0.25	0.00268647	0.00268647	985.799	985.799
constant-255	2.5e-05	2500	2.68647e-07	26.8647	0.0985799	9.85799e+06
random	0.00459637	848.337	0.00459637	848.337	0.00459637	848.337
1.1.01	0.000728811	801.872	0.000728131	801.872	0.00322151	801.874
1.1.02	0.00053418	1190.33	0.000533668	1190.33	0.00307944	1190.35
1.1.03	0.000455442	1269.42	0.000454939	1269.38	0.00298672	1269.6
1.1.04	0.000435743	1512.61	0.000435238	1512.61	0.0029784	1512.62
1.1.05	0.000460526	1014.66	0.000460018	1014.66	0.0030175	1014.67
1.1.06	0.000481247	942.4	0.000480745	942.4	0.00300935	942.403
1.1.07	0.00046351	1242.89	0.000462996	1242.89	0.00305552	1242.91
1.1.08	0.000452226	1677.24	0.000451688	1677.23	0.00316296	1677.27
1.1.09	0.000464228	1550.92	0.000463679	1550.64	0.00320913	1552.08
1.1.10	0.000453884	1322.02	0.000453365	1322.02	0.00306993	1322.05
1.1.11	0.000487384	1274.95	7.65921e-05	1274.95	0.0473478	1274.99
1.1.12	0.000463563	1271.63	0.000463047	1271.62	0.00306703	1271.69
1.1.13	0.000565068	650.971	0.000564545	650.97	0.0031765	650.976

Table 2: Additional run-length matrix features proposed in [2] and those proposed in [3] applied to the images in Figure 1.

[4] M. M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975. 1, 1