New Texture Similarity Metrics

David L. Neuhoff

University of Michigan

CSP Seminar

Oct 23, 2014

Collaborators:

UM: Yuanhao Zhai;

Northwestern: Thrasos Pappas, Jana Zujovic,

Xiaonan Zhao, Guoxin Jin

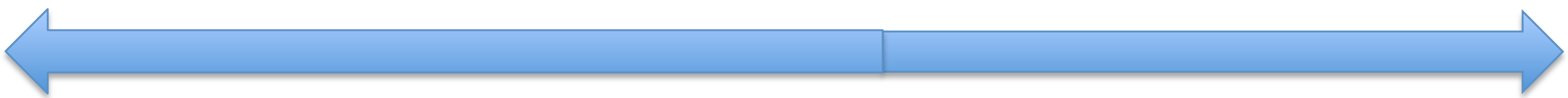
My Interests in Perceptual Similarity Measures

- Going way back
  - Lossy source coding (data compression)
  - Audio coding for digital jukebox
  - Image coding for FAX transmission
  - Halftoning
- Recent
  - Compressing images containing texture
  - Textured image retrieval
  - Texture classification

Research Spectrum: Analytical to Qualitative

much of my work

this work



analytical

qualitative

Precise mathematical formulations
Idealized, tractable models
Idealized, tractable performance measures
Definitions, derivations, theorems, proofs
Provably good methods/algorithms.
Clear understanding of how/why things work
Limited practical relevance

Rough formulations
Vague, qualitative models
Actual real-world data
Vague definitions
Intractable performance measures
Ad Hoc algorithms
Proof is in the pudding; success is in the eye of the beholder
Basis of success/failure not so clear
Direct practical relevance

Uses of Image Metrics

- Assess performance of image processing methods/systems.
  - Eliminate burden of human subjective assessments.
- Enable optimization (rare outside of MSE based metrics)
- Serve as a component of an image processing system
  - Restoration
  - Content based image retrieval
  - Classification
  - Lossy compression/coding
  - Question: If a metric is used as a system component, can it also be used to judge the performance of the system?

This talk

- Some metrics symmetrically assess similarity, i.e., $S(X,Y) = S(Y,X)$.
- Some (asymmetric) metrics assess quality of Y as reproduction of X.
- "No reference metrics" assess quality of Y all by itself.
- Almost none are "metrics" in the mathematical sense.

This talk

Goals of Image Similarity Metrics

- Reflect human perception/judgments
  - produce values monotonically related to human scoring
- Low computational and/or conceptual complexity
- Analytical tractability to permit analytical optimization (rare)
- Make system that uses it as a component work well
- Specialize to individual applications
- Possible invariance properties to have or not
  - rotation
  - spatial scaling
  - amplitude scaling (contrast)
  - amplitude shifting

This talk

Testing/Assessing Image Metrics

- Correlate metric values with subjective assessments by people
- Assess the performance of a system that uses the metric
  - Retrieval
  - Compression
  - Classification
  - Restoration

This talk

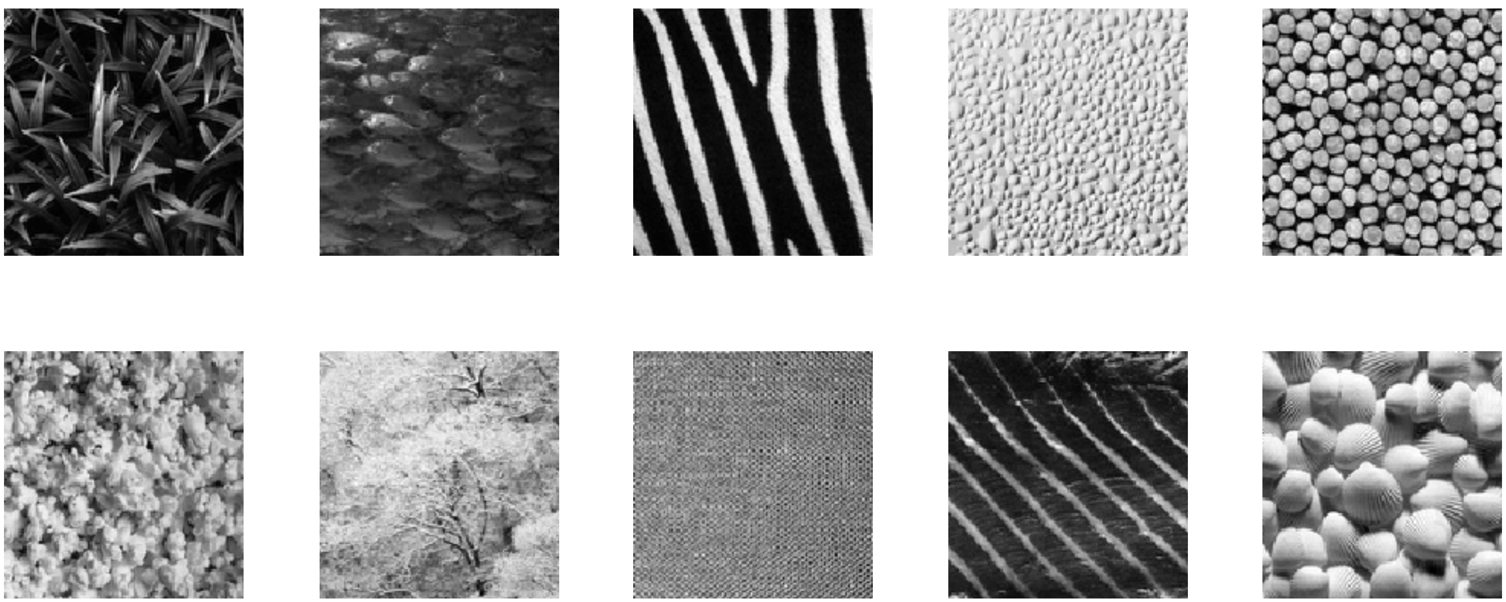
Our Goal: Similarity Metrics for Texture

- What is texture?

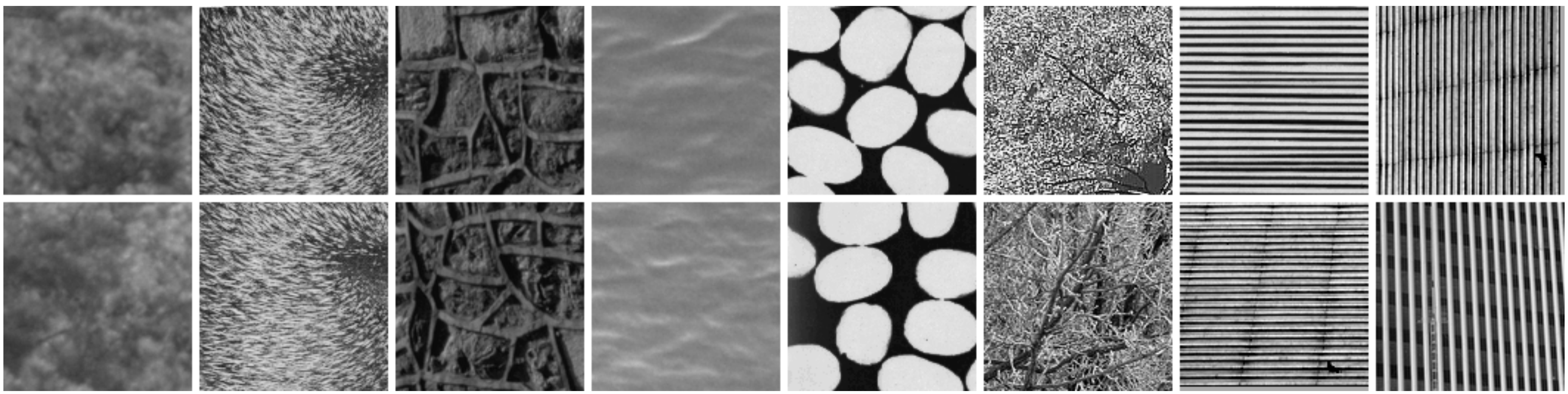
  "Loosely speaking, texture images are spatially homogeneous and consist of repeated elements, often subject to some randomization in their location, size, color, orientation, etc."

  [Portilla, Simoncelli, *Int. J. Comput. Vis.*, Oct. 2000]

- Textures are images for which point-by-point metrics like MSE are least relevant. (MSE can be very large for "identical" textures.)



Examples of “Similar Textures” that are not similar in MSE



Examples of "Similar Textures" that are not similar in MSE

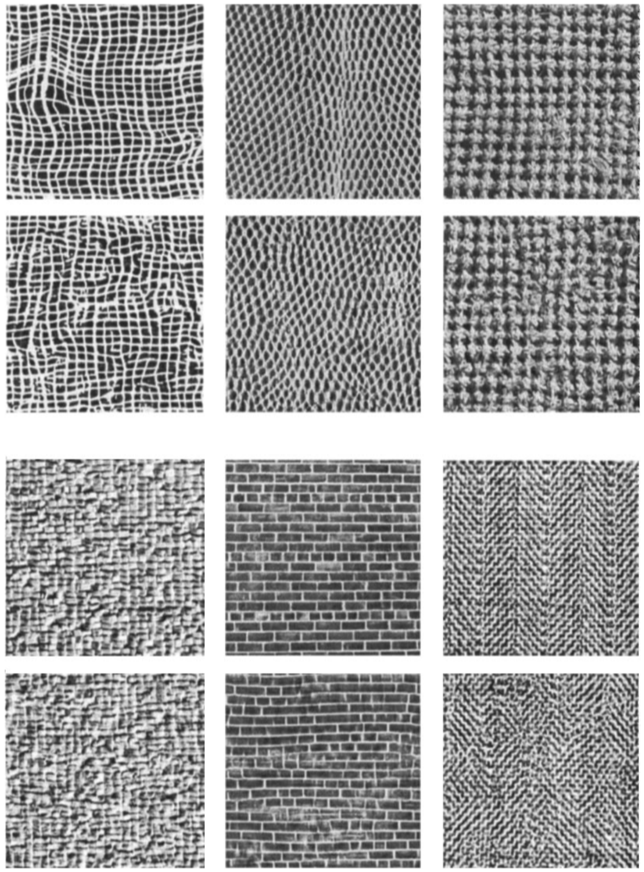


Figure 14. Synthesis results on photographic pseudo-periodic textures. See caption of Fig. 12.

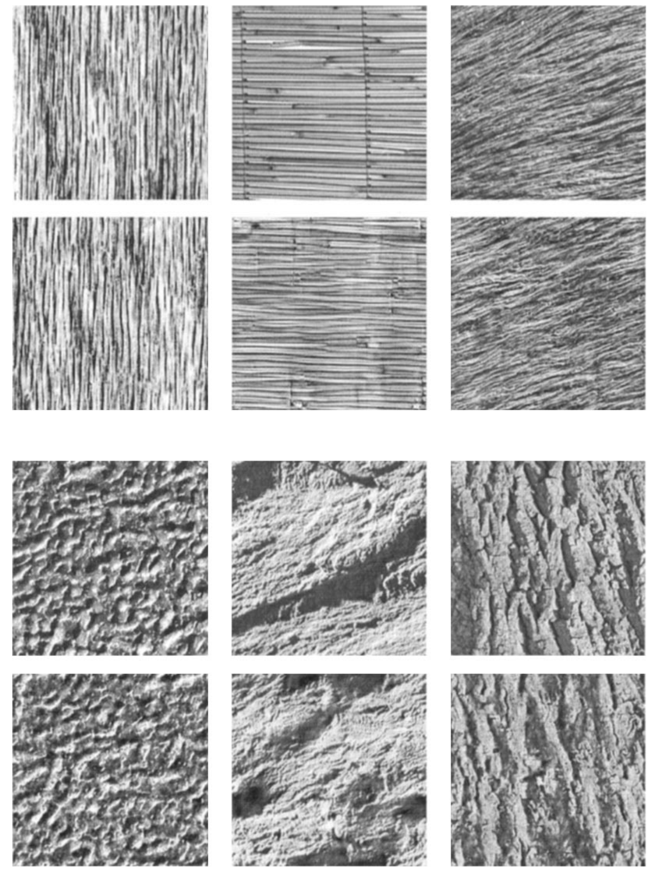


Figure 15. Synthesis results on photographic aperiodic textures. See caption of Fig. 12.

Image Metric Themes

- Much recent interest in metrics of the following types
  - Structural Similarity Image Metrics
    - SSIM: [Wang, Bovik, Sheikh, Simoncelli, *IEEE Tr. Im. Proc.*, 2004]
  - Local Pattern Metrics for texture similarity (my term)
    - Local Binary Patterns (LBP): [Ojala, Pietikainen,Maenpaa, *IEEE Tr. PAMI*, 2002]
  - Both of these are, largely, sliding statistic/feature based, instead of point-by-point comparison based
  - Both are applicable to textures

- This talk: Two new metrics of texture similarity, one of each type
  - STSIM: Structural Texture Similarity Metric
  - LRI: Local Radius Index

Outline

- Review
  - Point-by-point metrics
  - SSIM (Structural Similarity Metric)
- STSIM (Structural Texture Similarity Metric): new SSIM type metric:
  - Performance in identical texture retrieval and image compression
- Review
  - LBP (Local Binary Patterns)
- LRI (Local Radius Index): new local-pattern metric
  - Performance in identical texture retrieval and image coding
- RI-LRI (Rotation invariant LRI):
  - Performance in texture classification
- Concluding remarks

Review: Point-by-Point Metrics

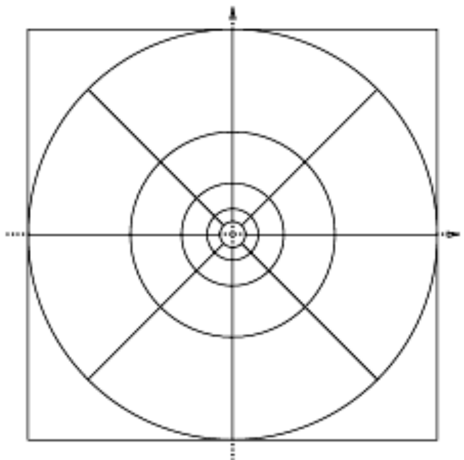
- Given N x N image X and reproduction Y
- MSE: $$\text{MSE}(X,Y) = \frac{1}{N^2}\sum_i \left(X_i - Y_i\right)^2$$
- Frequency weighted MSE: $$\text{FW-MSE}(X,Y) = \frac{1}{N^2}\sum_j w_j\left(\overline{X}_j - \overline{Y}_j\right)^2$$

$\overline{X}$ and $\overline{Y}$ are coefficients in a transform or subband decomposition

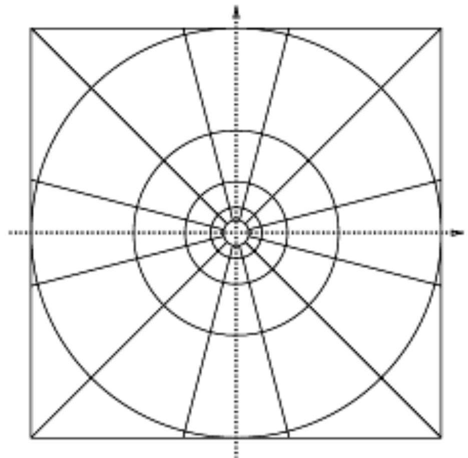
Weight w_j can depend on sensitivity of visual perception to changes in coef. $\overline{X}_j$ and the value of $\overline{X}_j$ and neighboring $\overline{X}_j$'s, i.e. w_j can reflect masking effects.

Some Frequency Decompositions

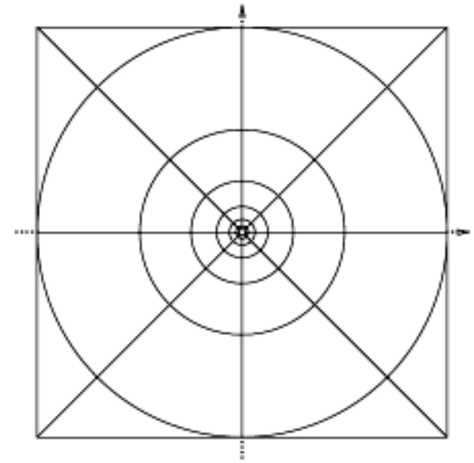
Cortex (Watson'87)



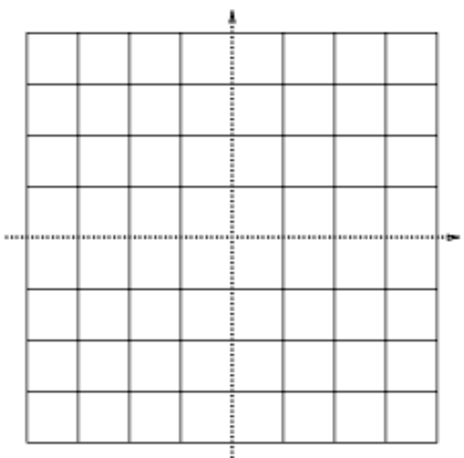
Cortex (Daly'92)



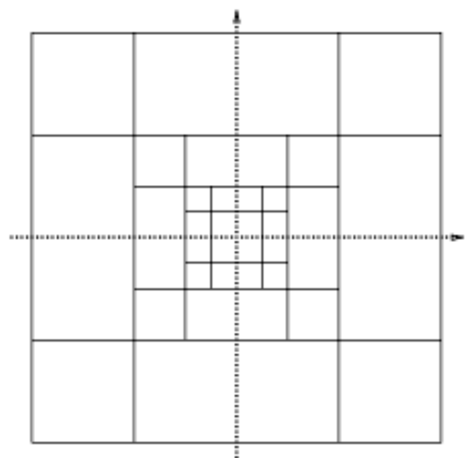
Lubin'91



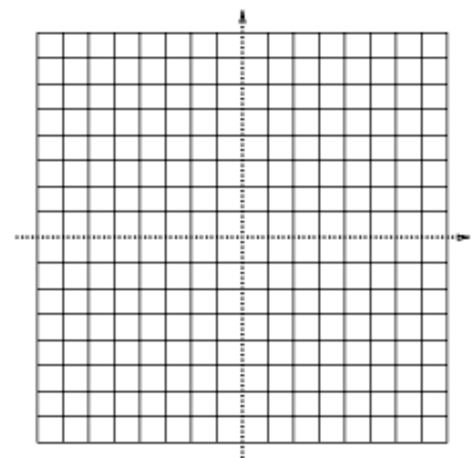
4x4 Subband



Wavelet



8x8 DCT



Review: The Original SSIM[1]

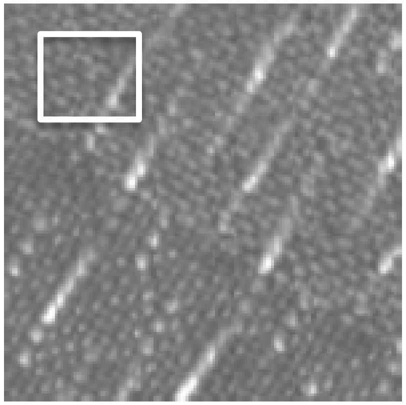
- Given images X and Y, to compute SSIM(X, Y):
  - Slide NxN window across each in steps of s, e.g. $N = 8$ to 32, $s = N/2$
  - For n-th window position, compute:
    - *luminances* (means) $m_{X,n} = \frac{1}{N^2}\sum_i X_{n,i}$ $\quad m_{Y,n} = \frac{1}{N^2}\sum_i Y_{n,i}$
    - *contrasts* (variances) $c^2_{X,n} = \frac{1}{N^2}\sum_n \left(X_{n,i} - m_{X,n}\right)^2$ $\quad c^2_{Y,n} = \frac{1}{N^2}\sum_n \left(Y_{n,i} - m_{Y,n}\right)^2$
    - statistic similarities: $M_n = \frac{2\, m_{X,n}\, m_{Y,n} + \varepsilon}{m^2_{X,n} + m^2_{Y,n} + \varepsilon}$ $\quad C_n = \frac{2\, c_{X,n}\, c_{Y,n} + \varepsilon}{c^2_{X,n} + c^2_{Y,n} + \varepsilon}$

      $\frac{\text{geom. mean}}{\text{arith. mean}}$
    - *structure* (crosscorrelation): $S_n = \dfrac{\frac{1}{N^2}\sum_i \left(X_{n,i} - m_{X,n}\right)\left(Y_{n,i} - m_{Y,n}\right) + \varepsilon}{c_{X,n}\, c_{Y,n} + \varepsilon}$
    - window similarity: $W_n = \left(M_n\, C_n\, S_n\right)^{1/3}$
  - Pool window similarities over all positions: $\mathrm{SSIM}(X,Y) = \frac{1}{\#\,\text{positions}}\sum_n W_n$
  - $0 \leq \mathrm{SSIM} \leq 1$. 1 = highest quality, 0 = lowest

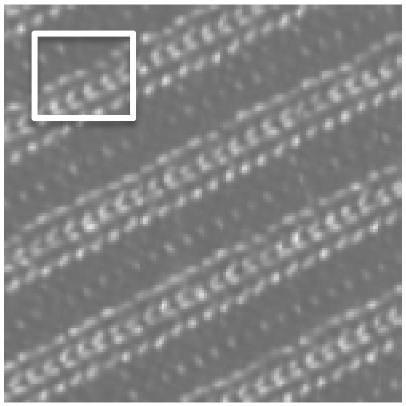
[1]Wang, Bovik, Sheikh, Simoncelli, *IEEE Tr. Im. Proc.*, 2004

Review: The Original SSIM[1]

- Given images X and Y, to compute SSIM(X, Y):
  - Slide NxN window across each in steps of s, e.g. N = 8 to 32, $s = N/2$
  - For n-th window position, compute:
    - *luminances* (means) $$m_{X,n} = \frac{1}{N^2}\sum_i X_{n,i} \qquad m_{Y,n} = \frac{1}{N^2}\sum_i Y_{n,i}$$
    - *contrasts* (variances) $$c^2_{X,n} = \frac{1}{N^2}\sum_n \left(X_{n,i} - m_{X,n}\right)^2 \qquad c^2_{Y,n} = \frac{1}{N^2}\sum_n \left(Y_{n,i} - m_{Y,n}\right)^2$$
    - statistic similarities: $$M_n = \frac{2\, m_{X,n}\, m_{Y,n} + \varepsilon}{m^2_{X,n} + m^2_{Y,n} + \varepsilon} \qquad C_n = \frac{2\, c_{X,n}\, c_{Y,n} + \varepsilon}{c^2_{X,n} + c^2_{Y,n} + \varepsilon}$$
      geom. mean / arith. mean
    - *structure* (crossc X $$\frac{\frac{1}{N^2}\sum_i \left(X_{n,i} - m_{X,n}\right)\left(Y_{n,i} - m_{Y,n}\right) + \varepsilon}{\quad}$$



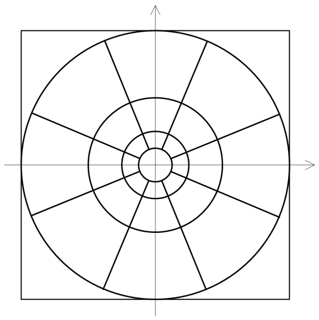
    - window similarity: $S_n)^{1/3}$
  - Pool window similar s: S — ns $\sum_n W_n$
  - $0 \leq$ SSIM ≤ 1. 1 = lowest



WHY?

[1] Wang, Bovik, Sheikh, Simoncelli, *IEEE Tr. Im. Proc.*, 2004

SSIM Improvements



- CW-SSIM (Complex Wavelet SSIM)[1]
  - Apply SSIM (without luminance) to 7x7 windows of each subband of *steerable pyramid* wavelet/subband decomposition,
  - For example, 12 bands, 3 scales, 4 orientations, no lowpass or highpass bands.
  - Magnitude of the structure/crosscorrelation terms
  - Pool over subbands.
  - Advantage: Does not penalize small spatial shifts, rotations, scalings.
- Many other improved versions of SSIM

[1]Sampat, Wang, Gupta,. Bovik & Markey, *IEEE Tr. Im. Proc.*, Nov. 2009

Outline

- Review
  - Point-by-point metrics
  - SSIM (Structural Similarity Metric)
- STSIM (Structural Texture Similarity Metric): new SSIM type metric:
  - Performance in identical texture retrieval and image compression
- Review
  - LBP (Local Binary Patterns)
- LRI (Local Radius Index): new local-… c
  - Performance in identical texture retri… co
- RI-LRI (Rotation invariant LRI):
  - Performance in texture classification
- Concluding remarks

Thrasos Pappas

Jana Zujovic

STSIM for Textured Images

- Structural Texture Similarity Metric (STSIM)[1]
  - Start with CW-SSIM (based on steerable pyramid decomposition)
  - Discard "structure" terms –-- they are not even "statistics"
  - Add lag 1 autocorrelations as statistics

horizontal

$$\rho_{H,n} = \frac{\frac{1}{N^2}\sum_{i,j}\left(X_{n,i,j} - m_n\right)\left(X_{n,i,j+1} - m_n\right)^*}{c_{X,n}^2}$$

vertical

$$\rho_{V,n} = \frac{\frac{1}{N^2}\sum_{i,j}\left(X_{n,i,j} - m_n\right)\left(X_{n,i+1,j} - m_n\right)^*}{c_{X,n}^2}$$

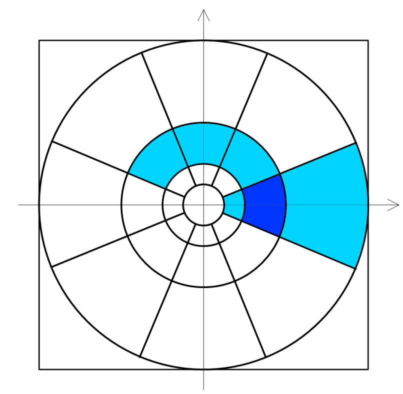
  - Autocorrelation similarity: $R_{H,n} = 1 - \frac{1}{2}\left|\rho_{X,H,n} - \rho_{Y,H,n}\right|$
  - 56 statistics total
  - Geometric mean pooling of statistic similarity for each window position
  - Arithmetic average pooling over window positions
  - Or, for homogeneous textures, treat entire image as one window

[1]Zhao, Reyes, Pappas, Neuhoff, ICIP, Oct. 2008

STSIM2[1] for Textured Images

- To STSIM, add
  - Add interband crosscorrelations (of magnitudes) between bands of pyramid decomposition of image *X*. Same for *Y*.

$$l_n^{(k,l)} = \frac{\frac{1}{N^2}\sum_i \left(\left|X_{n,i}^{(k)}\right| - m_n^{(k)}\right)\left(\left|X_{n,i}^{(l)}\right| - m_n^{(l)}\right)}{c_{X,n}^{(k)}\, c_{X,n}^{(l)}}$$



  - Between
    - all pairs of bands within each scale
    - bands at adjacent scales with same orientation
  - Same similarity scoring as for autocorrelations
  - 82 statistics total
  - Geometric mean pooling of statistic similarity for each window position
  - Arithmetic average pooling over window positions
  - Or, for homogeneous textures, treat entire image as one window

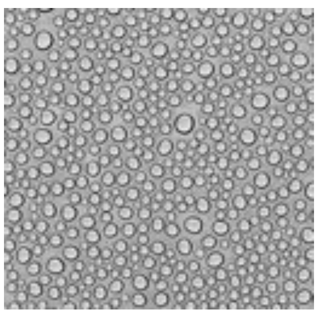
[1] Zujovic, Pappas, Neuhoff, *IEEE Tr. Im. Proc.*, 2013.

Metric Test: Identical Texture Retrieval[1]

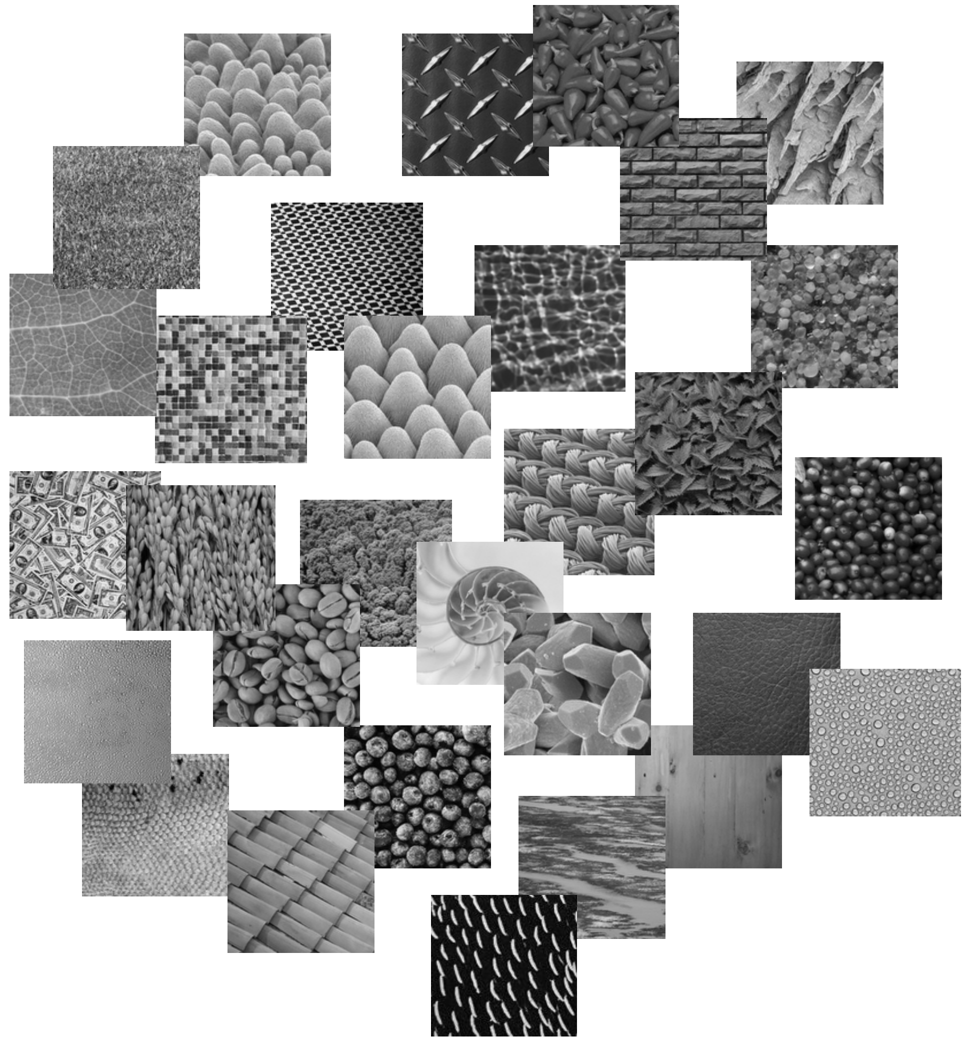
- Test metric as component of a retrieval system.
  - Given database of textures partitioned into classes of "identical textures", find closest texture in database to query image.
- Goodness of metric:
  - "Precision@1" = % of queries for which closest image is in same class.
- No subjective evaluations needed!

[1]Zujovic, Pappas, Neuhoff, *IEEE Tr. Im. Proc.,* 2013

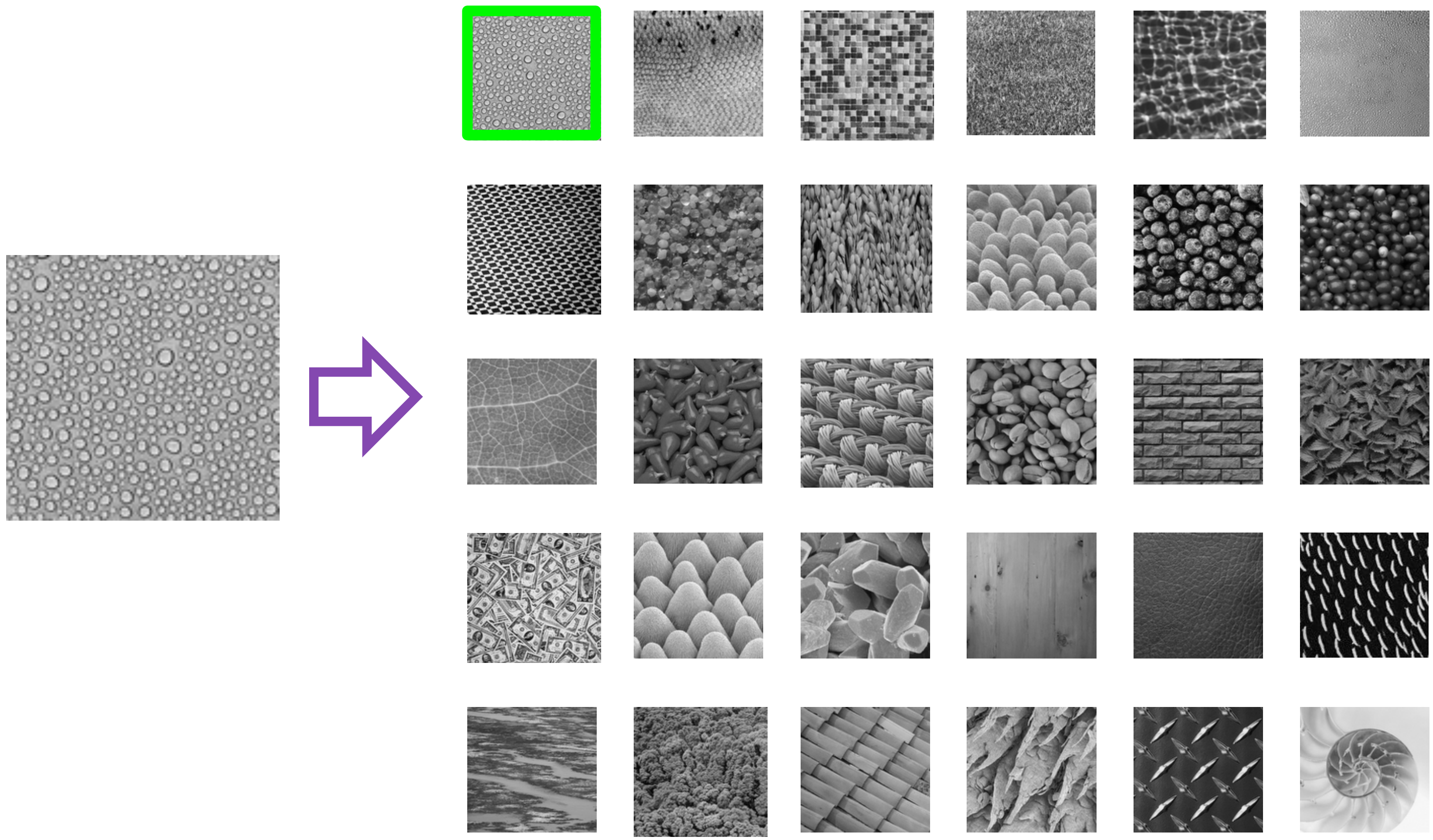
Identical Texture Retrieval:
Find Closest Image to Query in Database



?

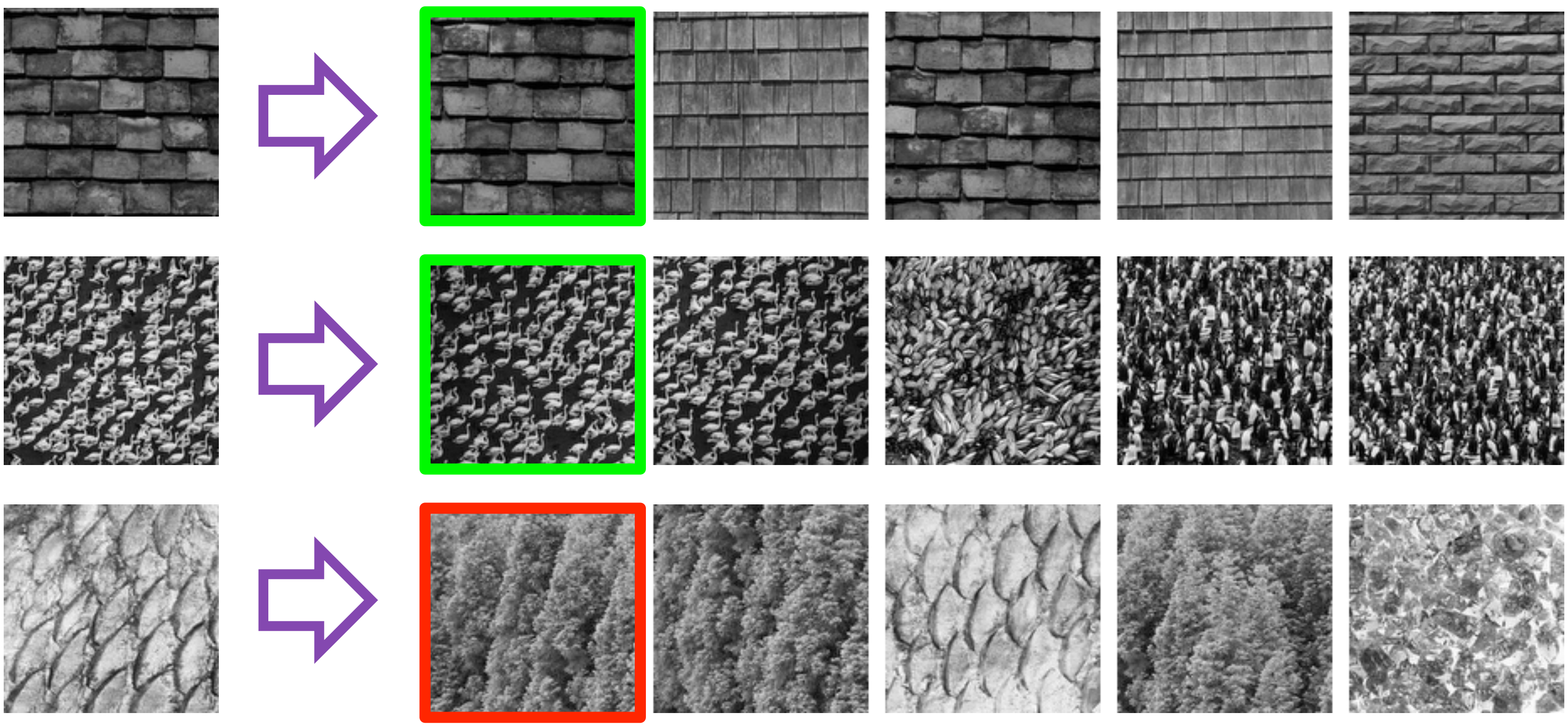


Order Database by Metric Similarity



Precision @ One

- Measures how many times the first retrieved texture was the correct one



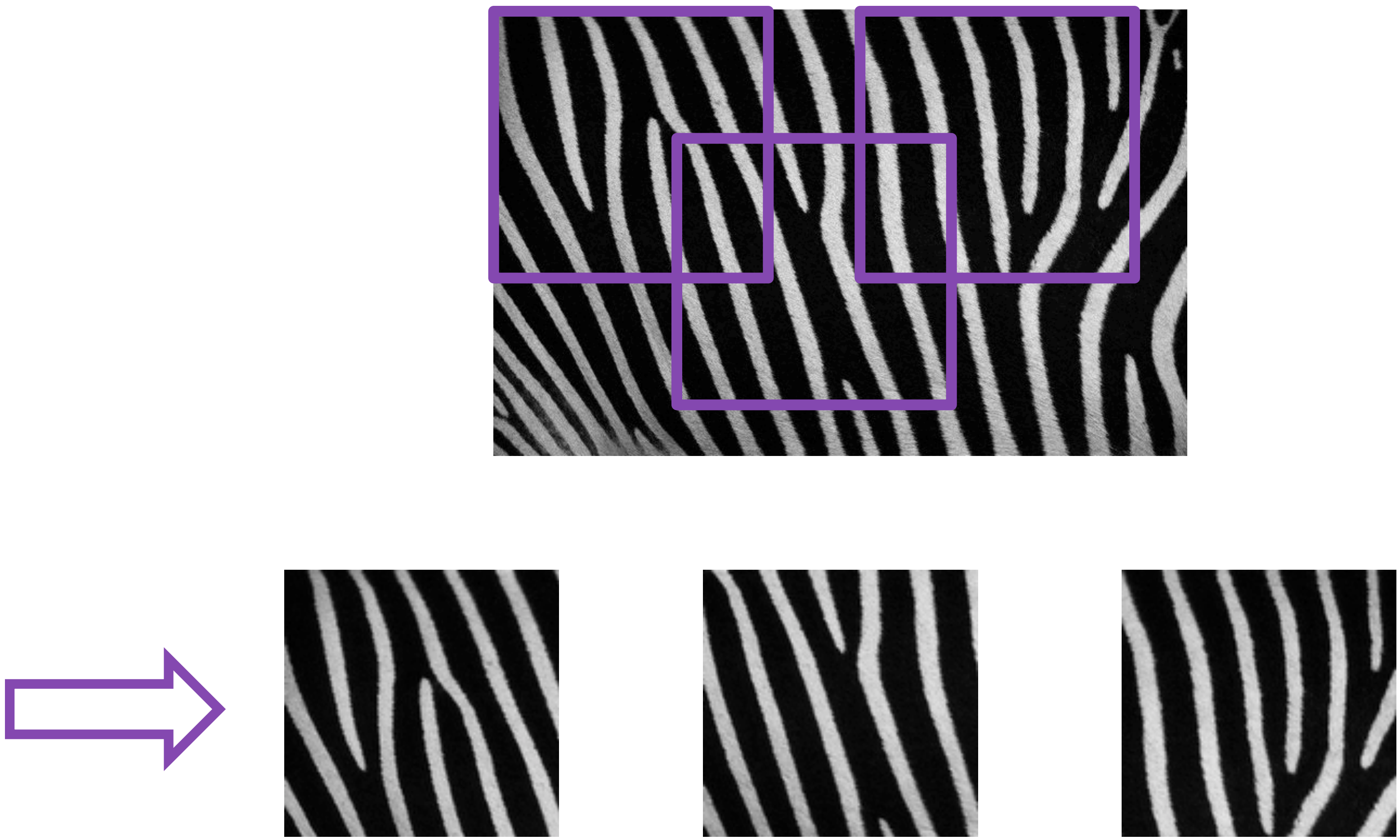
Metric Test: Identical Texture Retrieval[1]

- Test metric as component of a retrieval system.
  - Given database of textures partitioned into classes of “identical textures”, find closest texture in database to query image.
- Goodness of metric:
  - “Precision@1” = % of queries for which closest image is in same class.
- No subjective evaluations needed!
- Corbis-based database: 1181 textures in 425 “identical” classes.

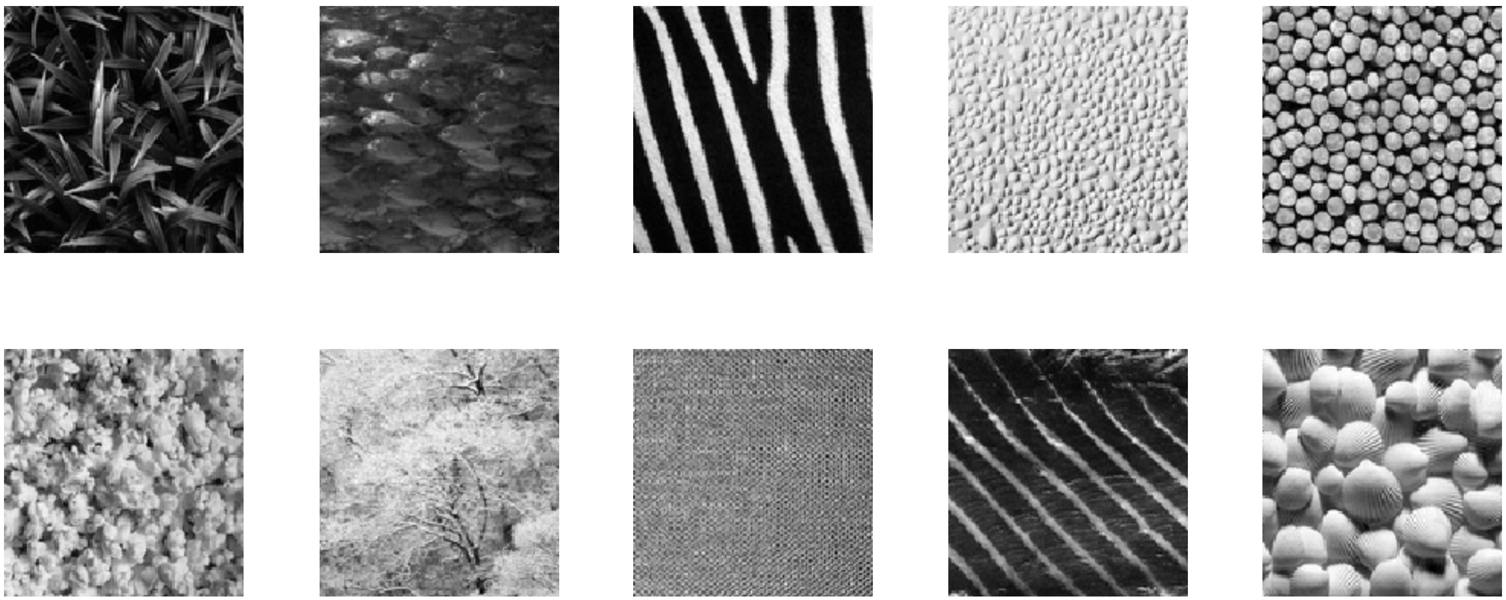
  Each class formed by extracting several 128x128 textures from one homogeneous larger texture image.

[1]Zujovic, Pappas, Neuhoff, *IEEE Tr. Im. Proc.*, 2013

Building the Database



Sample Images in Corbis Database



Metric Test: Identical Texture Retrieval[1]

- Test metric as component of a retrieval system.
  - Given database of textures partitioned into classes of "identical textures", find closest texture in database to query image.
- Goodness of metric:
  - "Precision@1" = % of queries for which closest image is in same class.
- No subjective evaluations needed!
- Corbis-based database: 1181 textures in 425 "identical" classes.

  Each class formed by extracting several 128x128 textures from one homogeneous larger texture image.
- Results

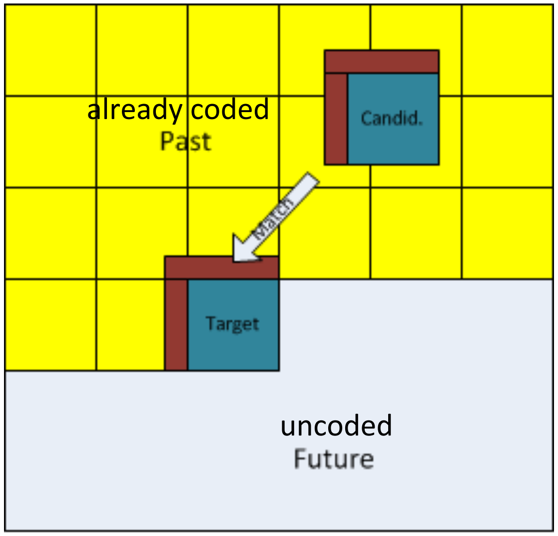
PSNR	SSIM	CW-SSIM	CW-SSIM global	STSIM	STSIM global	STSIM 2	STSIM2 global	STSIM-M global
4%	9%	39%	27%	74%	86%	74%	93%	96%

- Similar results for other performance measures that take into account location of other siblings in the ordering of textures according to distance from query.

[1]Zujovic, Pappas, Neuhoff, *IEEE Tr. Im. Proc.*, 2013

Another Test: Matched Texture Coding (MTC)[1]

- Consider encoding current block, e.g., 32x32, by pointing to a previously encoded block.
- If no sufficiently good previous block, encode with JPEG.
- Key to success is a texture similarity metric for judging if a candidate previous block is sufficiently good.
- Other issues:
  - Search strategy
  - Blending around blocks to avoid blocking artifacts.



- Notes:
  - Textured blocks require the most bits when coded with a pixel-accurate criteria.
  - Whereas the eye is maximally forgiving to changes in a textured block.
  - Goal of MTC is "structurally lossless", rather than "perceptually lossless": Decoded image might look different viewed next to original, but has "equal quality".

- [1]Jin, Zhai, Pappas, Neuhoff, ICIP, 2012.

Encoding at 0.34 bits/pixel

MTC

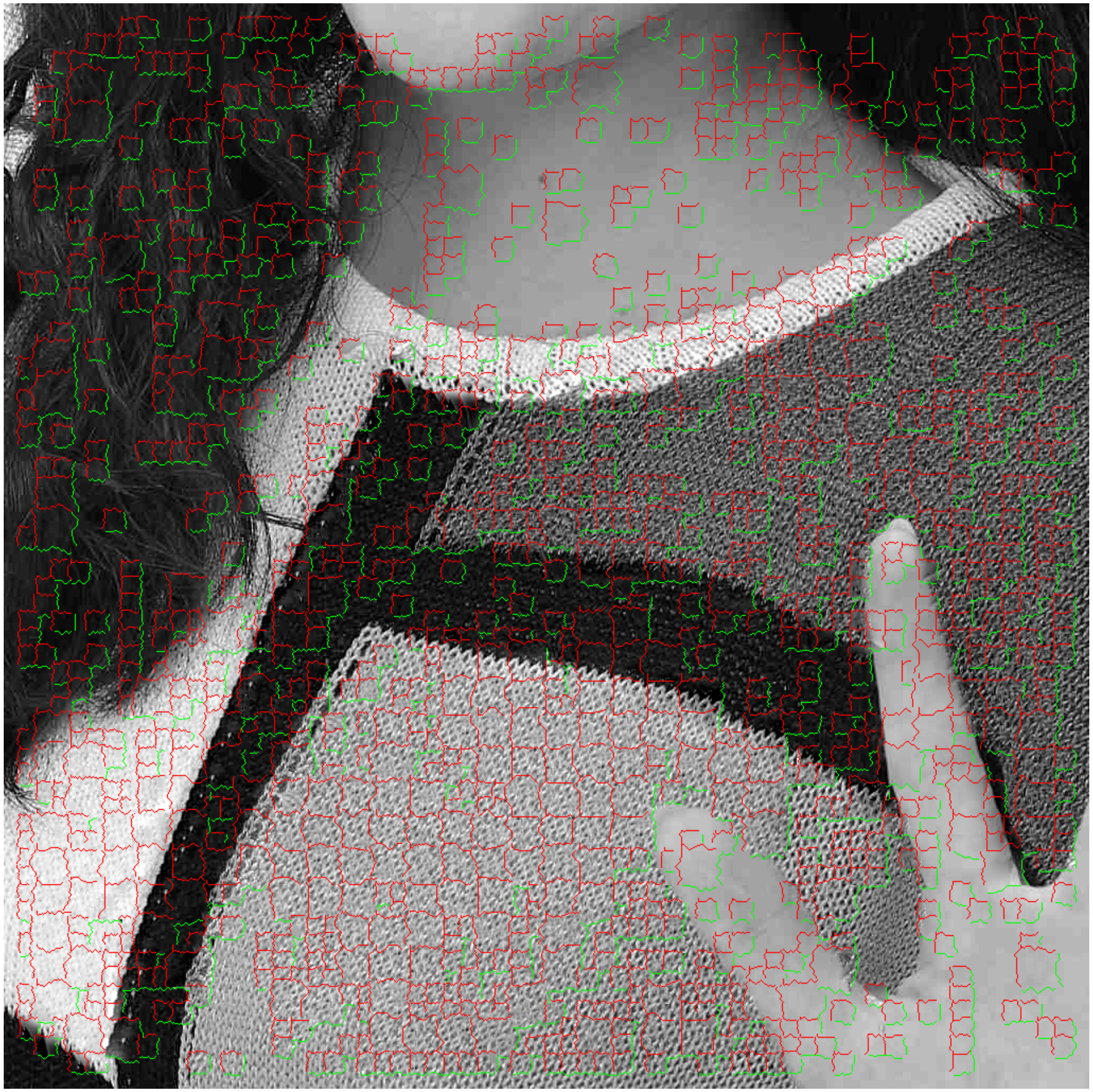
JPEG

Structurally Lossless?

original

MTC

MTC @ 0.37 bpp



blending seams of matched-texture blocks

- red – matched-texture block to right or below
- green - blending seam between texture block and JPEG

MTC @ 0.37 BPP

	32x32 matched-texture	16x16 matched-texture	8x8 JPEG	Overall
fraction of image coded	20.5%	23.6%	55.9%	100%
coding rate within block, bpp	0.03	0.11	0.61	
contribution to total, bpp	0.006	0.026	0.34	0.37

- Recall key ideas:
  - Matched-texture coding saves bits,
  - And saved bits can be used to improve JPEG coding
  - Which in turn leads to better matched-texture coding, by improving quality of available candidates

Outline

- Review
  - Point-by-point metrics
  - SSIM (Structural Similarity Metric)
- STSIM (Structural Texture Similarity Metric): new SSIM type metric:
  - Performance in identical texture retrieval and image compression
- Review
  - LBP (Local Binary Patterns)
- LRI (Local Radius Index): new local-pattern metric
  - Performance in identical texture retrieval and image coding
- RI-LRI (Rotation invariant LRI):
  - Performance in texture classification
- Concluding remarks

LBP (Local Binary Patterns)[1]

- Another approach to computing sliding-window statistics
- Intended for texture classification, where rotation invariance is required. (In contrast for identical texture retrieval and MTC coding, rotations should be penalized monotonically.)

- For each image, compute LBP Index at each pixel,
- LBP Feature = histogram H of LBP Indices.
- LBP Metric

LBP(X, Y) = similarity of histograms H_X and H_Y as measured by divergence of H_X wrt H_Y

$$= \sum_i H_Y(i) \log \frac{H_Y(i)}{H_X(i)}$$

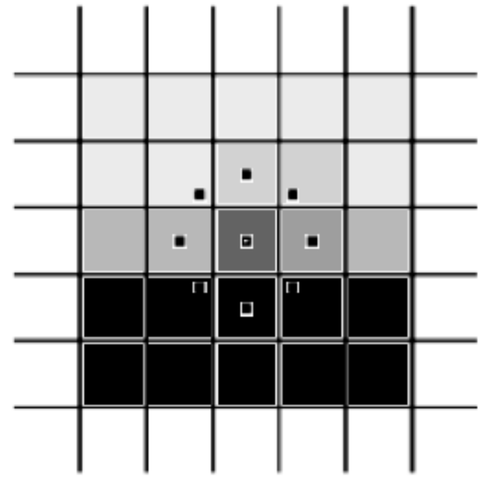
[1] Ojala, Pietikainen, Maenpaa, *IEEE Tr. PAMI*, 2002

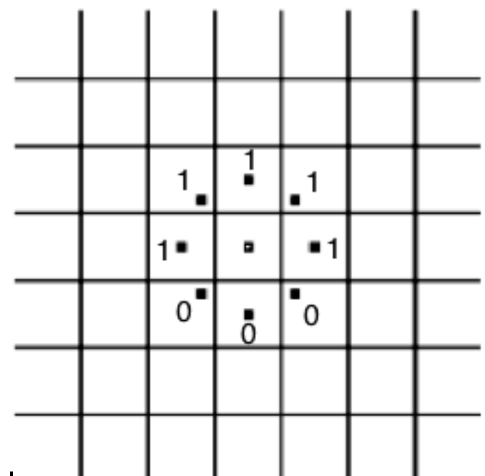
LBP Indices and Feature for Image *X*

- Parameters *r*, *P*.
  - For pixel x_i , let $y_1,\ldots,y_P$ denote image values at *P* equally spaced points on circle of radius *r* centered at *i*-th pixel. Interpolate as needed.
  - Form *P*-dimen'l binary pattern vector $b_1,\ldots,b_P$ with
    $$b_j = \begin{cases} 1, & \text{if } y_j > x_i \\ 0, & \text{else} \end{cases}$$
  - Circularly shift $b_1,\ldots,b_P$ to maximize # leading 0's.
  - Assign integers 0,…,*P* to *uniform* patterns:

00000000	00000001	00000011	…	11111111
0	1	2		P

  - Assign P+1 to all other patterns.
  - LBP index at pixel i = assigned integer
  - LBP index provides circular pattern information
  - Rotation invariant, and gray-scale invariant
  - LBP Feature = histogram of LBP indices for all pixels, which provides frequencies of circular patterns.





Outline

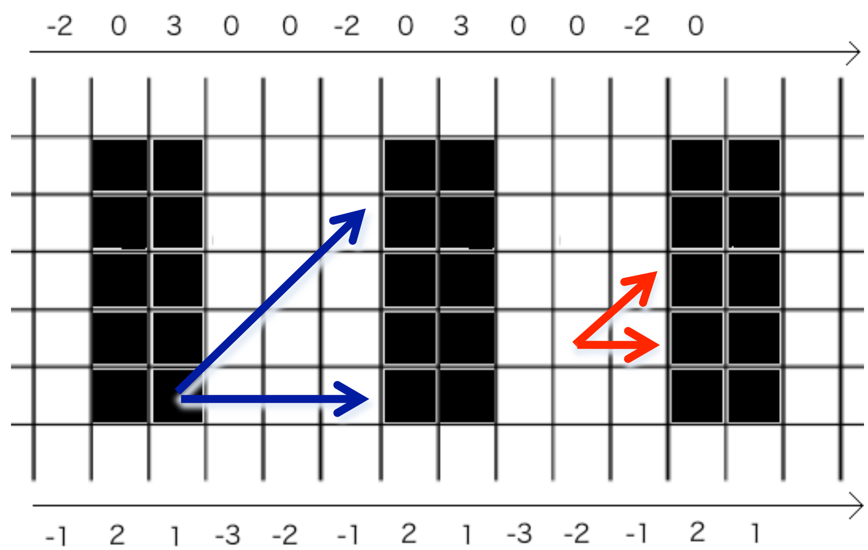
- Review
  - Point-by-point metrics
  - SSIM (Structural Similarity Metric)
- STSIM (Structural Texture Similarity Metric): new SSIM type metric:
  - Performance in identical texture retrieval and image compression
- Review
  - LBP (Local Binary Patterns)
- LRI (Local Radius Index): new local-pattern metric
  - Performance in identical texture retrieval and image coding
- RI-LRI (Rotation invariant LRI):
  - Performance in texture classification
- Concluding remarks

Yuanhao Zhai

New Feature: Local Radius Index (LRI)[1]

- Basic ideas:
  - Generally speaking, a texture contains repetitive smooth elements delineated by edges.
  - One meaningful characteristic of a homogeneous texture is the distribution of distances between adjacent edges at some particular angle.
  - LRI produces 8 such histograms -- one for each neighboring direction.

- Two types of LRI:
  - LRI-A: measures widths of Adjacent smooth regions, i.e. inter-edge distances, in each given direction.
  - LRI-D: measures Distances from pixels to nearest edge, i.e., to boundary of next smooth region, in each given direction.



[1]Zhai, Neuhoff, Pappas, ICASSP, 2013.

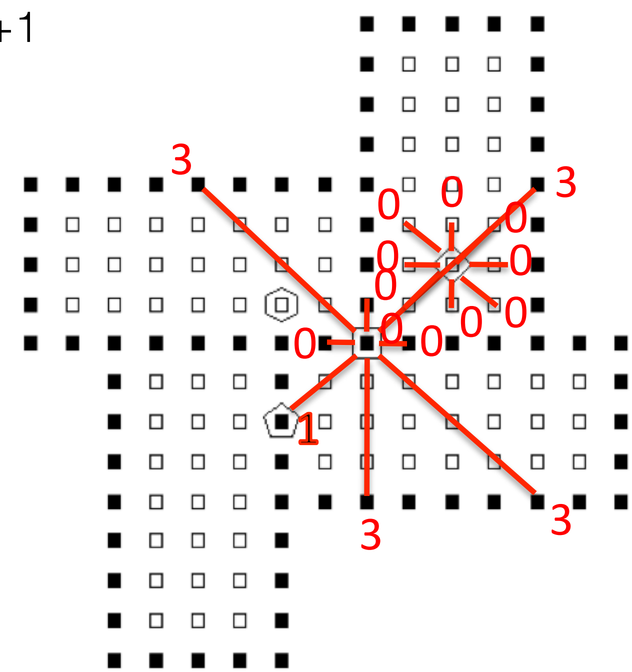
LRI-A

- LRI-A Indices

For i-th pixel x_i and directions $d = 1,\ldots,8$,
let $a_1, a_2, a_3, \ldots$ denote successive Adjacent pixels in direction d.

$$
\begin{aligned}
L_d &= 0\,, \quad \text{if } |x_i - a_1| \le \text{threshold } T \\
&= k\,, \quad \text{if } a_j > x_i + T \text{ for } j = 1,\ldots,k, \text{ but not } k{+}1 \\
&= -k\,, \quad \text{if } a_j < x_i - T \text{ for } j = 1,\ldots,k, \text{ but not } k{+}1
\end{aligned}
$$

Limit max value of L to K and $-K$, respectively.



- LRI-A Feature for image X:

Histograms: $H_1,\ldots, H_8$ for indices $L_1,\ldots,L_8$,

- Threshold T: Determines what is an edge, and controls noise sensitivity. Typically, $T = \sigma/2$.

- Size limit K: Limits size of texture elements detected by LRI, and reduces computation. Typically, $K = 4$.

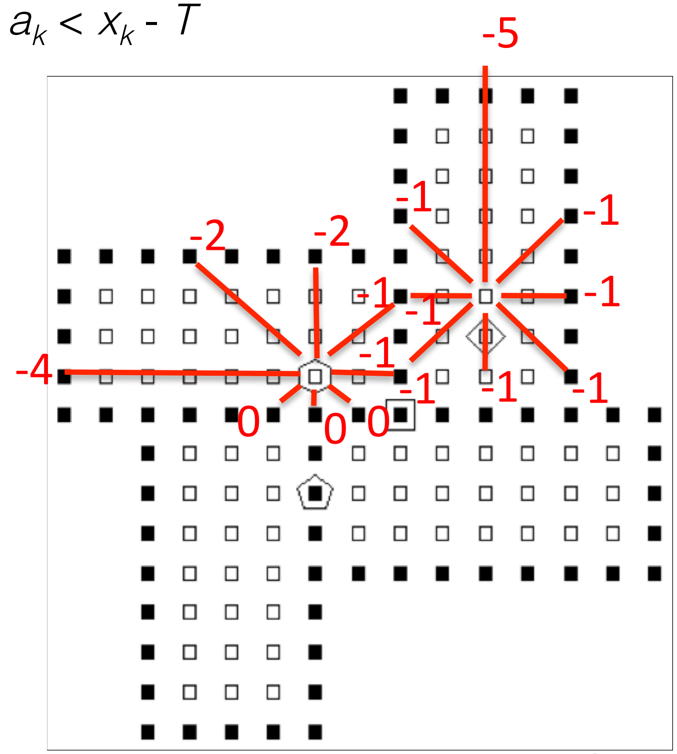
LRI-D

- LRI-D Indices

  For i-th pixel x_i and directions $d = 1,\ldots,8$,
  let $a_1, a_2, a_3, \ldots$ denote successive adjacent pixels in direction d

$$
\begin{aligned}
L_d &= 0, \quad \text{if } |x_i - a_1| > T, \\
&= k, \quad \text{if } |x_i - a_j| \le T, \text{ for } j = 1,\ldots,k\text{-}1 \text{ and } a_k > x_k + T \\
&= -k, \quad \text{if } |x_i - a_j| \le T, \text{ for } j = 1,\ldots,k\text{-}1 \text{ and } a_k < x_k - T
\end{aligned}
$$

  Limit max value of L to K and $-K$, respectively.



- LRI-D Feature for image X:

  Histograms: $H_1,\ldots, H_8$ for indices $L_1,\ldots,L_8$,

- Threshold T: Determines what is an edge, and controls noise sensitivity. Typically, $T = \sigma/2$.

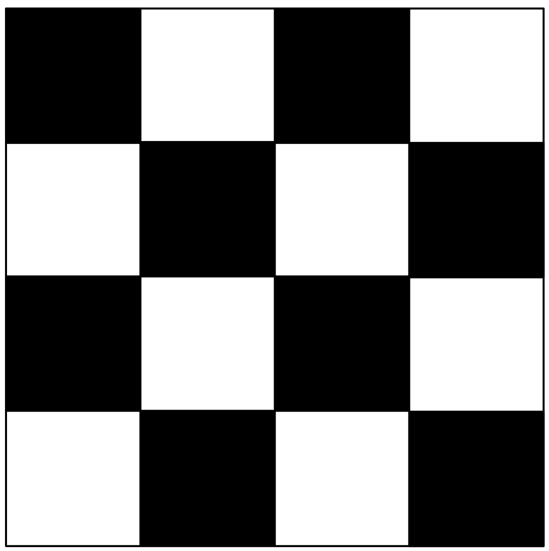
- Size limit K: Limits size of texture elements detected by LRI, and reduces computation. Typically, $K = 4$.

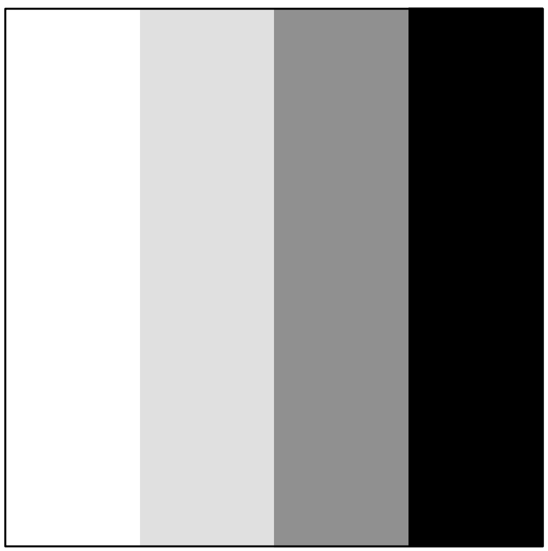
Comments

- LRI-A vs. LRI-D:
  - LRI-A index $L_{i,d}$ measures size of adjacent texture element, if there is one. It is zero at i unless there is edge adjacent to i in direction d.
  - LRI-D index $L_{i,d}$ measures dist. from i to nearest edge in direction d. It usually decreases by 1 one when moving in direction d from i.
  - Hence, LRI-A values are sparser. LRI-D values are more correlated/ redundant.
- LRI and LBP are complementary:
  - LBP provides angular pattern info; no direction or element size info.
  - LRI provides direction and size info; no angular pattern info.

Why Do LRI Indices Have "±" Signs?

- Without signs, the following have same horizontal LRI histograms:

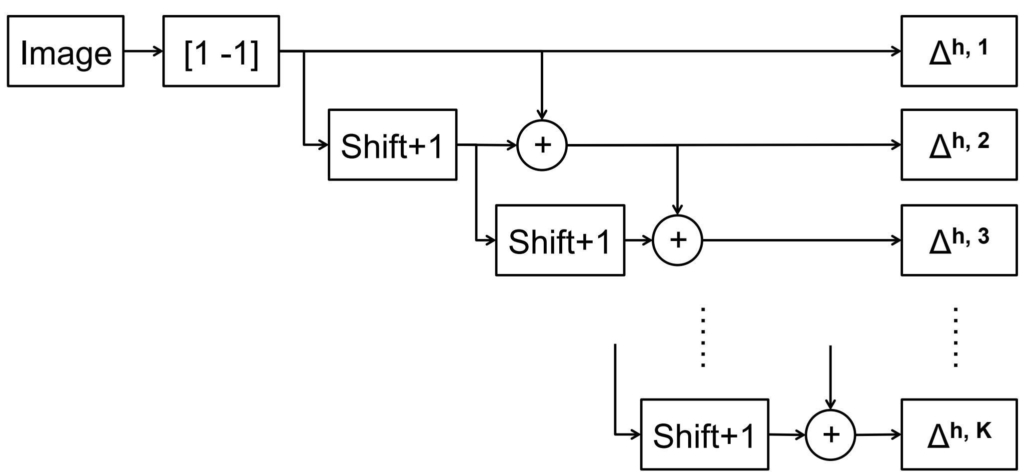




Computation of LRI Indices

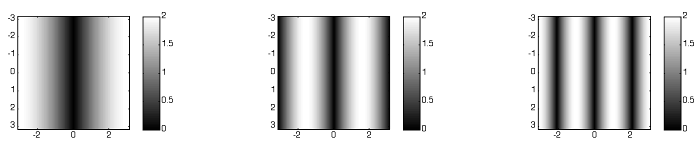
- Fast algorithm:
  - Precalculate & store pairwise differences between all pairs of pixels separated by k =1, …, K in unsigned directions d (horiz, vert, diag, anti-diag)
  - Pairwise differences are then used to compute LRI indices.
  - ~$4K$ operations per pixel.
  - Pairwise differencing in direction d of pixels separated by k can be viewed as a filtering,

    e.g., h = [1 0 0 0 -1] is impulse response of horizontal filter for k = 4.
  - Filtering can be done in simple recursive fashion:



Pixel Differencing Filters Create De Facto Subband Decomposition

- Pairwise differencing in direction d of pixels separated by k can be viewed as a filtering.
- The $4K$ filters comprise a simple multiscale, multiorientation subband decomposition, with 4 orientations and K scales.
- Example: Frequency response for 3 scales in horizontal direction:



- Might use this decomposition instead of steerable pyramid; requires much less computation

Texture Similarity Metrics Based on LRI

1. Purely LRI-based similarity metric:
   - LRI feature: Combine 8 histograms into vector of dim. 8 x (2*K*+1)
   - Compare similarity of the vectors obtained for images *X, Y* with divergence.

$$\sum_i H_Y(i)\log\frac{H_Y(i)}{H_X(i)}$$

Texture Similarity Metrics Based on LRI

2. Combine/pool LRI with other features:

   a. LBP

      – Use 8 nearest neighbor pixels, without interpolating values on circle.

   b. Luminance (mean) μ

      – Comparison term: $$M(X,Y) = \left(\frac{\max\left(|\mu_X - \mu_Y|,\ 10 \right)}{256} \right)^2$$

      – Penalizes only mean differences larger than 10.

   c. Subband Contrast Distrib'n (SCD) $\sigma_1^2, \ldots, \sigma_{12}^2$ from steerable filter.

      – Comparison term: $$SCD(x,y) = \prod_{b=1}^{12} \frac{2\sigma_{x,b}\sigma_{y,b} + \varepsilon}{\sigma_{x,b}^2 + \sigma_{y,b}^2 + \varepsilon}$$

   c'. Subband Contrast Distrib'n (SCD_f) $\sigma_1^2, \ldots, \sigma_{4K}^2$ from pairwise differencing filters, which are much simpler to compute since filtering is already done.

Texture Similarity Metric Based on LRI

- New LRI-based texture similarity metric:

$$LRI^{+} = LRI \times LBP^{1.1} \times M \times \left(\tan\left((1\text{-}SCD)\,\pi/2\right)\right)^{1.2}$$

  - Metric is nonnegative with zero meaning identical, and a small value meaning similar.
  - LBP computed from pixel differences with few add'l computations.

- Complexity:
  - Leaving aside subband costs, STSIM2 (with a global window, which is the simplest version), requires approximately 500 operations per pixel.
  - LRI and SCD requires 16 and 36 operations per pixel, respectively.
  - Steerable filter pyramid needs 14 FFTs, each with $5 \log N$ op's per pixel.
  - With SCD_f, no transform needed for LRI^+, beyond that needed for LRI.
  - Conclusion: LRI^+ with SCD_f is at least 10 times faster than STSIM2.

Metric Test 1: Identical Texture Retrieval

Corbis-derived database

CW-SSIM	STSIM 2	STSIM 2 global	STSIM M global	LPB (8,1)	LRI-A	LRI-D	SCD	LRI^+-a	LRI^+-b	LRI^+-b
39	74	93	96	90.0	91.8	83.2	83.7	98.7	98.1	99.0

LRI-A, LBP, M, SCD

LRI-A, LBP, M, SCD_f

LRI-D, LBP, M, SCD

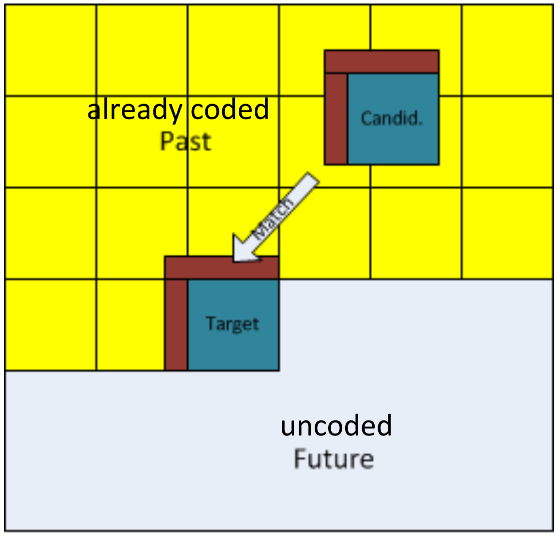
Notes:

- LRI-A > LBP by themselves.
- LRI-A > LRI-D by themselves.
- LRI-A ≈ LRI-D when combined with other features.
- LRI^+_b is simplest of LRI^+ methods
- Combining features exploits complementarities

Metric Test 2: MTC

- Consider encoding current block, e.g., 32x32, by pointing to a previously encoded block.
- If no sufficiently good previous block, encode with JPEG.
- Key to success is a texture similarity metric for judging if a candidate previous block is sufficiently good.

- Use LRI^{+}_{b} instead of STSIM2



original: 1024x1024

JPEG coded

all encodings at 0.18 bpp

MTC coded with STSIM2

MTC coded with LRI_b

- STSIM2 & LRI^{+}_{b} give similar performance.
- LRI-based coder runs ~6 times faster.

Outline

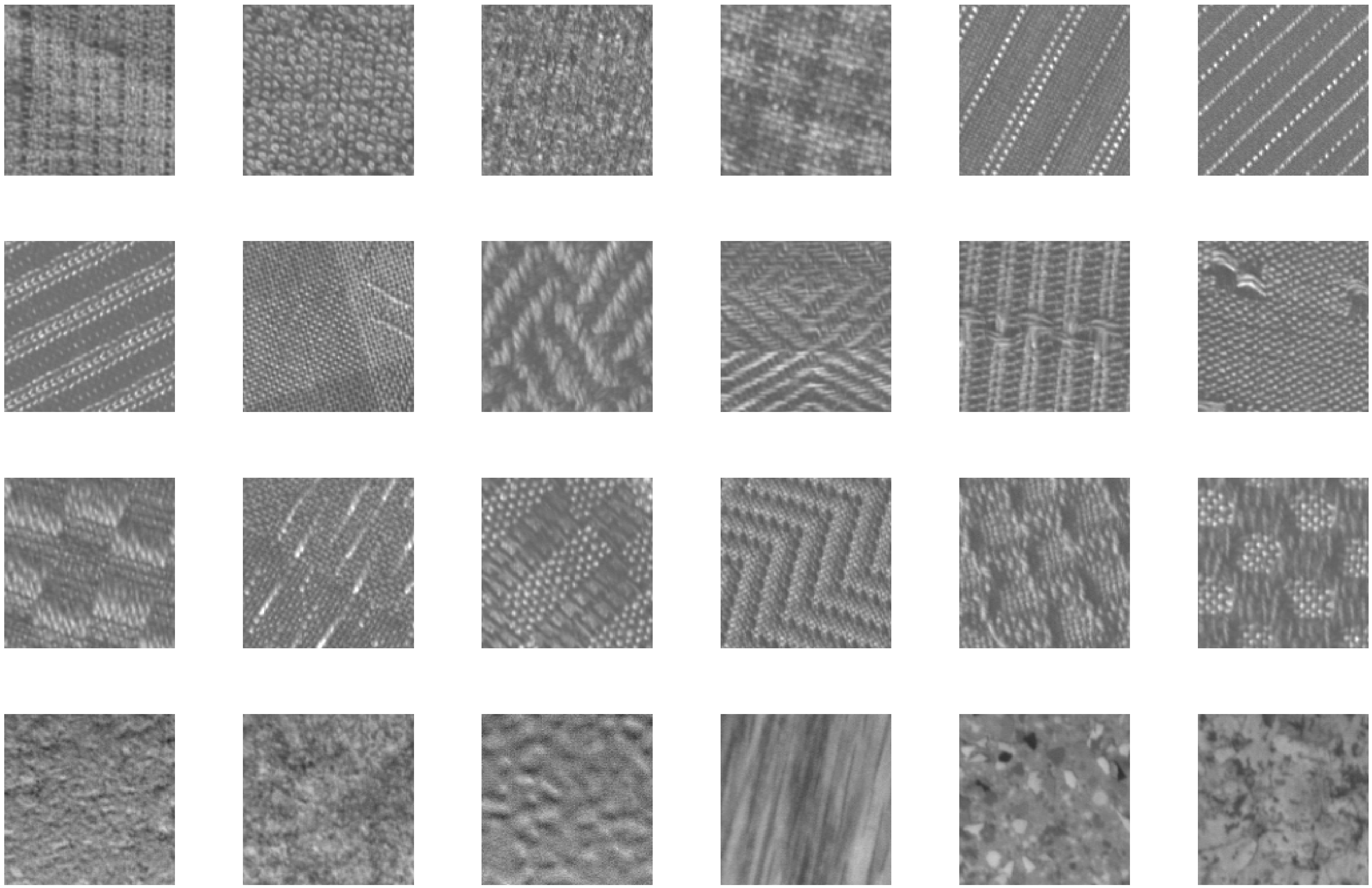
- Review
  - Point-by-point metrics
  - SSIM (Structural Similarity Metric)
- STSIM (Structural Texture Similarity Metric): new SSIM type metric:
  - Performance in identical texture retrieval and image compression
- Review
  - LBP (Local Binary Patterns)
- LRI (Local Radius Index): new local-pattern metric
  - Performance in identical texture retrieval and image coding
- RI-LRI (Rotation invariant LRI):
  - Performance in texture classification
- Concluding remarks

Yuanhao Zhai

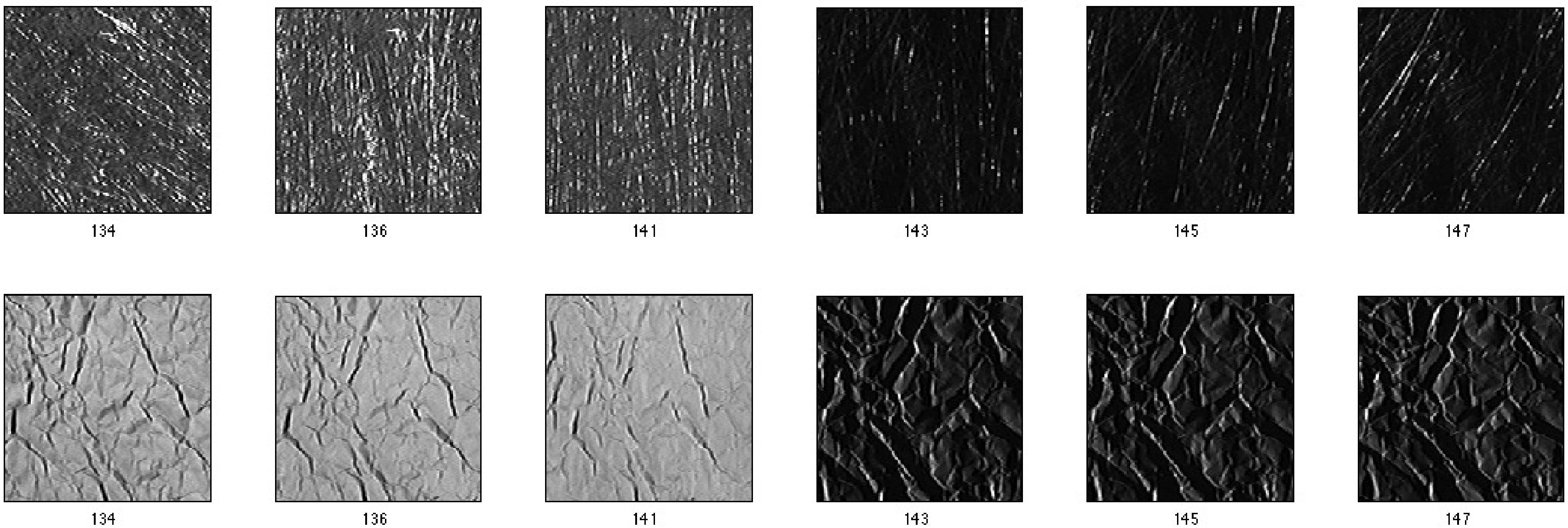
Metric Test 3: Texture Classification

- Problem: Given examples of *K* texture classes and a query from an unknown one of these classes, identify the class of the query.
- Common approach: *M nearest neighbors* (*M-NN*)
  - Measure the "distance" of query to each example in each class.
  - Decide the class that occurs the most among the *M* closest examples.
- Use a texture similarity metric to determine "distance"
- The goodness of a similarity metric is judged by the classification accuracy that results, expressed as a percent.
- OUTEX and CUReT databases are de facto "standards" for testing texture metrics in classification.
- Texture classification cannot presume texture has known orientation. So metric must be rotation invariant.
- LBP is rotation invariant. LRI is not.
- Gray-scale invariance might also be desired.

Samples of OUTEX Database

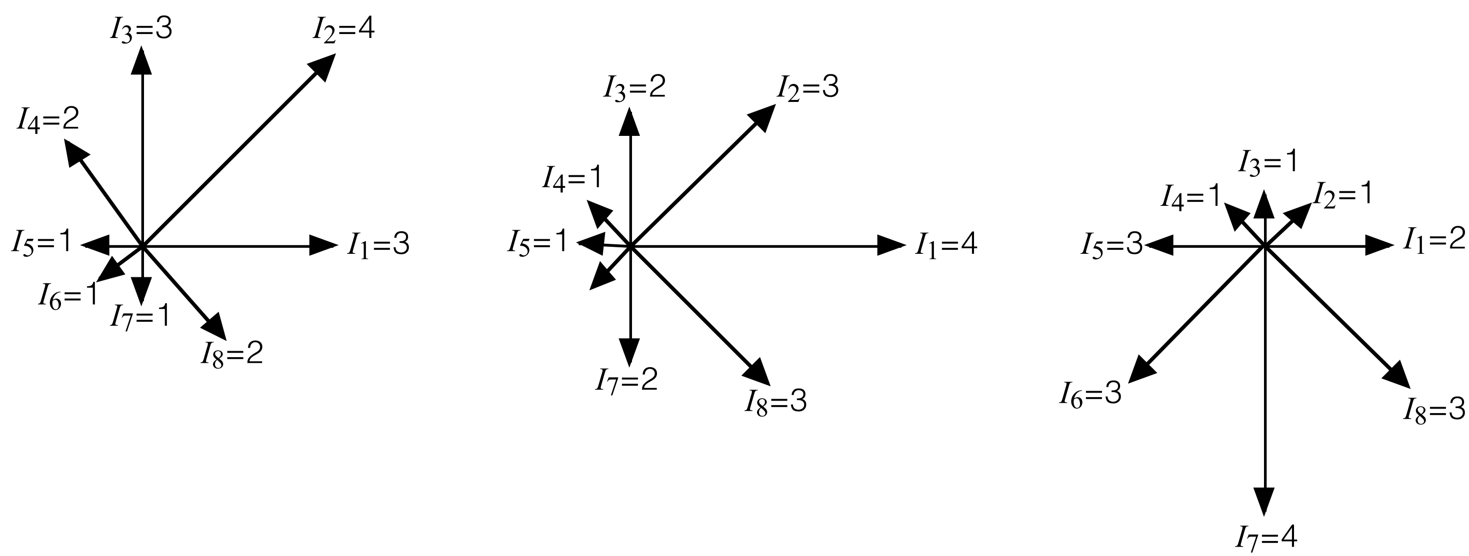


Samples of CURet Database



RI-LRI (Rotation Invariant LRI)[1]

original LRI indices



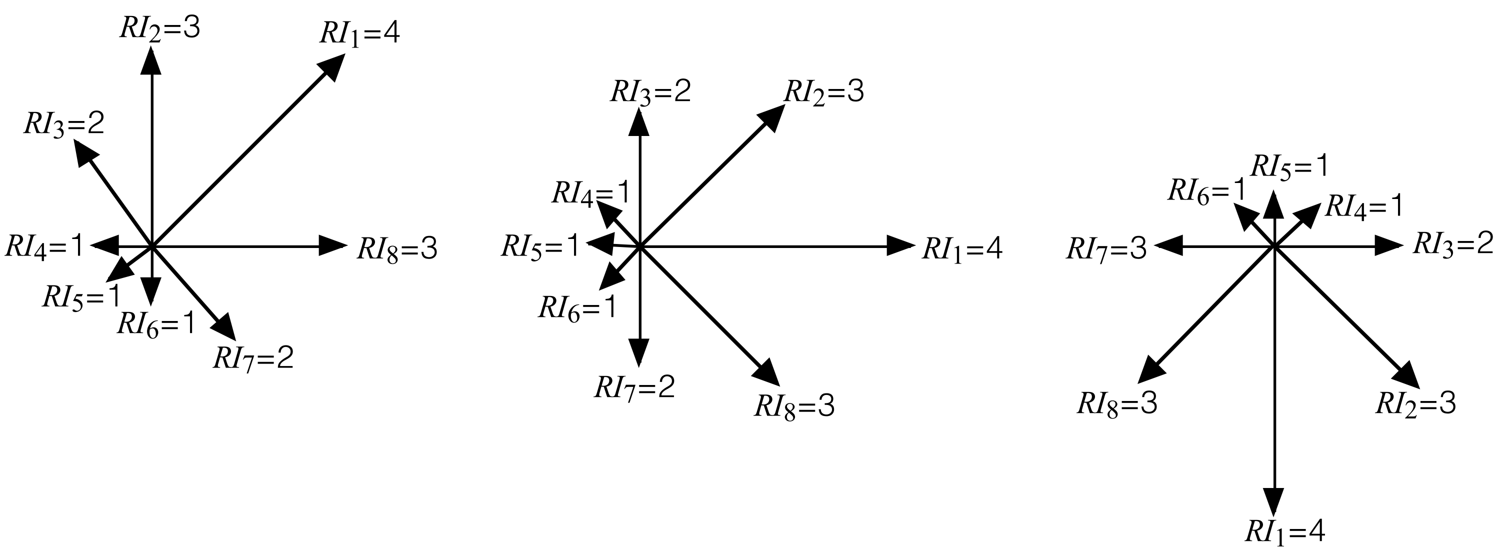
Now, H_1 is histogram of indices RI_1 in dominant directions.

H_2 is histogram of indices adjacent to dominant directions, etc.

[1]Zhai, Neuhoff, Pappas, ICIP, 2014.

RI-LRI (Rotation Invariant LRI)[1]

rotation invariant LRI indices



Now, H_1 is histogram of indices RI_1 in dominant directions.

H_2 is histogram of indices adjacent to dominant directions, etc.

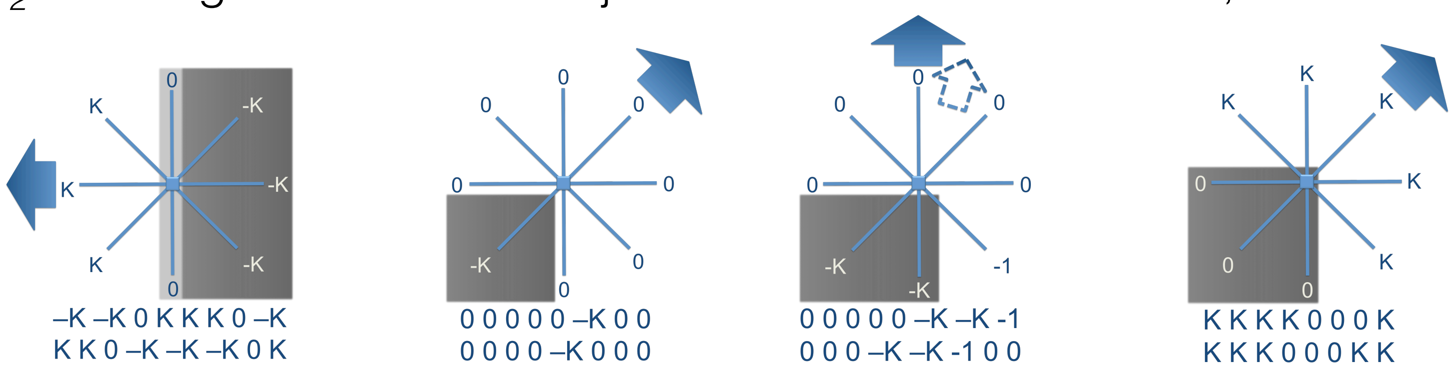
[1]Zhai, Neuhoff, Pappas, ICIP, 2014.

RI-LRI (Rotation Invariant LRI)[1]

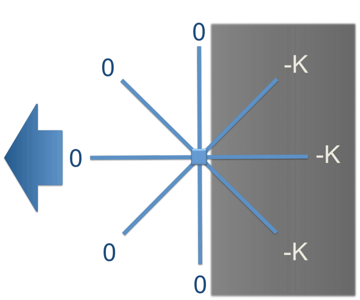
- To make LRI rotation invariant, for each pixel i,
  - For each pixel i estimate dominant direction: $D_i = \text{angle}\left(\sum_{d=1}^{8} I_{i,d} \times \phi_d\right)$
    where $[\phi_1,\ldots,\phi_8] = [1, 1+j, j, -1+j, -1, -1-j, -j, 1-j]$
  - Circularly shift pixel i's LRI indices, $I_{i,1},\ldots,I_{i,8}$, forming $RI_{i,1},\ldots,RI_{i,8}$, such that $RI_{i,1}$ is the index in direction closest to the dominant direction D_i :

$$RI_{i,d} = I_{i,\,1+(d-1+\overline{D}_i) \bmod 8}$$

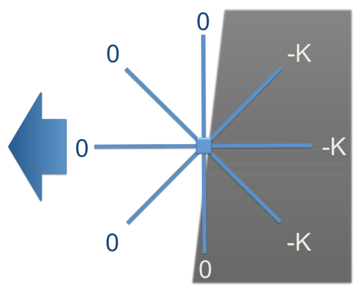
    where $\overline{D}_i$ is D_i rounded to nearest multiple of $\pi/4$.
  - Now, H_1 is histogram of indices in dominant directions.
  - H_2 is histogram of indices adjacent to dominant directions, etc.



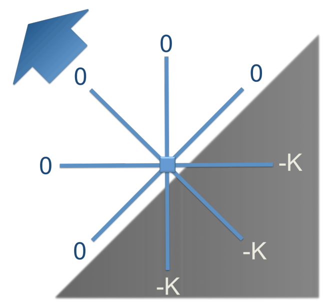
[1]Zhai, Neuhoff, Pappas, ICIP, 2014.



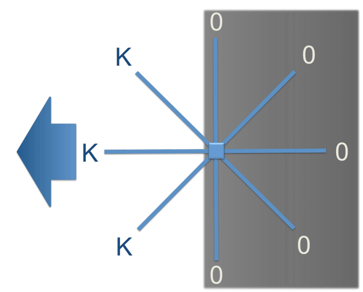
–K –K 0 0 0 0 0 –K
0 0 0 –K –K –K 0 0



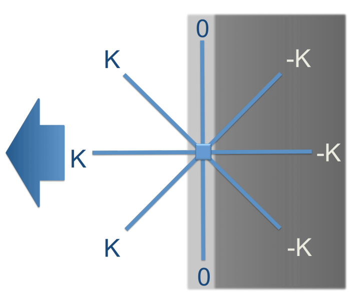
–K –K 0 0 0 0 0 –K
0 0 0 –K –K –K 0 0



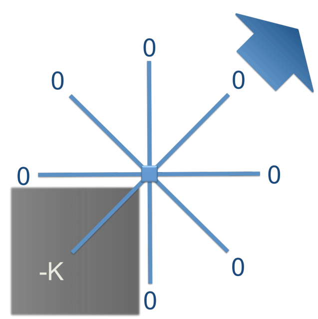
–K 0 0 0 0 0 –K –K
0 0 0 –K –K –K 0 0



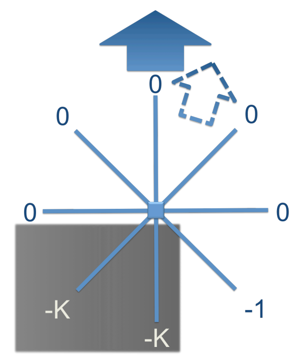
0 0 0 K K K 0 0
K K 0 0 0 0 0 K



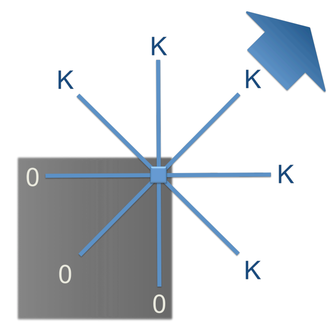
–K –K 0 K K K 0 –K
K K 0 –K –K –K 0 K



0 0 0 0 0 –K 0 0
0 0 0 0 –K 0 0 0

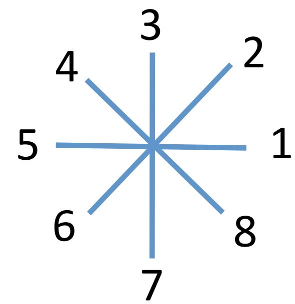


0 0 0 0 0 –K –K -1
0 0 0 –K –K -1 0 0



K K K K 0 0 0 K
K K K 0 0 0 K K

RI-LRI Feature Vector



- Observations:
  - Most distinctive information is in dominant direction (d=1), opposite direction (d=5), & orthogonal directions (d=3,7).
  - Orthogonal directions (1 & 3), (5 & 7) contain correlated information.
- LRI Feature Vector is 2-D histogram
  - For each m,n between $-K$ and K

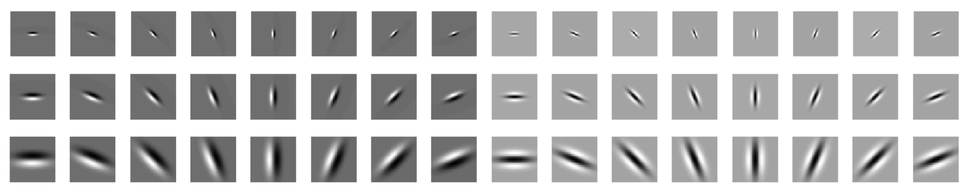
$$H(m,n) = \left[\#\text{pixels s.t. } (RI_{i,1}, RI_{i,3}) = (m,n)\right] + \left[\#\text{pixels s.t. } (\mathrm{RI}_{i,5}, \mathrm{RI}_{i,7}) = (m,n)\right]$$

  - Feature Vector length = $(2K+1)^2$. (Typically, $K = 4$)

Feature	Complexity		Feature	Complexity
LBP(8,1) w/o interp.	30		LBP(8,1)	90
LRI (K = 4)	36		LBP(16,2)	242
RI-LRI (K = 4)	47		LBP(24,3)	394

RI-LRI^{+} Metric Based on RI-LRI

- Combine RI-LRI with
  - LBP ((8,1) no interpolation)
  - RI Subband Contrast Distribution (RI-SCD)
    - Motivated by LEBC[1], measure variances $s_1^2, \ldots, s_{48}^2$ from subband decomposition generated by "edge" (first deriv.) and "bar" (second deriv.) filters with 3 scales and 8 orientations.



scale 3

scale 2

scale 1

    - To attain rotation invariance, for each scale and choice of "edge" or "bar" arrange the 8 variances in decreasing order.
    - RI-SCD similarity of images: $\text{RI–SCD}(X,Y) = \prod_{i=1}^{48} \frac{2\, s_{X,i} s_{Y,i} + \varepsilon}{s_{X,i}^2 + s_{Y,i}^2 + \varepsilon}$
- RI-LRI^{+}:

$$\text{RI-LRI}^{+} = (\text{RI-LRI})^{0.7} \times \text{LBP}^{0.4} \times \big(1-(\text{RI-SCD})\big)^{1.3}$$

Metric Test: Texture Classification

- OUTEX database: percent accuracy averaged over 3 test scenarios

LBP (8,1)	RI-LRI (K=4)	VZ-Joint	VZ-MR8	DLPB +NGF	CLBP	DNS +LBP	LEBC	RI-LRI^{+}
71.8	84.9	91.8	93.0	94.1	96.7	95.5	98.7	98.4

- CUReT database: percent accuracy

LBP (8,1)	RI-LRI (K=4)	VZ-Joint	VZ-MR8	CLBP	DNS +LBP	LEBC	RI-LRI^{+}
80.8	91.9	97.7	97.8	97.0	95.0	98.5	98.1

- RI-LRI^{+} (with 139-dim'l feature vector) is nearly as good as LEBC (with 1280-dim'l feature vector).
- RI-LRI^{+} uses 1-NN, i.e., Nearest Neighbor classification

Concluding Remarks

- New texture features: LRI, RI-LRI
- New texture similarity metrics:
  - STSIM, STSIM2 –-- SSIM-type
  - LRI, LRI^+ ---- local pattern type
  - RI-LRI, $RI\text{-}LRI^+$ ---- local pattern type, rotation invariant
- State-of-the-art performance in
  - Identical texture retrieval
  - Texture classification
  - MTC compression
- Order of magnitude lower dimensionality of feature vector than similarly performing feature vectors.

Current Work

- LRI based on “genuine” edge detectors, e.g., Canny
- Analytically predict LRI histograms on periodic texture patterns
- Study sensitivity of LRI to noise
- New pooling methods

References

1. J. Portilla and E.P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *Int. J. Comput. Vis.*, 40: 49-70, Oct. 2000.
2. Z.Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Proc.*, vol. 13, pp. 600-612, Apr. 2004.
3. T. Ojala, M. Pietikainen and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 971-987, Jul. 2002.
4. M.P. Sampat, Z.Wang, S. Gupta, A.C. Bovik and M.K. Markey, “Complex wavelet structural similarity: a new image similarity index,” *IEEE Trans. Image Proc.*, vol. 18, pp. 2385-2401, Nov. 2009.
5. X. Zhao, M.G. Reyes, T.N. Pappas and D.L. Neuhoff, “Structural texture similarity metrics for retrieval applications,” IEEE Intl. Conf. on Image Proc. (ICIP), pp. 1196-1199, Oct. 2008.
6. J. Zujovic, T. N. Pappas and D. L. Neuhoff, “Structural texture similarity metrics for image analysis and retrieval,” *IEEE Tr. Im. Proc.*, 22, pp. 2545-2558, Jul. 2013.
7. G. Jin, Y. Zhai, T.N. Pappas and D.L. Neuhoff, “Matched-texture coding for structurally lossless compression,” IEEE Intl. Conf. on Image Proc. (ICIP) , Oct. 2012.
8. Y. Zhai, D.L. Neuhoff and T.N. Pappas, “Local Radius Index -- A New Texture Similarity Feature,” *Proc. IEEE ICASSP*, pp. 4266-4270, Vancouver, CA, May 2013.
9. Y. Zhai, and D.L. Neuhoff, “Rotation-Invariant Local Radius Index: a Compact Texture Similarity Feature for Classification,” to be presented *IEEE ICIP*, Oct. 2014.
10. (VZ-Joint, VZ-MR8) Z. Guo, L. Zhang and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657-1663, Jun. 2010.
11. (DLBP+NGF) S. Liao, M. Law and A. Chung, “Dominant local binary patterns for texture classification,” *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1107-1118, May 2009.
12. (CLBP, LEBC) T. Song, F. Meng, B. Luo and C. Huang, “Robust texture representation by using binary code ensemble,” VCIP, pp. 1-6, 2013.
13. DNS+LBP) F. M. Khellah, “Texture classification using dominant neighborhood structure,” *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3270-3279, Nov. 2011.