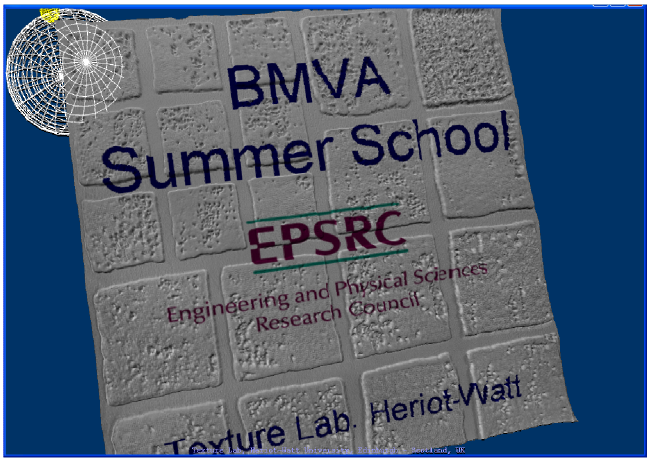


Texture notes v001

Prof Mike Chantler
Texture lab.
Heriot-Watt University
www.macs.hw.ac.uk/~mjc

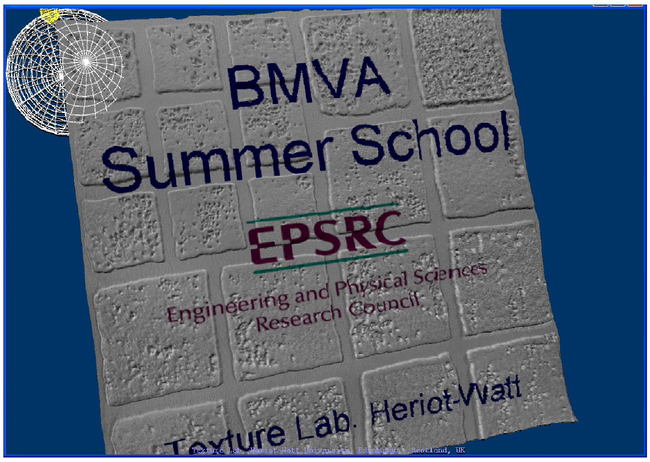
Aims

- **Overview and basics of texture analysis**
- **Environmental factors & why they are important**
- **A few pointers to what’s hot and what’s not**

- **Thursday – so lots of pics & minimum maths!** (its in the references)

Lecture Outline

- **What is Texture?**
- **Computer Vision ‘Texture’?**
- **Texture Features**
- **Linear Filter based Features**
- **Local Variance Estimation**
- **1D Example & 2D Result**
- **Other Features**
- **A Simple Strategy**
- **Image Texture Problems**
- **Other Applications**



What is texture?

What is texture?

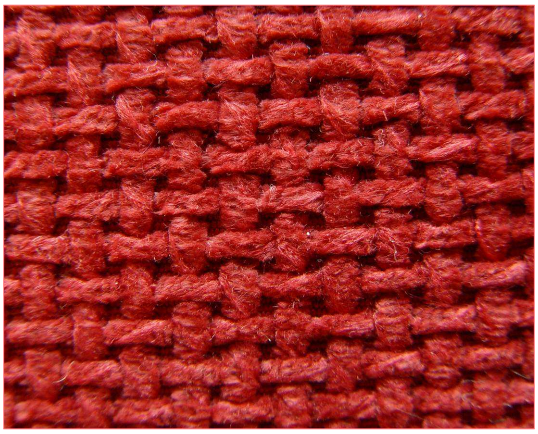
- **Associated with**
  - Touch?
  - Taste?
  - Feel?
- **Depends upon the task?**
- **Used to infer perceived properties?**

BMW leather

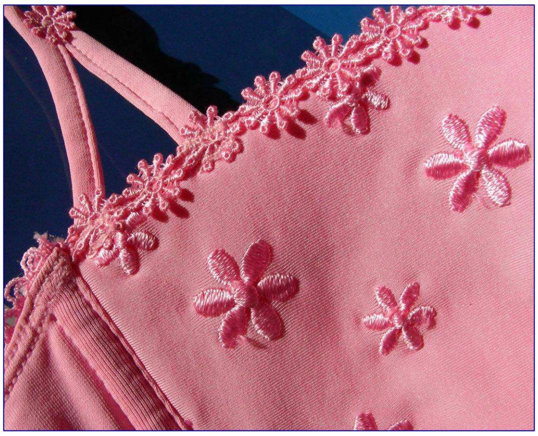
Sony vaio laptop

B&Q wallcovering

Accessorize accessory



Marks and Spencer upholstery



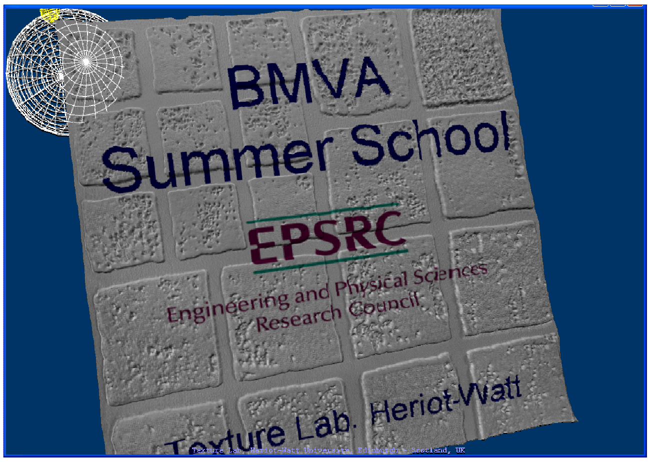
La Senza lingerie

Types of texture

1. **Surface texture**
   - Spatial height variation (physical relief)
   - What you can touch and feel
2. **Albedo texture**
   - Spatial colour variation of a Lambertain surface
3. **Reflectance texture**
   - Spatial variation of gloss/matte function
4. **Image texture**
   - What is projected onto the retina or camera sensor
   - 2D phenomena
   - Function of 1, 2, 3 and imaging conditions
5. **Volume texture**
   - 3D spatial variation
6. **Dynamic texture**
   - Additional temporal variation

Texture, Mike Chantler, 28 June 2007

page 15

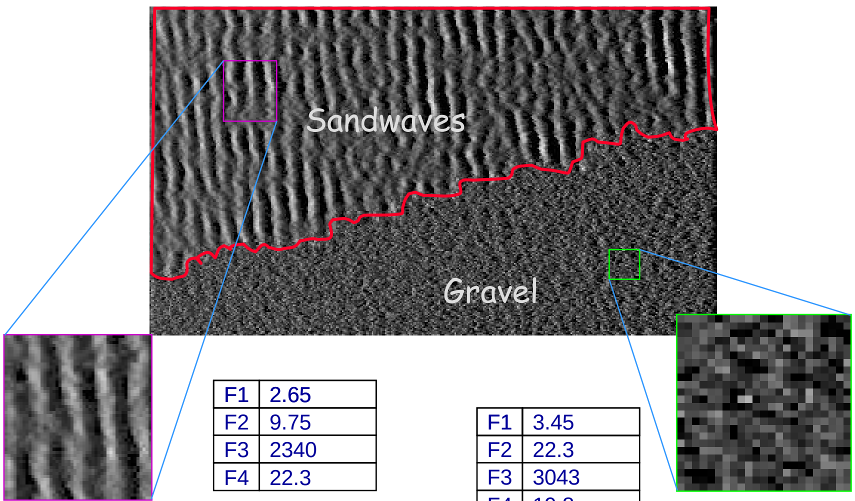


Computer Vision ‘texture’?

Computer Vision View?

- Image data that are used to infer physical properties or boundaries
- Usually single still images
- Can be multi-modal but often grey-scale
- Characterisation of spatial variation
- Characteristion = set of feature values (the feature vector)
- Characterisation of ‘homogeneous’ textures should be stationary

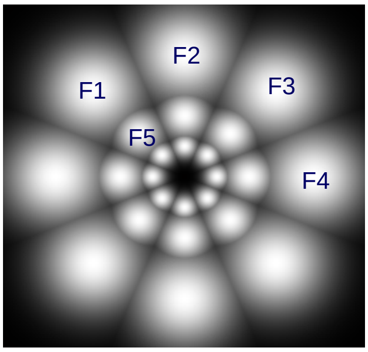
Computer Vision View?

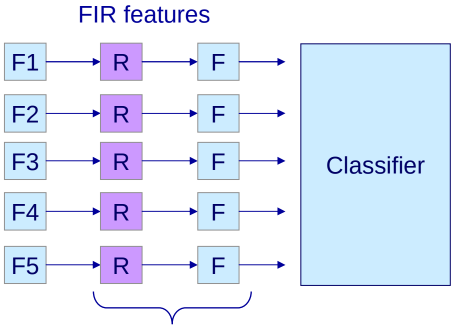


F1	2.65
F2	9.75
F3	2340
F4	22.3

F1	3.45
F2	22.3
F3	3043
F4	19.8

Filter banks: psychophysical inspirations

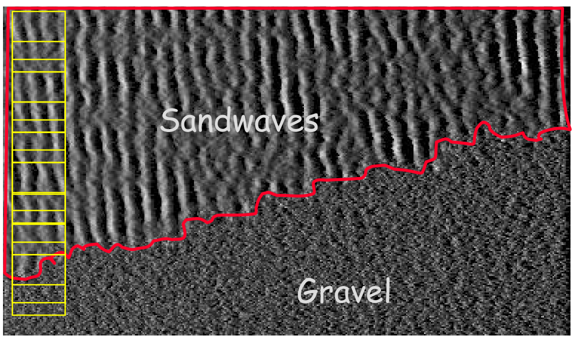




Rectification + averaging
=
local variance measure

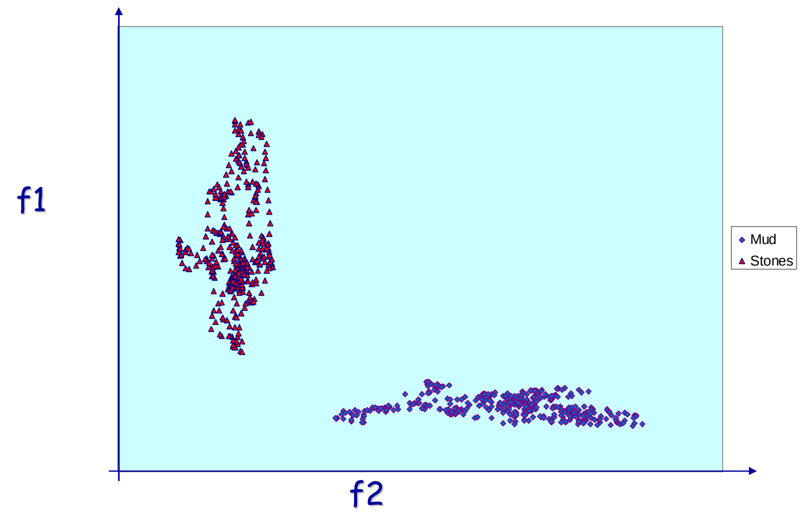
[Randen & Husoy, PAMI, 1999]

Computer Vision View?



F1	2.65
F2	9.75
F3	2340
F4	22.3

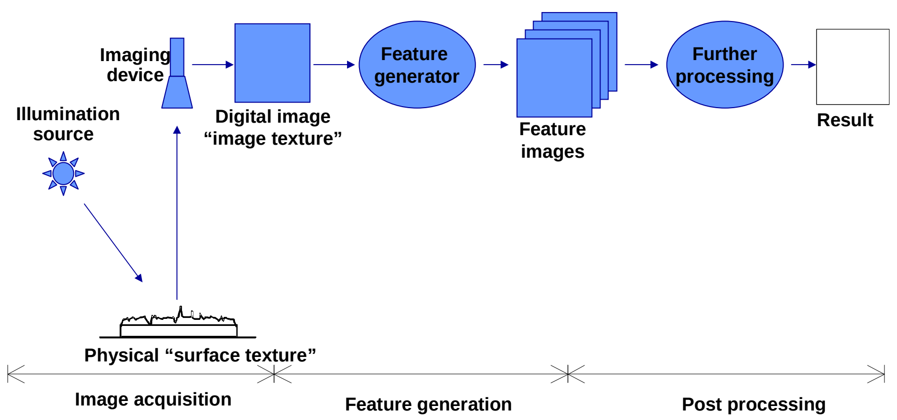
2 feature scatter plot



Texture, Mike Chantler, 28 June 2007

page 21

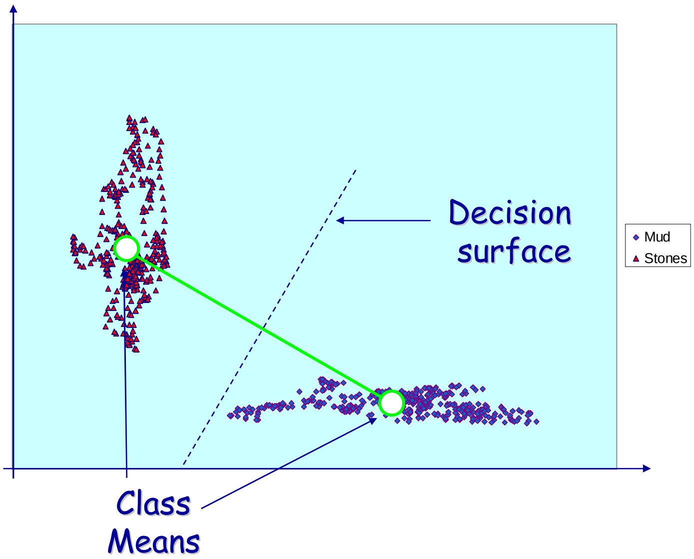
The texture pipeline



Texture, Mike Chantler, 28 June 2007

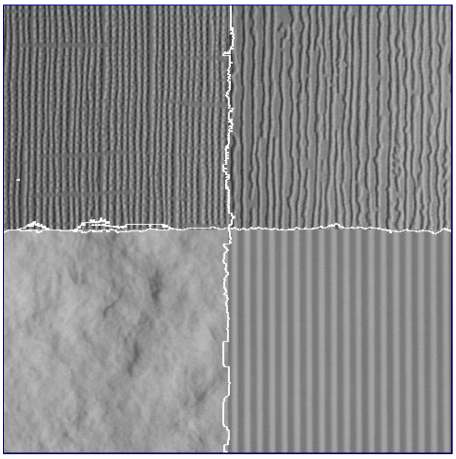
page 22

Minimum (mean) distance classifier



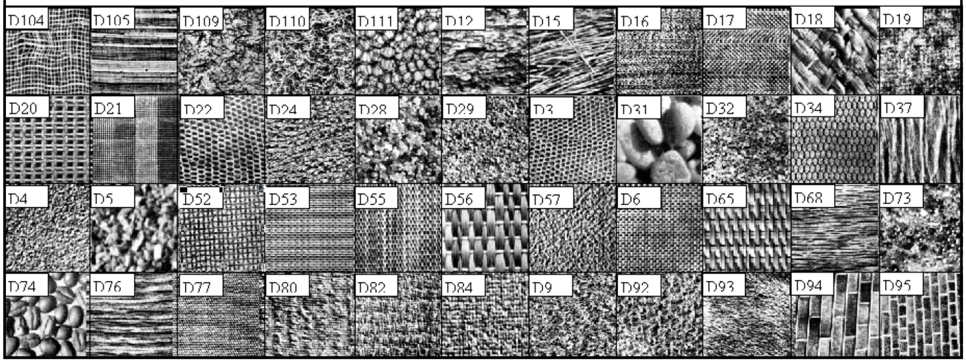
Applications

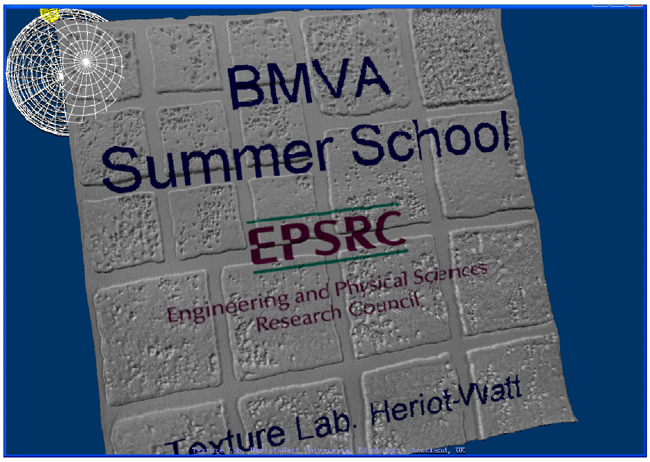
- **Segmentation**
- **Classification**
- **Retrieval**
- **Synthesis**
  - from example
  - parameter driven
- **Shape from texture**
- **Measurement of appearance**



Previous Research

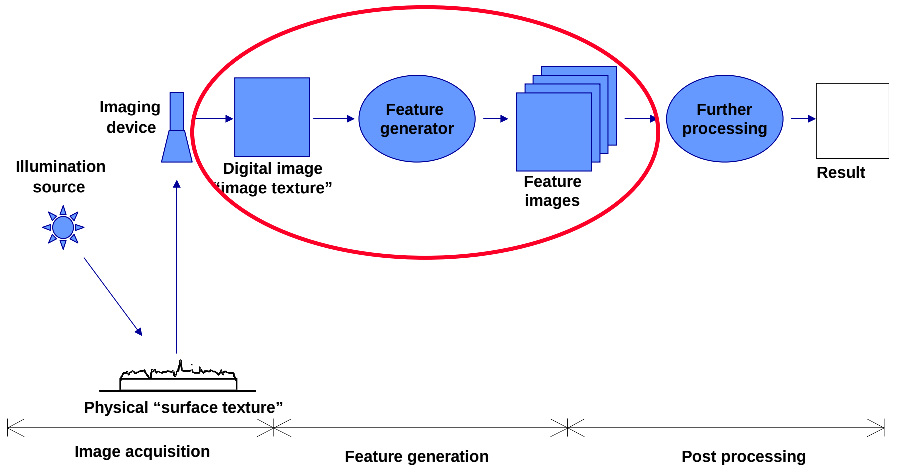
– Brodatz66-based papers
– Haralick, Van Gool, Reed/Du Buff, Randen





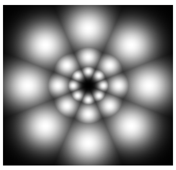
Texture features

Features



Texture features

- **Linear Filters**
  - Marginal statistics (particularly variance) of
    - Laws
    - Gabor
    - Steerable pyramids
    - Wedge, ring etc.



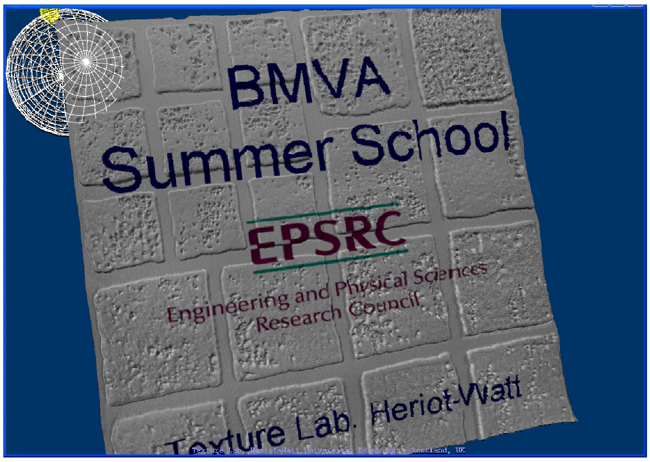
- **Other**
  - Morphological operators
  - Local binary patterns
  - Co-occurence matrices
  - MRF

Other features

- Other marginal stats (skew, kurtosis..)
- Histogram dist.
- Joint stats.

Texture, Mike Chantler, 28 June 2007

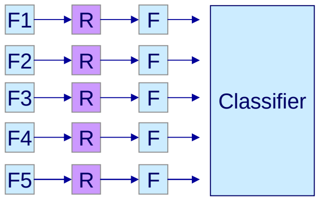
page 29



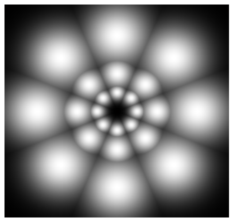
Linear filter based texture features

Filter banks

- **Terminology**
  - Filter banks
  - FIRs (Finite Impulse Response)
  - Non-recursive filters
  - Frequency channel model
  - Power spectrum measures
  - Bandpass filters
  - FRF (Filter Rectify Filter – from psychophysics)



- **Types**
  - Steerable filters and other pyramids
  - Gabors
  - Laws
  - Wedge/ring filters

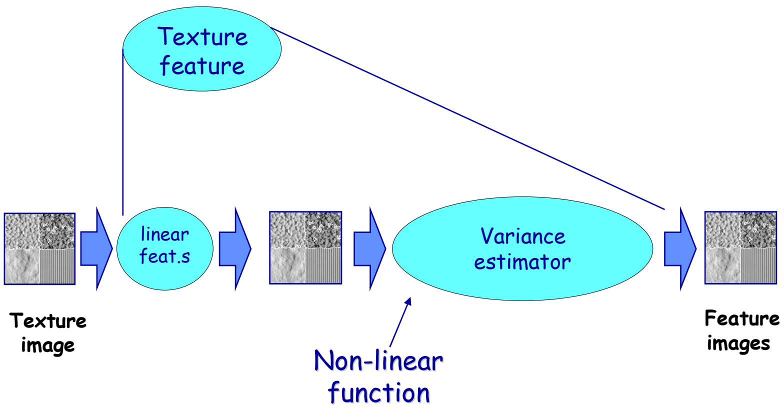


Texture, Mike Chantler, 28 June 2007

page 31

Simplest post-processing: variance

$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$

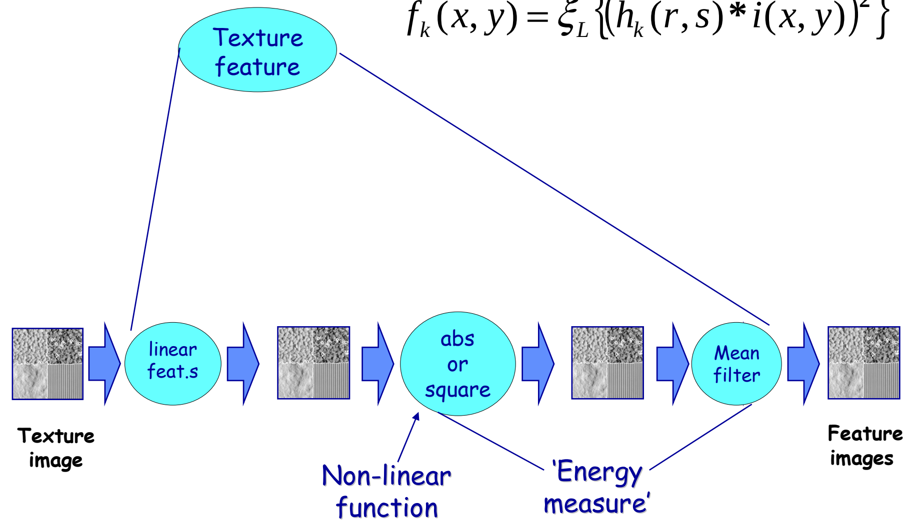


Texture, Mike Chantler, 28 June 2007

page 32

‘Energy’ or variance estimators

$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$



Simple example: Laws Filters

Developed from simple set of filters

L3 = (1,2,1) - "level detection"

E3 = (-1,0,1) - "edge detection"

S3 = (-1,2,-1) - "spot detection"

Convolved with each other they give:

L5 = L3*L3 = (1,4,6,4,1)

E5 = L3*E3 = (-1,-2,0,2,1)

S5 = E3*E3 = (1,0,-2,0,1)

R5 = S3*S3 = (1,-4,6,-4,1)

Laws Filters

Note

$L3^T * E3 = (1,2,1)^T * (-1,0,1)$

1
2
1

*

-1	0	1

=

-1	0	1
-2	0	2
-1	0	1

Gx Sobel
i.e. a directional
edge detector

Laws' filters (L5E5)

$$h_{L5E5}(r,s) = (1,4,6,4,1)^T * (-1,-2,0,2,1)$$

-1	-2	0	2	1
-4	-8	0	8	4
-6	-12	0	12	6
-4	-8	0	8	4
-1	-2	0	2	1

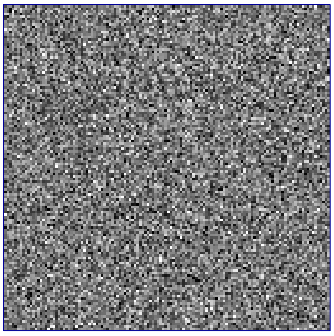
=

1
4
6
4
1

*

-1	-2	0	2	1

Laws L5E5: white noise response

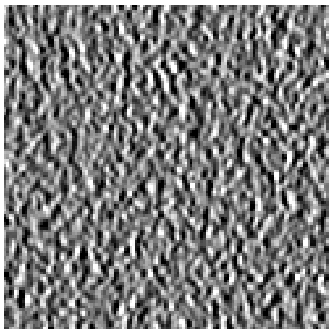


White noise

(2dfrac -r -b0 -n128 2dfrac.f)

L5E5

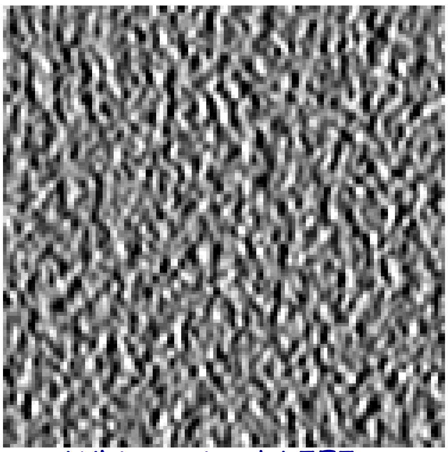
-1	-2	0	2	1
-4	-8	0	8	4
-6	-12	0	12	6
-4	-8	0	8	4
-1	-2	0	2	1



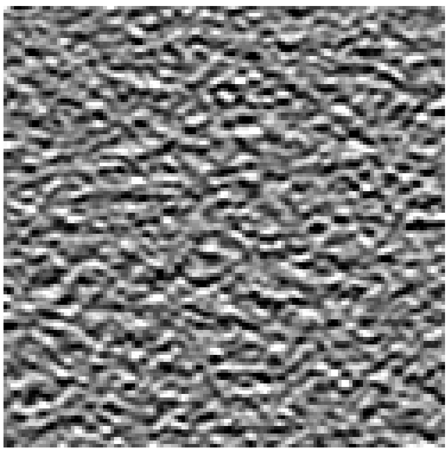
White noise * L5E5

(mask -f -m masks/L5L5 2dfrac.f l5l5.f)

Laws L5E5 & E5L5 response

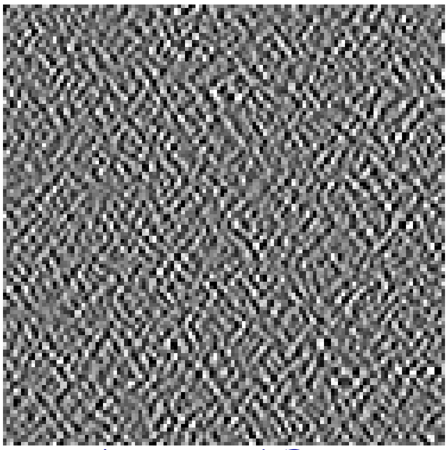


White noise * L5E5

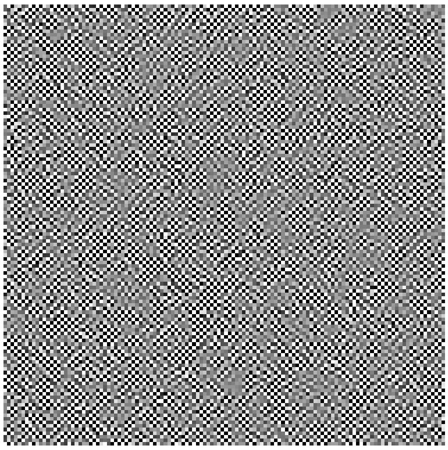


White noise * E5L5

Laws: E5S5 & R5R5



White noise * E5S5



White noise * R5R5

Laws Filters (frequency response)

$$h_{L3}(r,s) = \mathbf{L3} = \mathbf{(1,2,1)}$$

$$H_{L3}(\omega) = 1.e^{-j\omega} + 2.\ e^{0} + 1.e^{j\omega}$$

From FIR digital filter theory (see EE intro. signal processing texts)

Laws Filters (frequency response)

L3 = (1,2,1)

$H_{L3}(\omega) = 1.e^{-j\omega} + 2.\ e^{0} + 1.e^{j\omega}$

Lag = +1

Lag = 0

Lag = -1

Laws Filters (frequency response)

L3 = (1,2,1)

$H_{L3}(\omega) = 1.e^{-j\omega} + 2.\ e^{0} + 1.e^{j\omega}$

$= (\cos\omega - j\sin\omega) + 2 + (\cos\omega + j\sin\omega)$

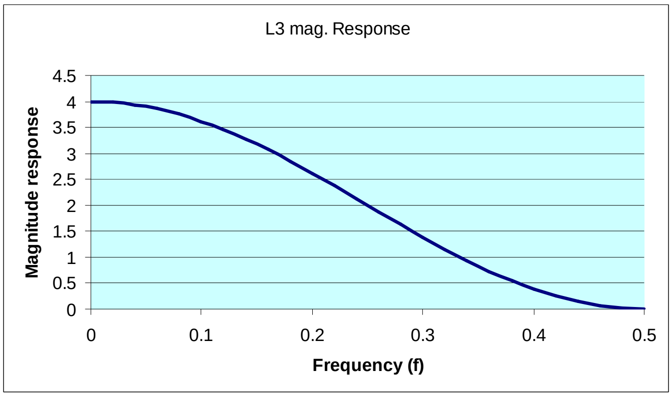
$= 2(\cos\omega + 1)$

Where ω is angular frequency =2Πf radians per second

Laws Filters (frequency response)

L3 = (1,2,1)

$H_{L3}(\omega) = 2(\cos 2\Pi f + 1)$

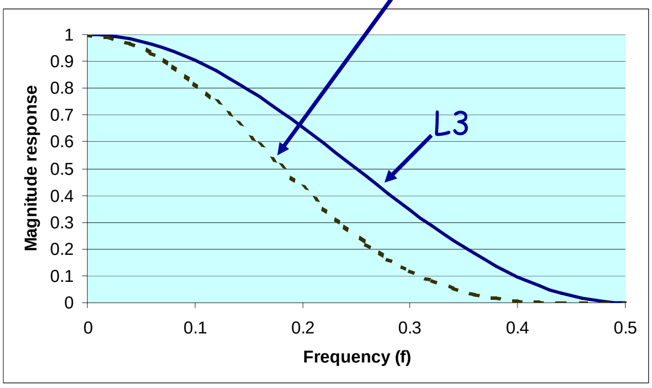


Laws (frequency response)

L5 = L3*L3 = (1,4,6,4,1)

$$|H_{L5}(\omega_1)| = |e^{-j2\omega_1} + 4e^{-j\omega_1} + 6 + 4e^{j\omega_1} + e^{j2\omega_1}|$$

$$= 4(1+\cos\omega_1)^2$$



Laws Filters (frequency response)

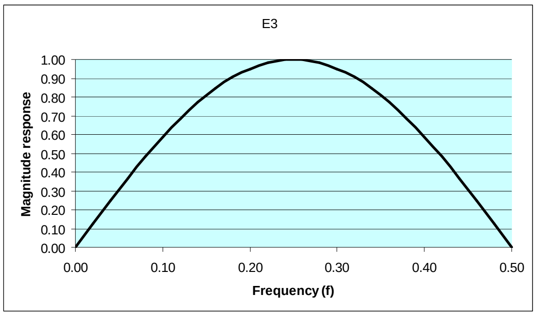
E3 = (-1,0,1)

$H_{L3}(\omega) = -1.e^{-j\omega} + 0.\ e^{0} + 1.e^{j\omega}$

$= -1.(\cos\omega - j\sin\omega) + (\cos\omega + j\sin\omega)$

$= 2j\sin\omega$

90° phase change



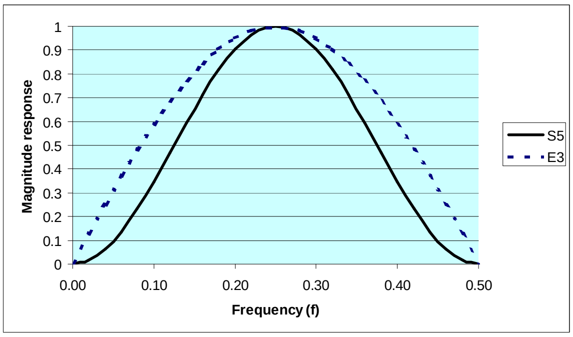
Laws Filters (frequency response)

S5 = E3*E3 = (1,0,-2,0,1)

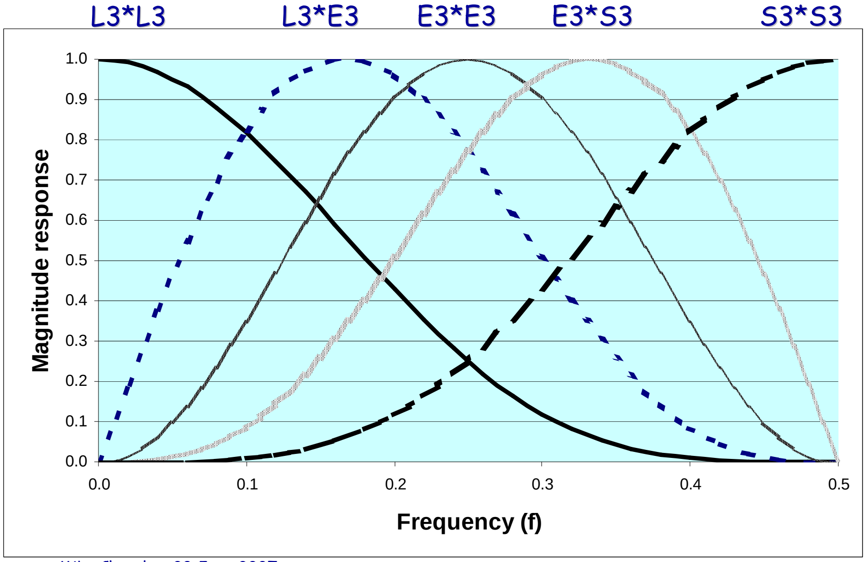
$H_{E5}(\omega) = 1.e^{-2j\omega} + 0.e^{-j\omega} - 2.\ e^{0} + 0.e^{j\omega} - 1.e^{2j\omega}$

$= (\cos 2\omega - j\sin 2\omega)\ \ -2 + (\cos 2\omega + j\sin 2\omega)$

$= 2\cos 2\omega - 2 = -4\sin^2\omega = (2j\sin\omega)^2$



Laws Filters (frequency response)

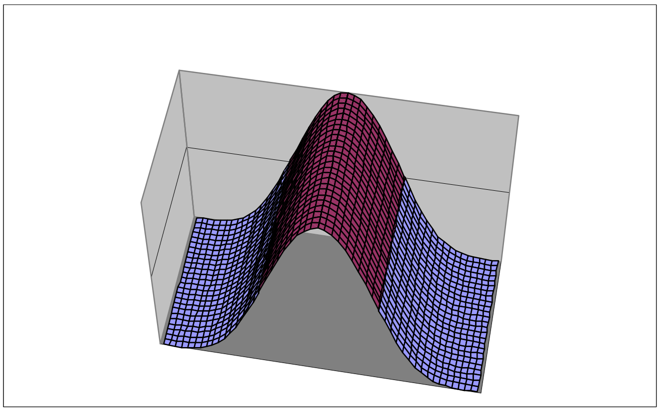


Texture, Mike Chantler, 28 June 2007

page 47

Laws: 2D frequency responses

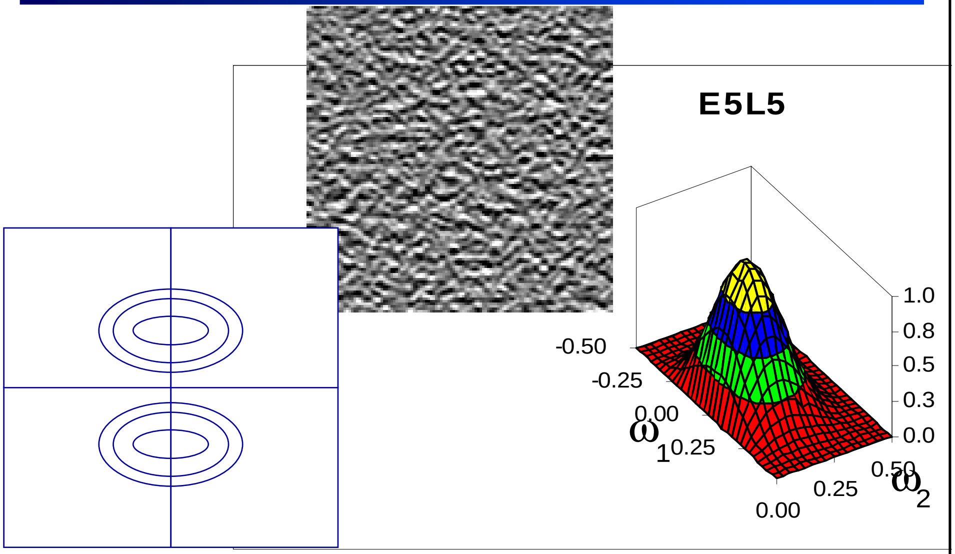
L5*1



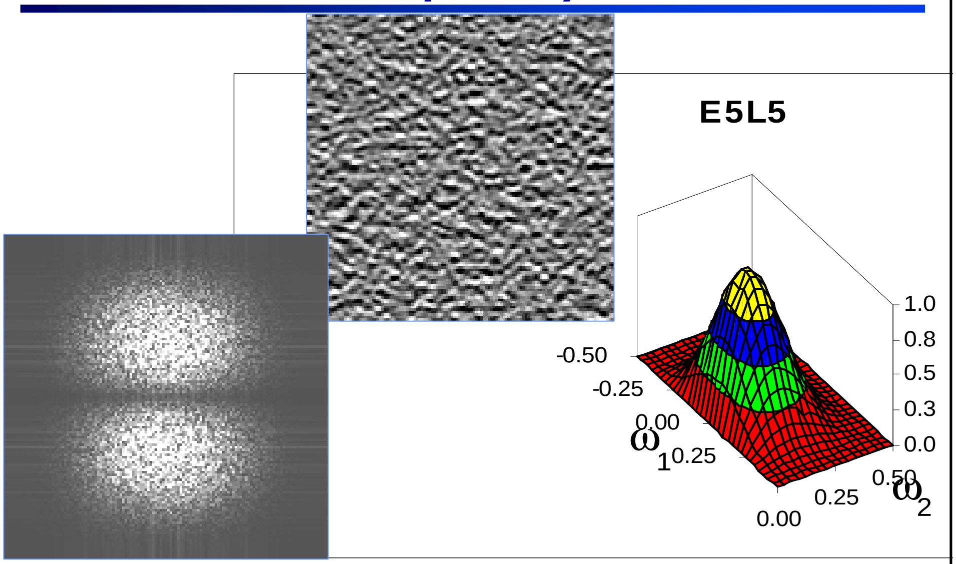
Texture, Mike Chantler, 28 June 2007

page 48

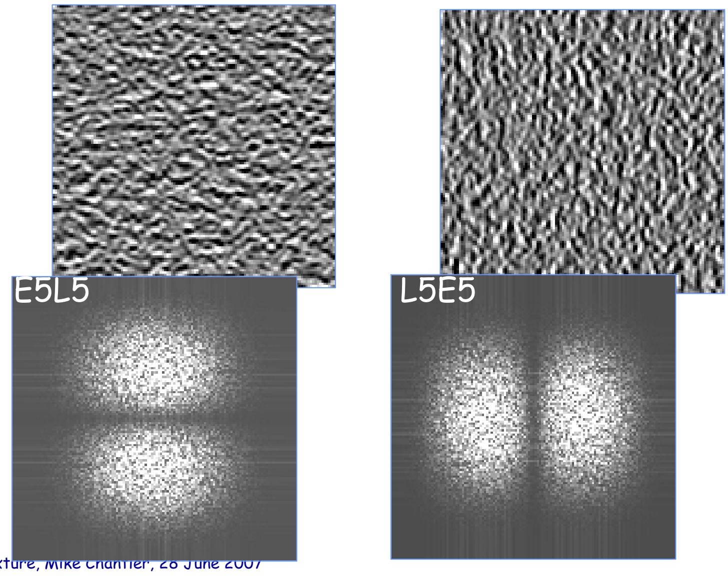
Laws Filters (2D frequency response)



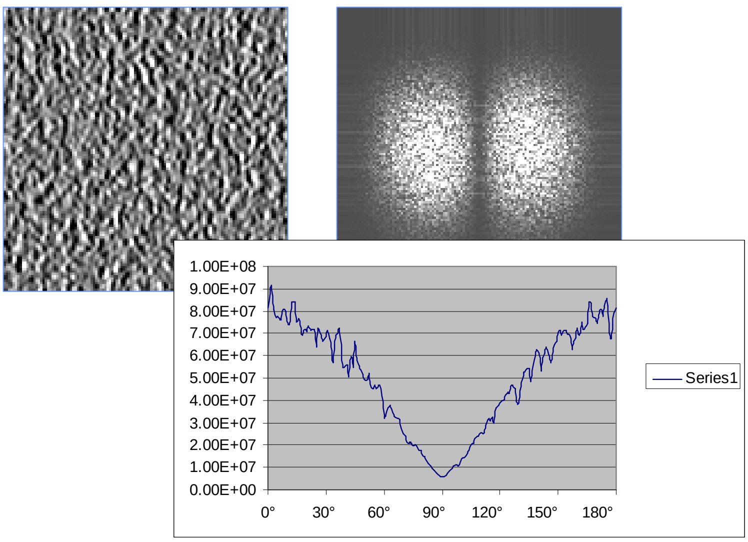
Laws Filters (2D frequency response)



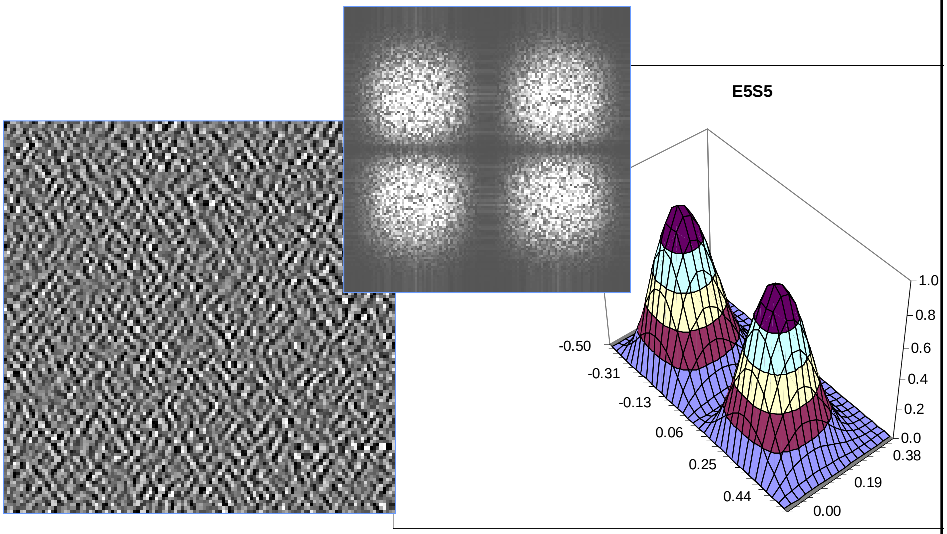
Laws Filters (2D frequency response)



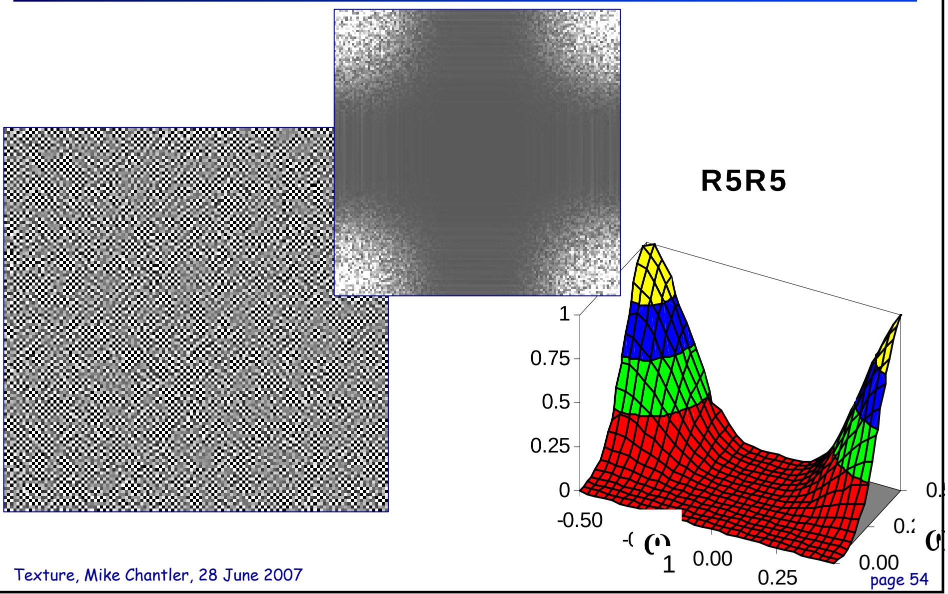
Laws Filters (2D frequency response)

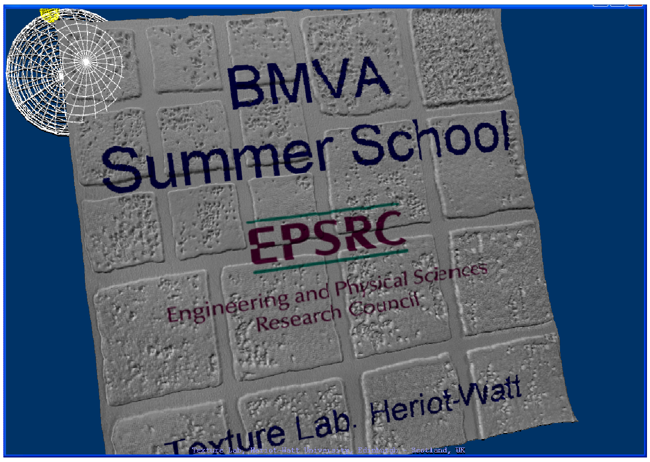


Laws Filters (2D frequency response)



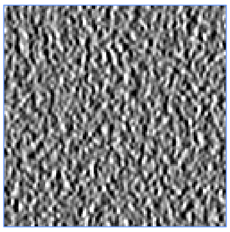
Laws Filters (2D frequency response)

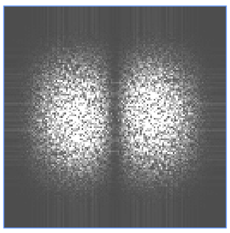




Local variance estimation

‘Energy’ or variance estimators



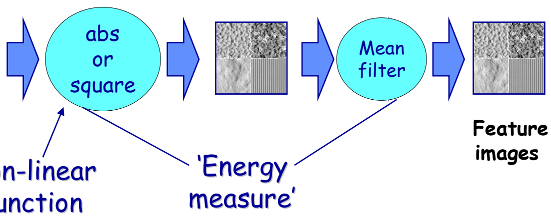


Laws L5E5

$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$

or for a cheaper feature:

$$f_k(x,y) = \xi_L\left\{\left|h_k(r,s) * i(x,y)\right|\right\}$$

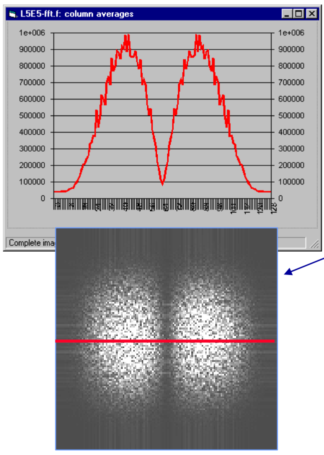


Texture, Mike Chantler, 28 June 2007

page 56

‘Energy’ or variance estimators

$$f_k = \sum_{x,y}\left\{(h_k(r,s)*i(x,y))^2\right\} = \sum_{u,v}\left|H_k(u,v)I(u,v)\right|^2$$

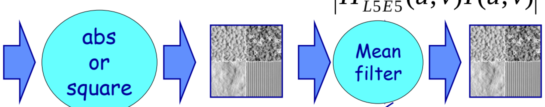


$H_{L5E5}(u,v)I(u,v)$

Integrate volume under the magnitude fft

(Parseval’s theorem)

$|H_{L5E5}(u,v)I(u,v)|^2$



Feature images

Non-linear function

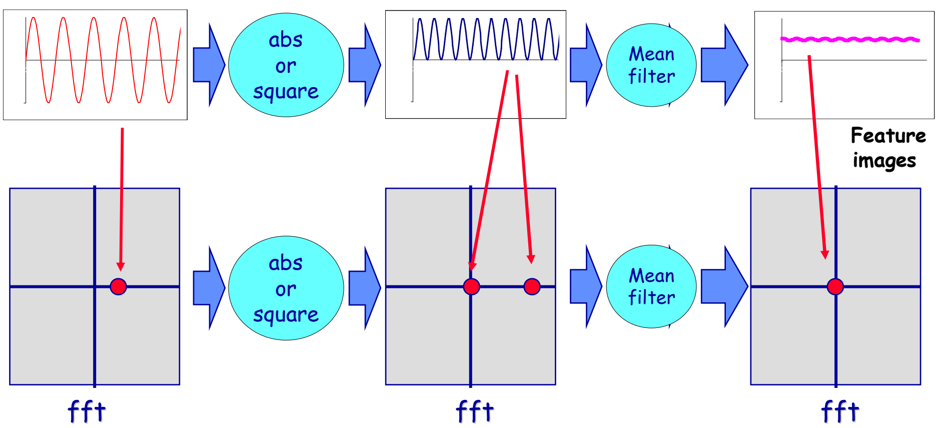
‘Energy measure’

Texture, Mike Chantler, 28 June 2007

page 57

Variance estimation: Single sinewave example

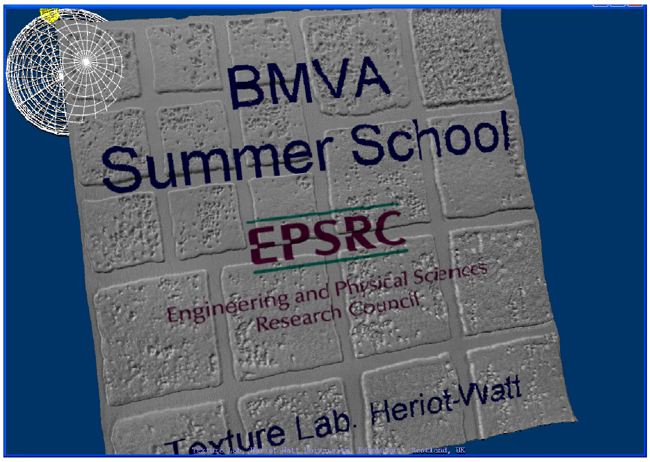
Spatial domain $f_k(x,y) = \xi_L\left\{(h_k(r,s)*i(x,y))^2\right\}$



Frequency domain $\sum_{u,v}\left|H_k(u,v)I(u,v)\right|^2$

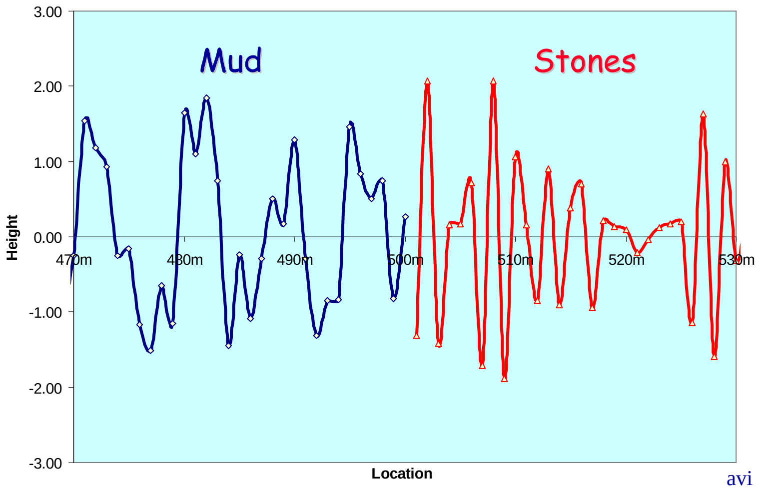
Texture, Mike Chantler, 28 June 2007

page 58



1D example

What does a local variance estimator do?

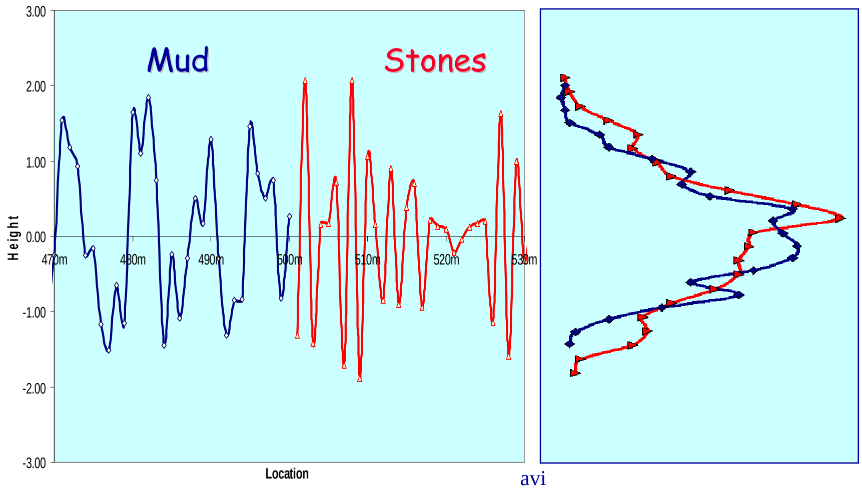


avi

Texture, Mike Chantler, 28 June 2007

page 60

1D-data example: distributions

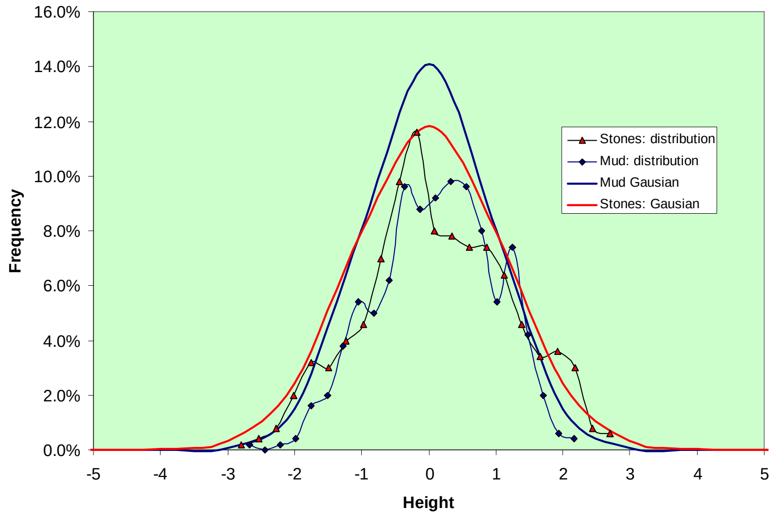


avi

Texture, Mike Chantler, 28 June 2007

page 61

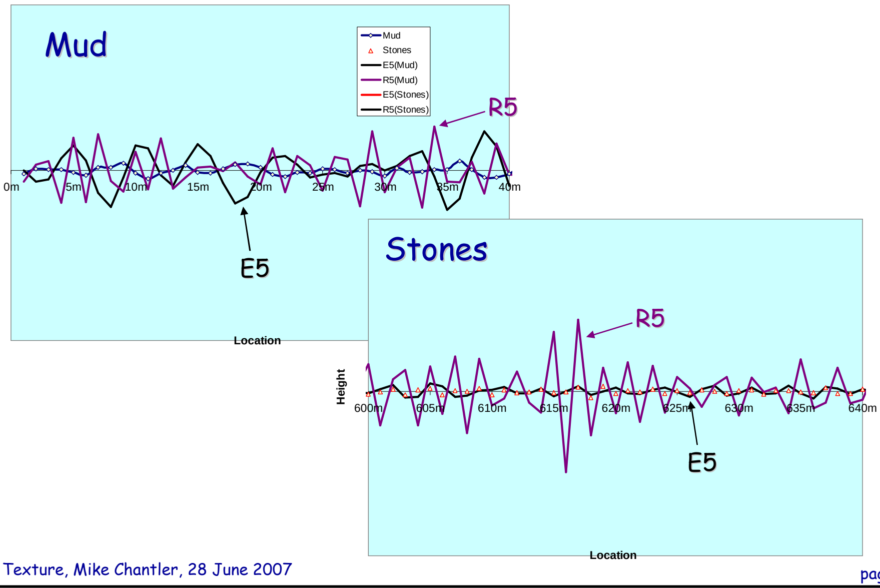
1D-data example: distributions



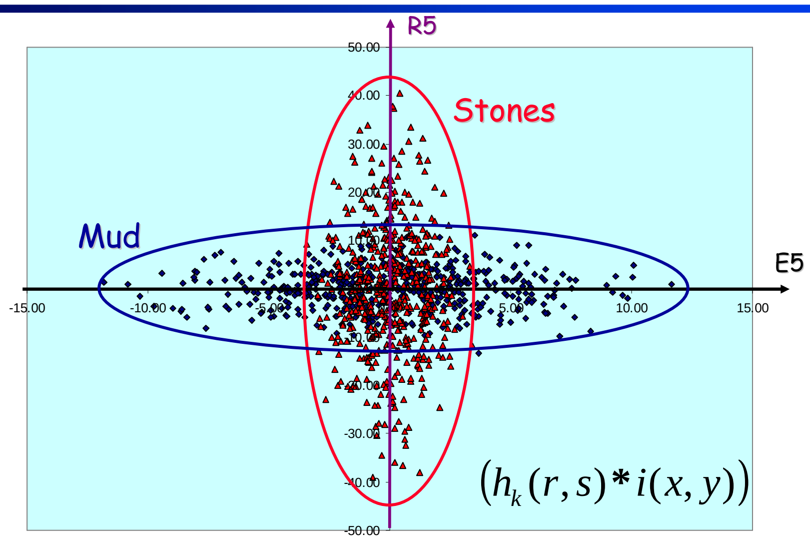
Texture, Mike Chantler, 28 June 2007

page 62

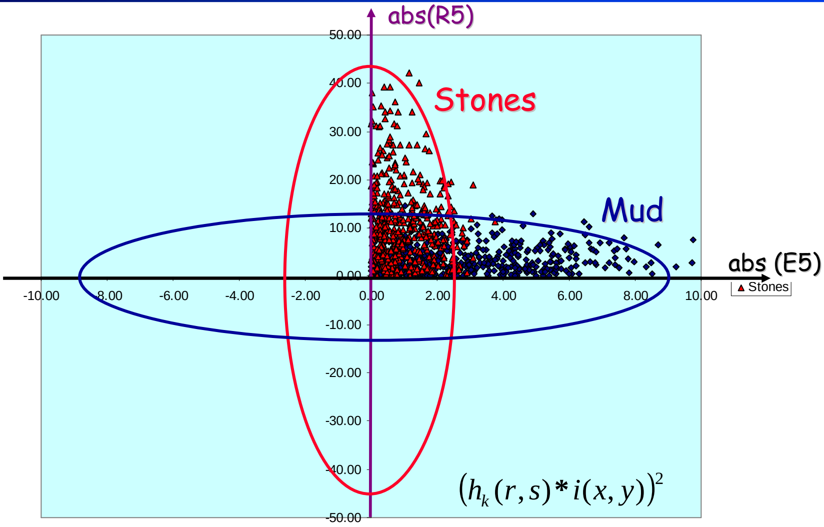
1D-data example: E5, R5 filters



1D-data example: E5, R5 scatter plot



1D-data example: abs(E5), abs(R5) scatter plot

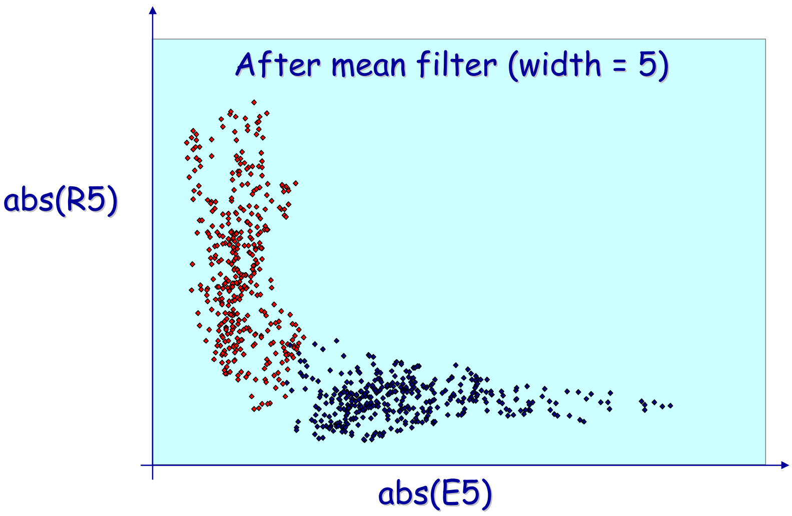


$$\left(h_k(r,s) * i(x,y)\right)^2$$

Texture, Mike Chantler, 28 June 2007

page 65

1D-data example: abs(E5), abs(R5) scatter plot

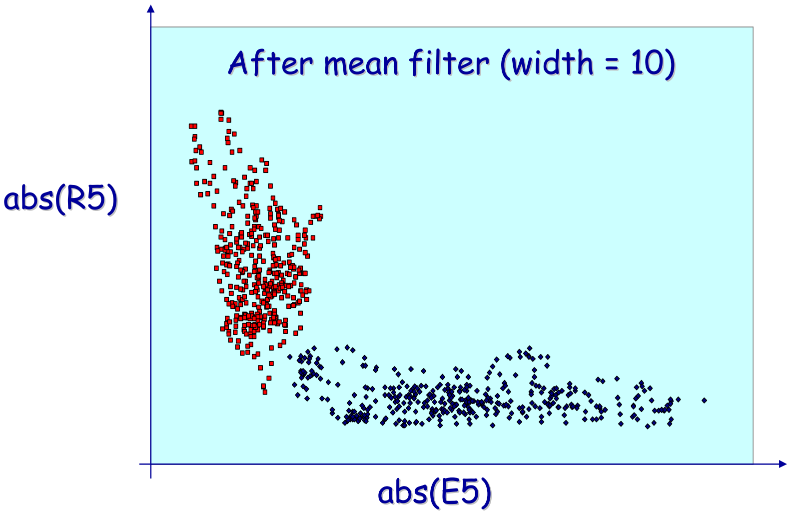


$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$

Texture, Mike Chantler, 28 June 2007

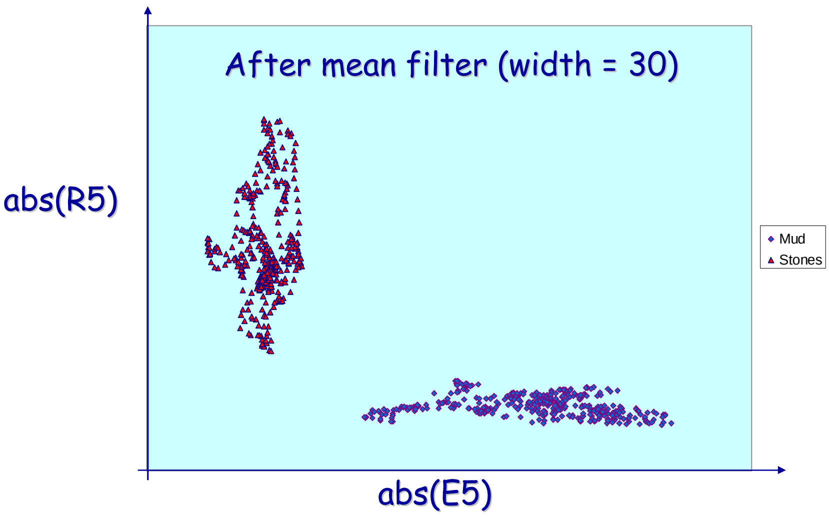
page 66

1D-data example: abs(E5), abs(R5) scatter plot



$$f_k(x,y) = \xi_L\{(h_k(r,s) * i(x,y))^2\}$$

1D-data example: abs(E5), abs(R5) scatter plot

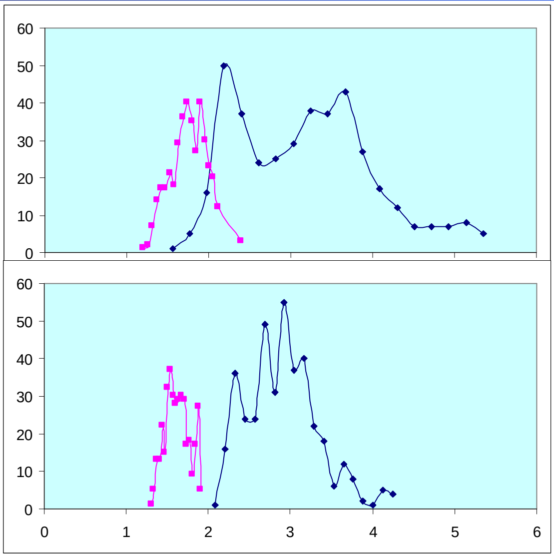


$$f_k(x,y) = \xi_L\{(h_k(r,s) * i(x,y))^2\}$$

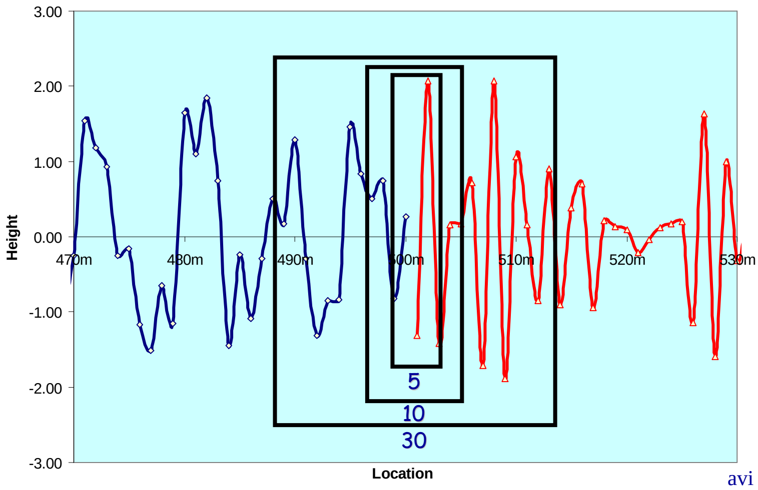
1D-data example: abs(E5) distributions

Mean [5]

Mean [30]



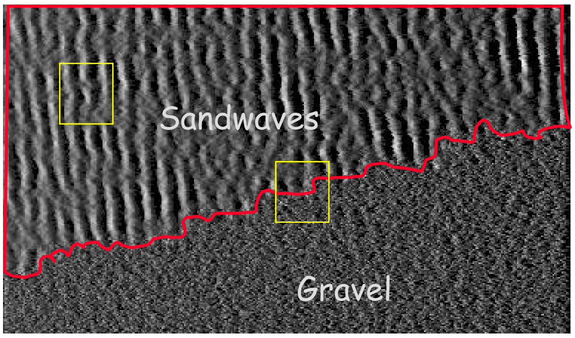
1D-data example: mean window size



avi

$$f_k(x,y) = \xi_L \left\{ (h_k(r,s) * i(x,y))^2 \right\}$$

Widow size?

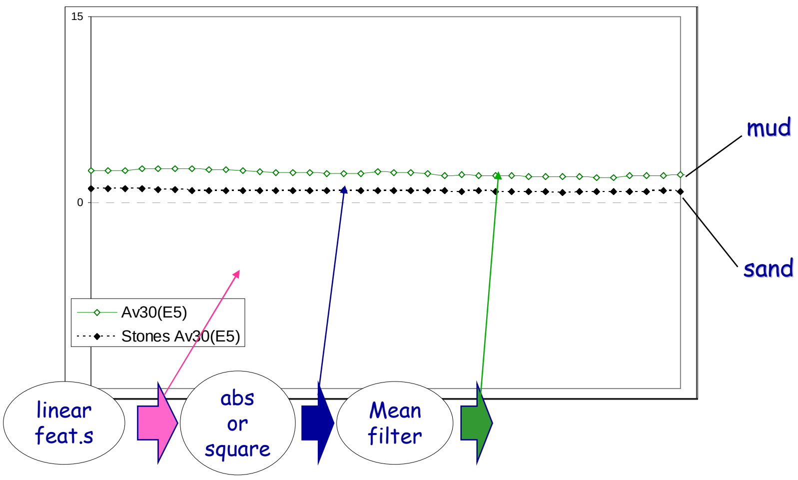


F1	2.65
F2	9.75
F3	2340
F4	22.3

- **Just large enough to ensure class separation?**

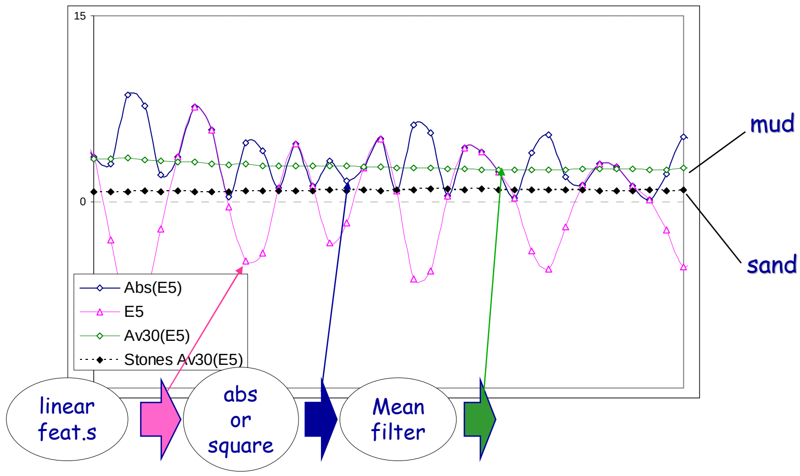
1D-data example: feature process

$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$



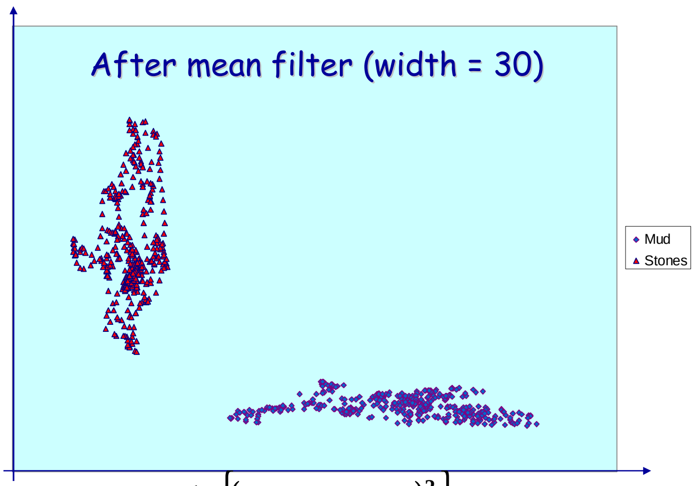
1D-data example: feature process

$$f_k(x,y) = \xi_L\left\{\left(h_k(r,s) * i(x,y)\right)^2\right\}$$



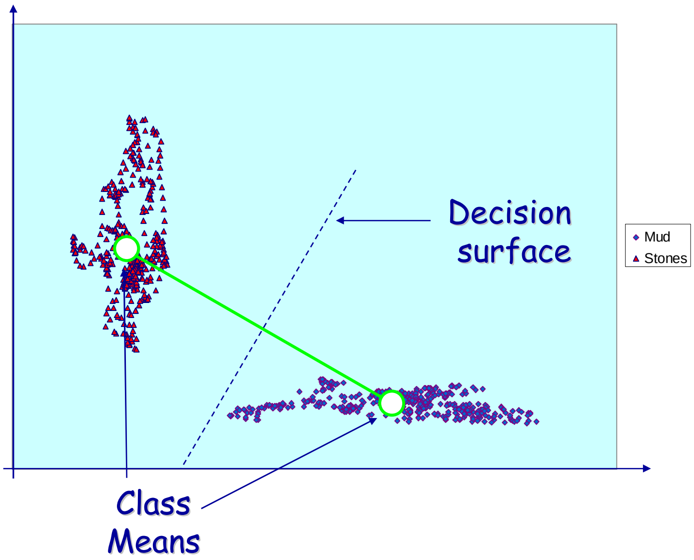
1D-data example: abs(E5), abs(R5) scatter plot

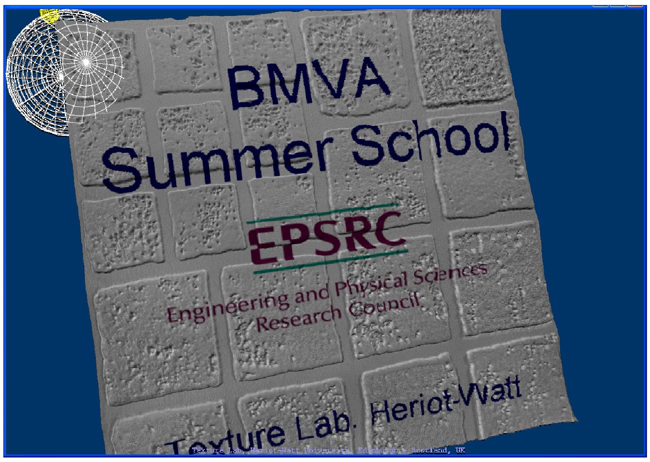
$$\xi_L\left\{\left(h_{R5}(r) * i(x)\right)^2\right\}$$



$$\xi_L\left\{\left(h_{E5}(r) * i(x)\right)^2\right\}$$

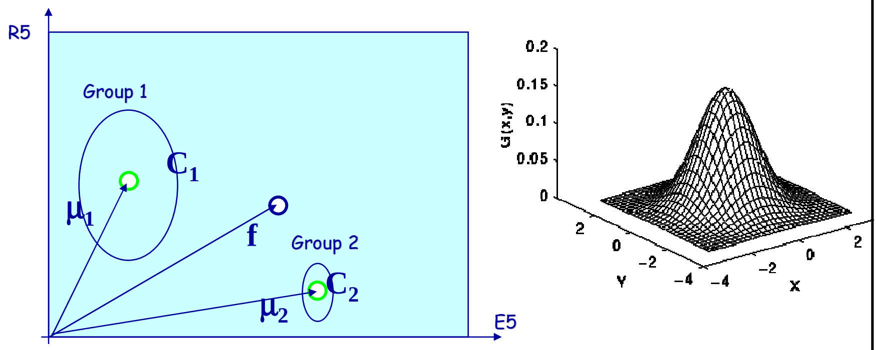
Minimum (mean) distance classifier





Bayes classifiers

Modelling 2D normal distributions

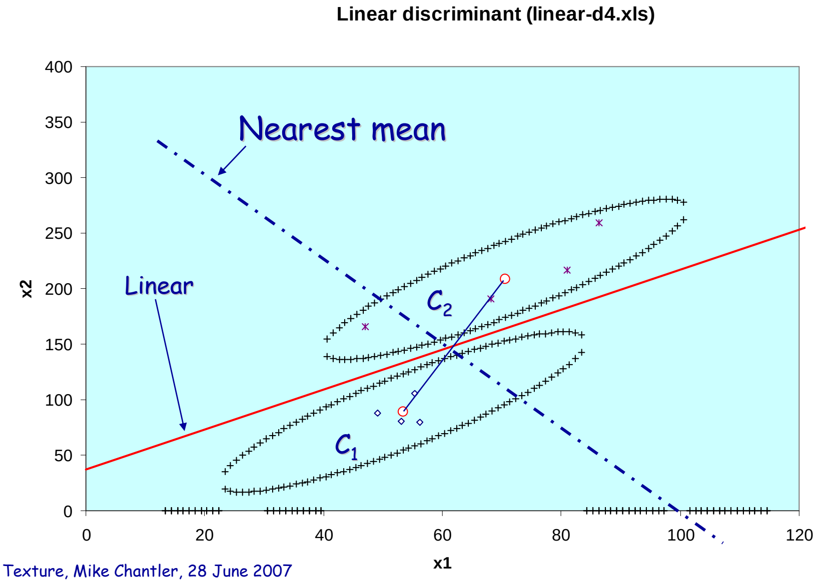


C_i = covariance matrix for group (class) i
μ_i= mean vector for group I
f = feature vector (E5, R5) of sample to be classified

Texture, Mike Chantler, 28 June 2007

page 77

Bayes linear & nearest (mean) classifiers

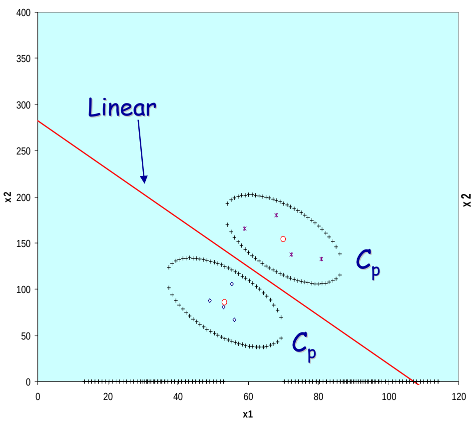


Texture, Mike Chantler, 28 June 2007

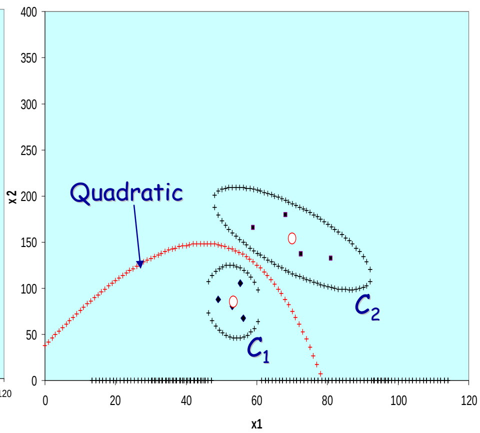
page 78

Quadratic and linear classifiers

Linear classifier

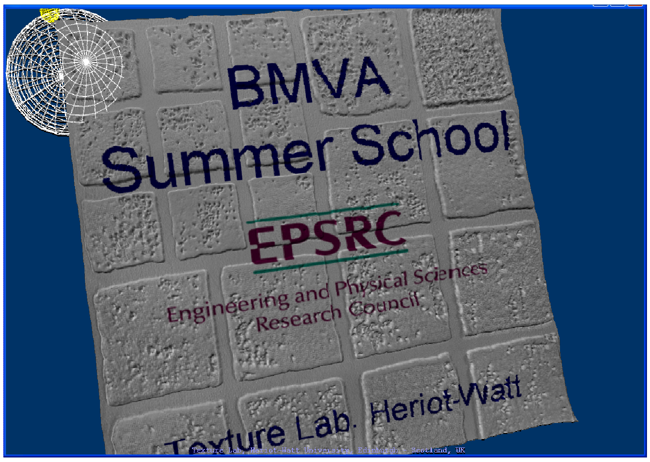


Quadratic classifier

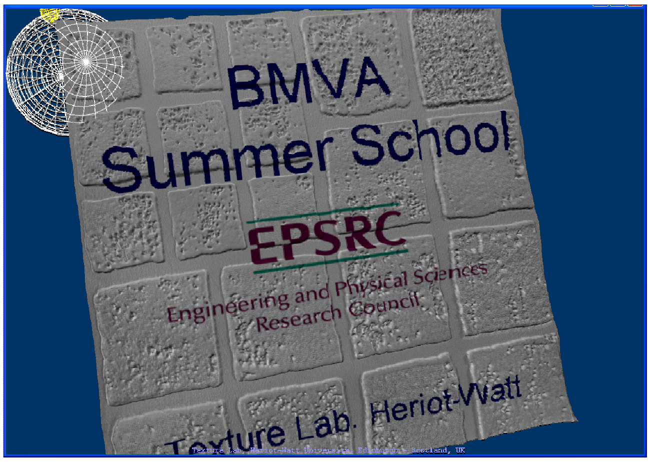


Supervised/unsupervised classification

- **Supervised classification**
  - uses pre-classified training data
- **Unsupervised**
  - No labeled training data



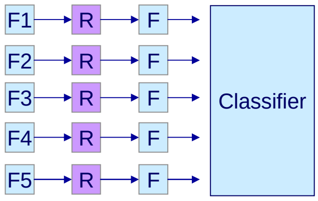
2D result



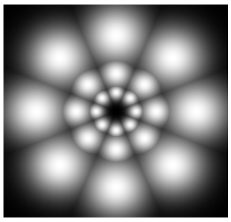
Linear features summary

Filter banks

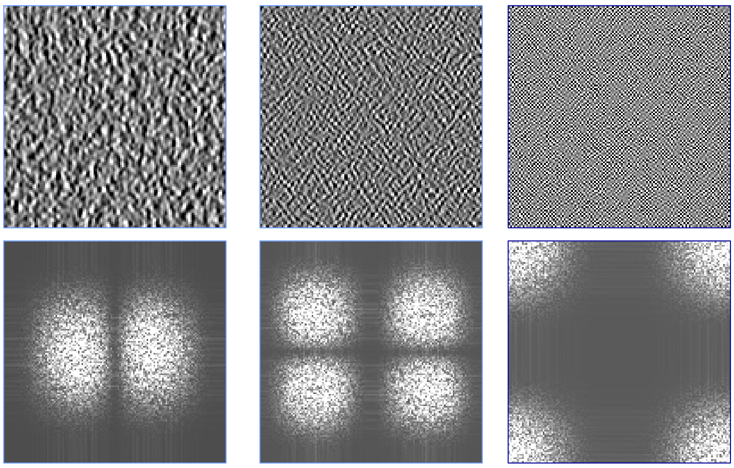
- **Terminology**
  - Filter banks
  - FIRs (Finite Impulse Response)
  - Non-recursive filters
  - Frequency channel model
  - Power spectrum measures
  - Bandpass filters
  - FRF (Filter Rectify Filter – from psychophysics)



- **Types**
  - Steerable filters and other pyramids
  - Gabors
  - Laws
  - Wedge/ring filters

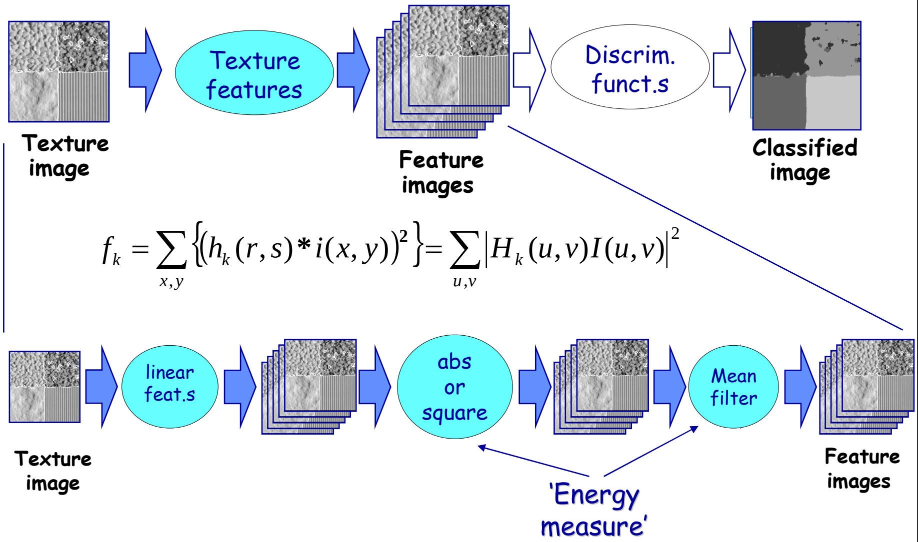


Example: Laws Filters



L5E5 E5S5 R5R5

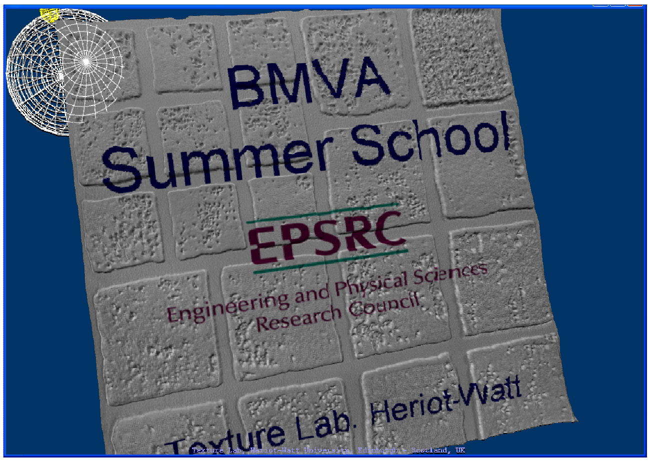
Linear features: summary



$$f_k = \sum_{x,y} \left\{ \left(h_k(r,s) * i(x,y) \right)^2 \right\} = \sum_{u,v} \left| H_k(u,v) I(u,v) \right|^2$$

Texture, Mike Chantler, 28 June 2007

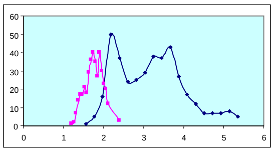
page 86



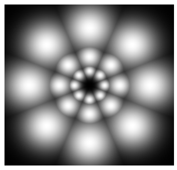
Other Features

Texture features

- **Linear Filters**
  - Other marginal statistics and moments
    - Histograms
      - raw bins
      - mixture of Gaussians etc
    - Skew, Kurtosis..
  - Joint statistics

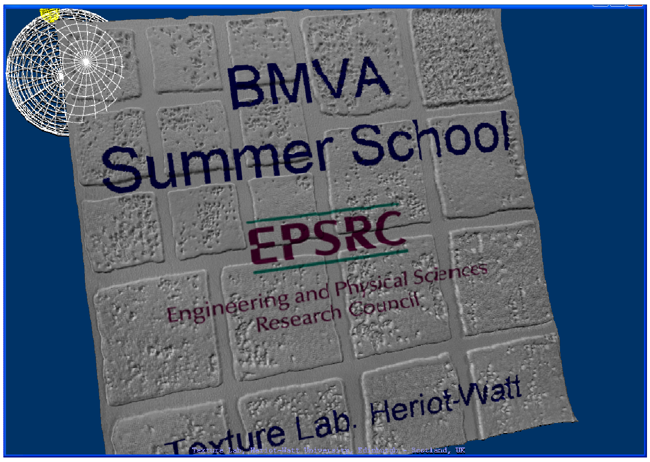


- **Other – modeling local neighbourhoods**
  - Morphological operators
  - Local binary patterns
  - Co-occurence matrices
  - MRF
  - Vector Quantisation



Texture, Mike Chantler, 28 June 2007

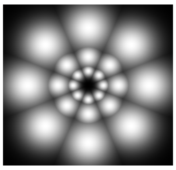
page 88



A Simple Strategy

A Simple Strategy

- **For classification and/or segmentation**
- **Start with a simple set of linear filters**
  - Use with a Bayes linear discriminant
  - Do step-wise feature selection
  - Repeat with increased feature pool if not successful



- **Add non-linear features as a last (or later) resort**
- **Formalise linear features in the frequency domain**

Texture, Mike Chantler, 28 June 2007

page 90

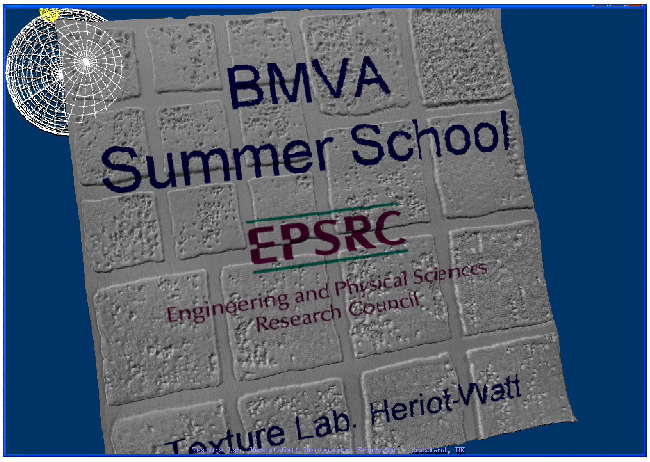
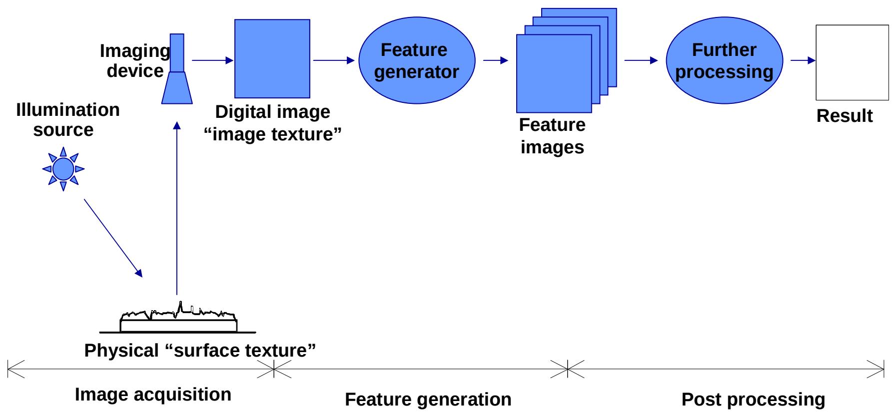


Image texture problems

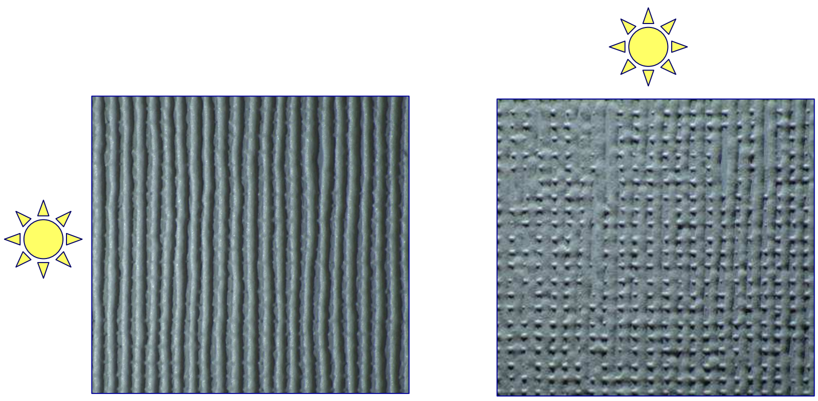
Capturing Image Texture



Texture, Mike Chantler, 28 June 2007

page 92

Image texture problems

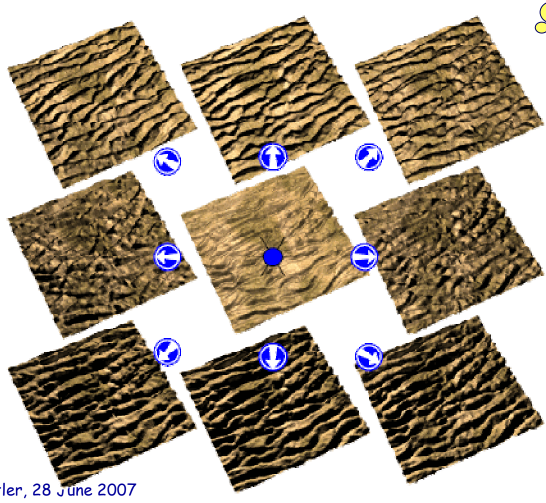


Texture, Mike Chantler, 28 June 2007

page 93

Images of surface texture affected by illumination

how?

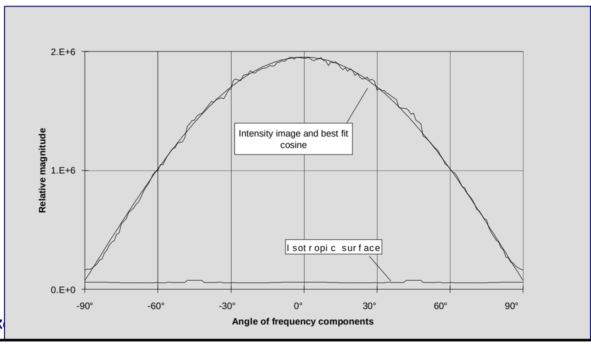


Linear shading (Fourier domain)

Surface → Image formation

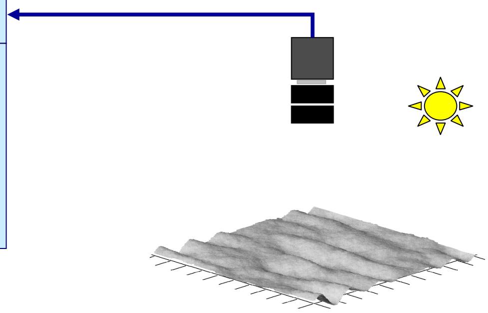
$$I(\omega,\theta)=\omega^2\cos^2(\theta-\tau)\sin^2(\sigma)S(\omega,\theta)$$

[Kube88, Chantler95, Koenderink03, Barsky03]



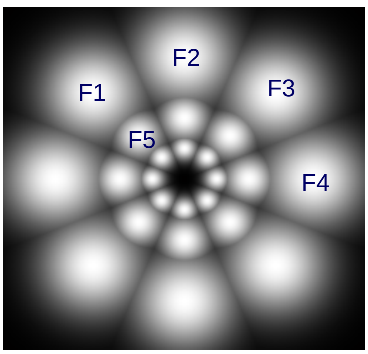
Frequency domain model

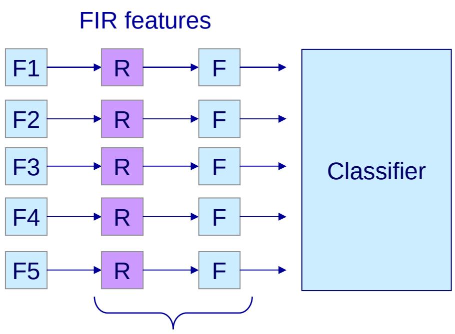
Image spatial domain	Power Spectrum (contours)
Power Spectrum (polar plot)	Power Spectrum (relief)



$$I(\omega,\theta) = \omega^2 \cos^2(\theta-\tau)\sin^2(\sigma)S(\omega,\theta)$$

How are the Filter Banks affected?

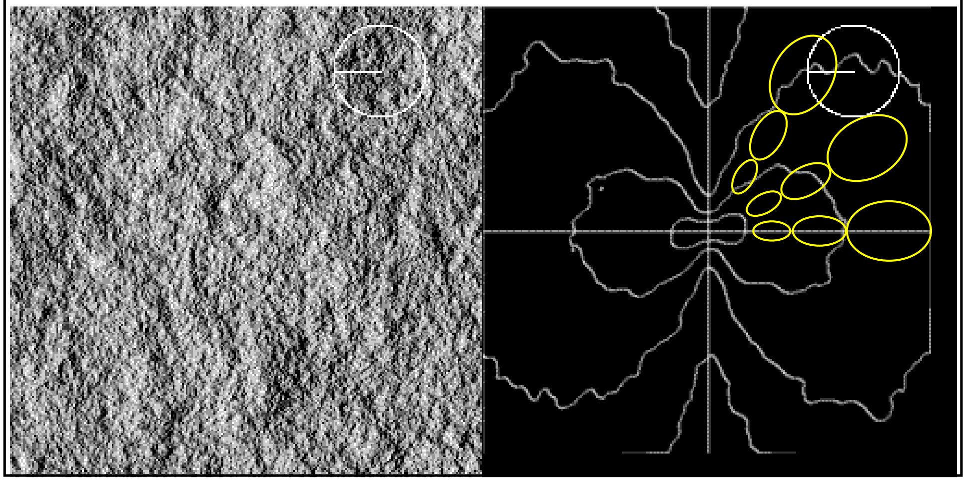




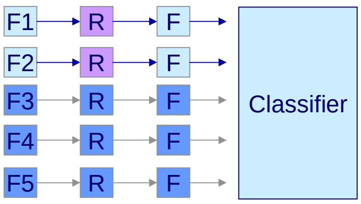
Rectification + averaging
=
local variance measure

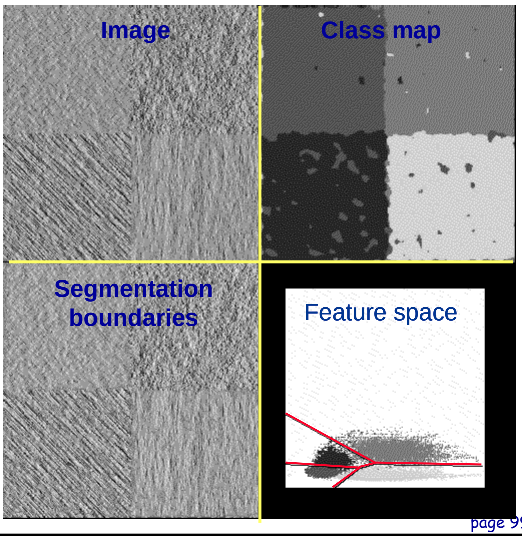
[Randen & Husoy, PAMI, 1999]

Frequency domain model

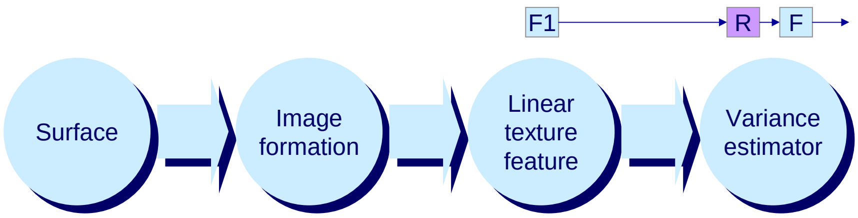


Effect of changing illumination on classification



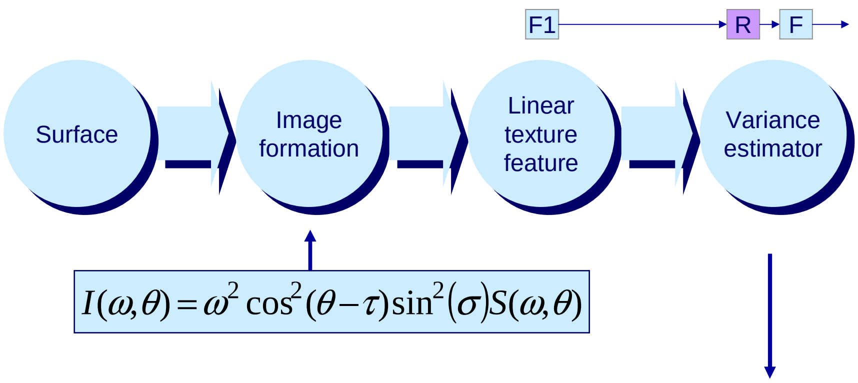


Texture feature behaviour



$$I(\omega,\theta) = \omega^2 \cos^2(\theta - \tau)\sin^2(\sigma) S(\omega,\theta)$$

Texture feature behaviour



$$I(\omega,\theta) = \omega^2 \cos^2(\theta - \tau)\sin^2(\sigma) S(\omega,\theta)$$

$$f(\tau) = a + b\cos 2\tau + c\sin 2\tau$$

a, b, c = fn(surface, filter, illuminant slant)

Texture feature behaviour

$$a = \frac{1}{2}\sin^2\sigma \int_0^\infty \omega^3 \int_0^{2\pi} [A(\omega,\theta)] d\omega d\theta$$

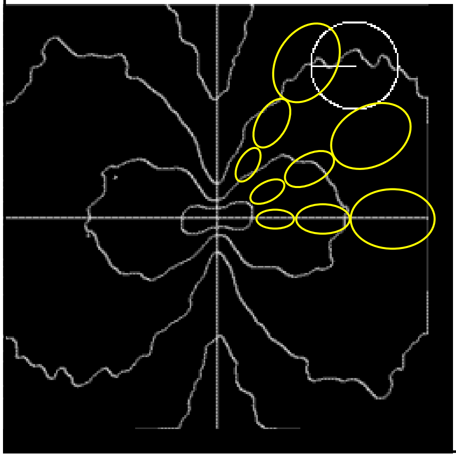
$$b = \frac{1}{2}\sin^2\sigma \int_0^\infty \omega^3 \int_0^{2\pi} [A(\omega,\theta)\cos 2\theta] d\omega d\theta$$

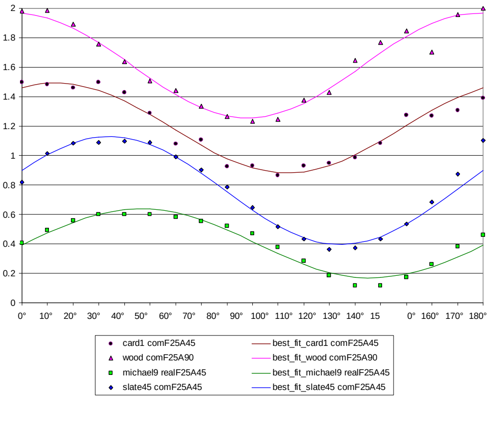
$$c = \frac{1}{2}\sin^2\sigma \int_0^\infty \omega^3 \int_0^{2\pi} [A(\omega,\theta)\sin 2\theta] d\omega d\theta$$

$$f(\tau) = a + b\cos 2\tau + c\sin 2\tau$$

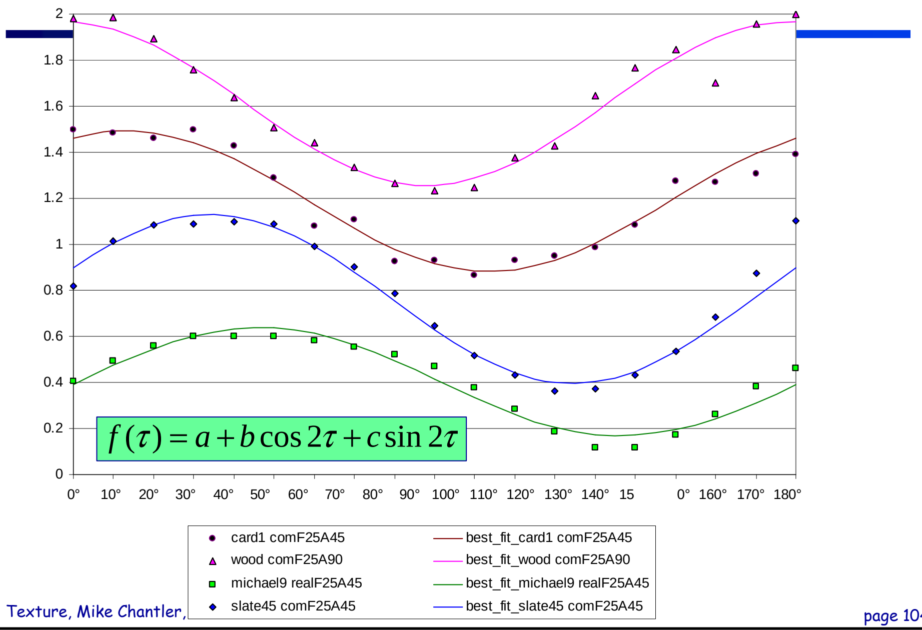
Texture, Mike Chantler, 28 June 2007

Texture Feature Behaviour





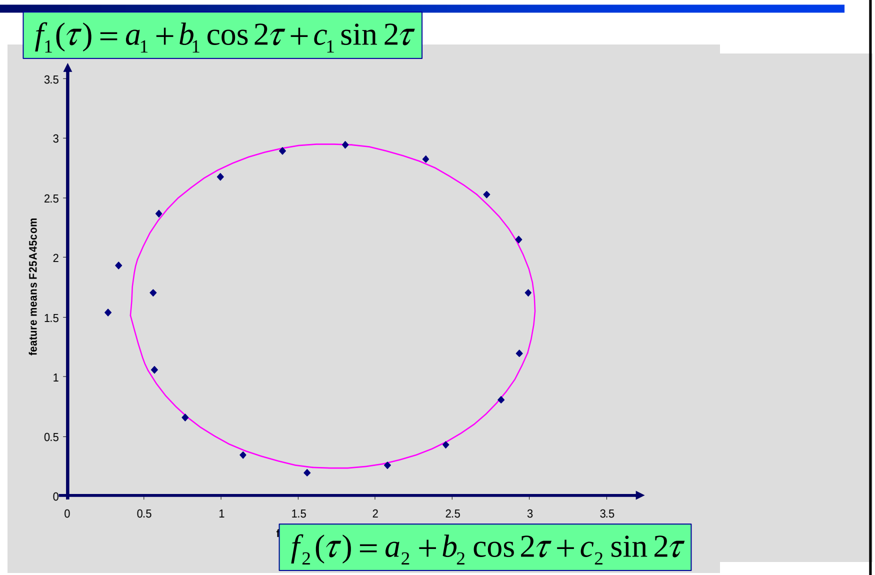
Texture feature behaviour



Texture, Mike Chantler,

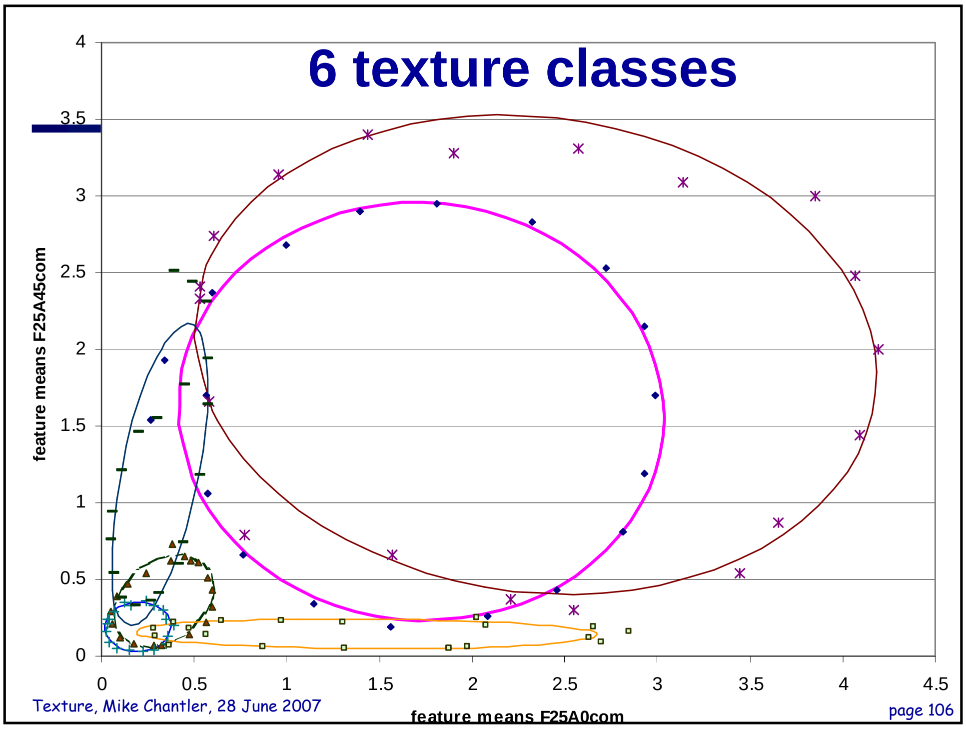
page 104

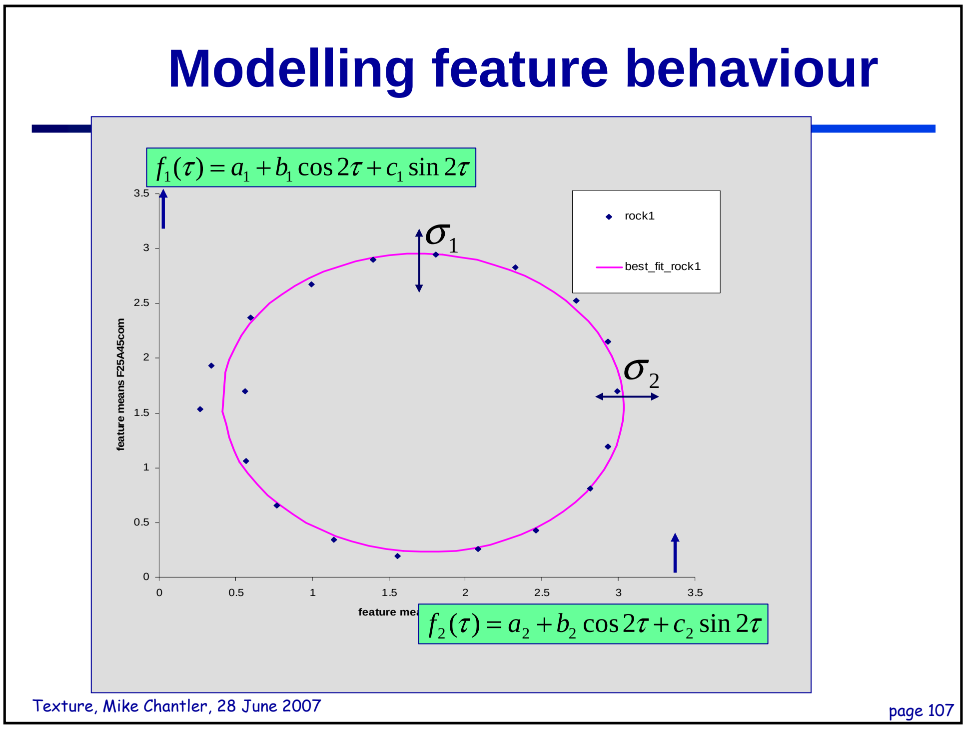
Behaviour in a 2D feature space

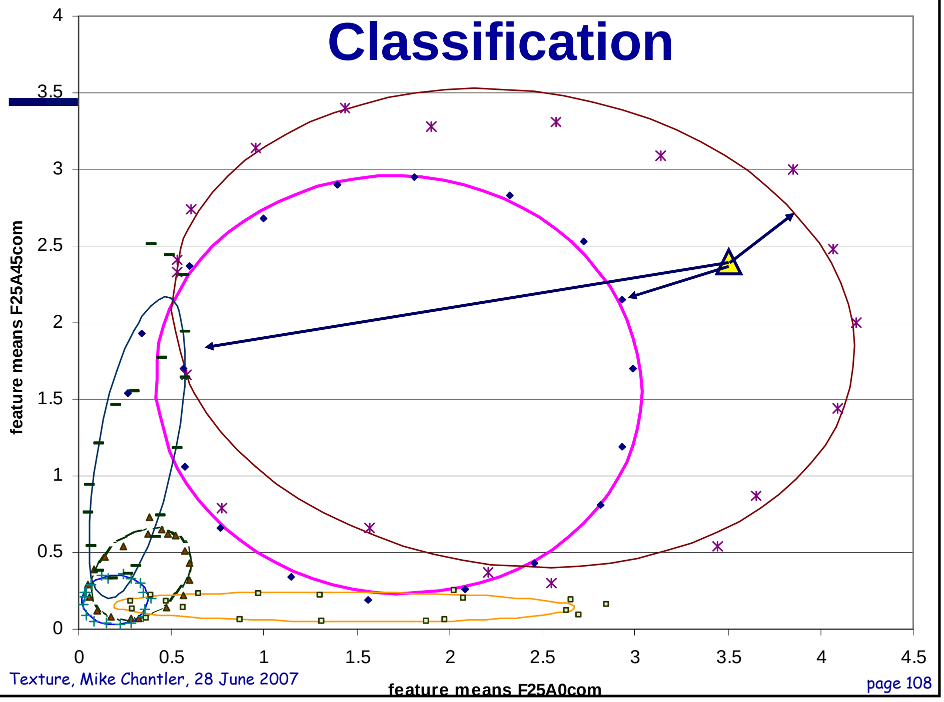


Texture, Mike Chantler, 28 June 2007

page 105


6 texture classes
feature means F25A45com
feature means F25A0com
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Texture, Mike Chantler, 28 June 2007
page 106


Modelling feature behaviour
$f_1(\tau) = a_1 + b_1 \cos 2\tau + c_1 \sin 2\tau$
$f_2(\tau) = a_2 + b_2 \cos 2\tau + c_2 \sin 2\tau$
σ_1
σ_2
rock1
best_fit_rock1
feature means F25A45com
feature mea
0
0.5
1
1.5
2
2.5
3
3.5
Texture, Mike Chantler, 28 June 2007
page 107



Texture, Mike Chantler, 28 June 2007

page 108

Data and Features

Features

Gabor + Laws filters

Data

30 textures

Imaged at illumiminat tilt angles of 0°-180°(10°-15°) steps

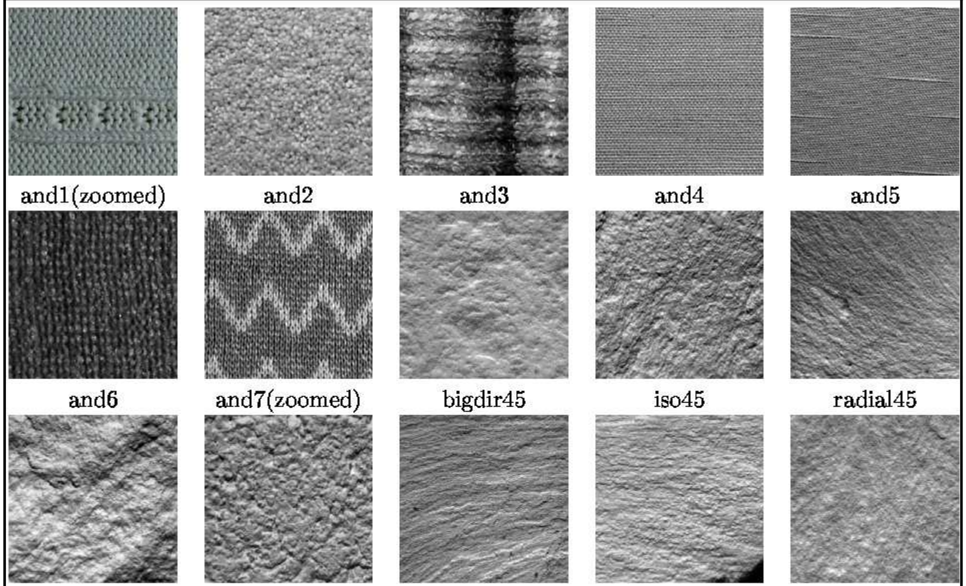
About **400** 512x512 8-bit monochrome images

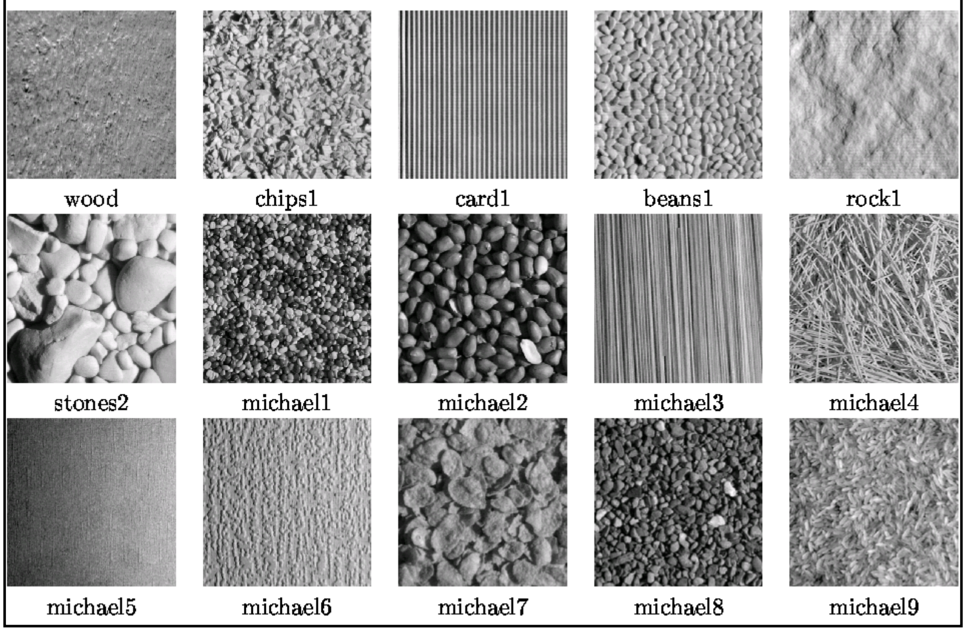
Training data

Every other image

Texture, Mike Chantler, 28 June 2007

page 109


and1(zoomed)
and2
and3
and4
and5
and6
and7(zoomed)
bigdir45
iso45
radial45

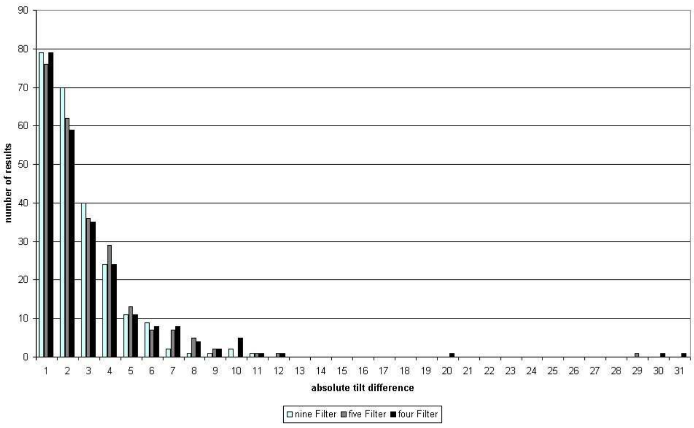

wood
chips1
card1
beans1
rock1
stones2
michael1
michael2
michael3
michael4
michael5
michael6
michael7
michael8
michael9

Results

Number of filters	6	5	4
Error rate	0.8%	2.0%	3.8%

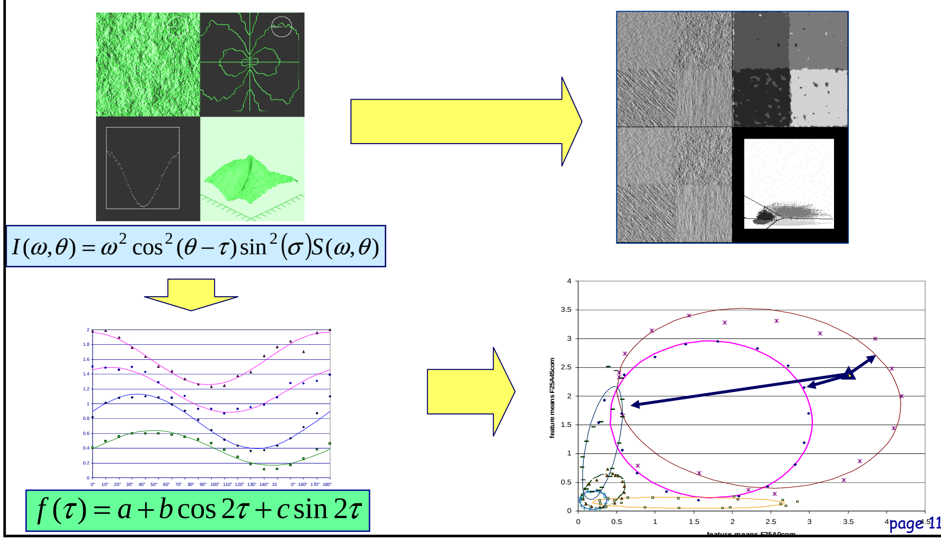
Illuminant tilt angle estimation
Error distribution

Number of errors



Absolute error

Analysis - summary

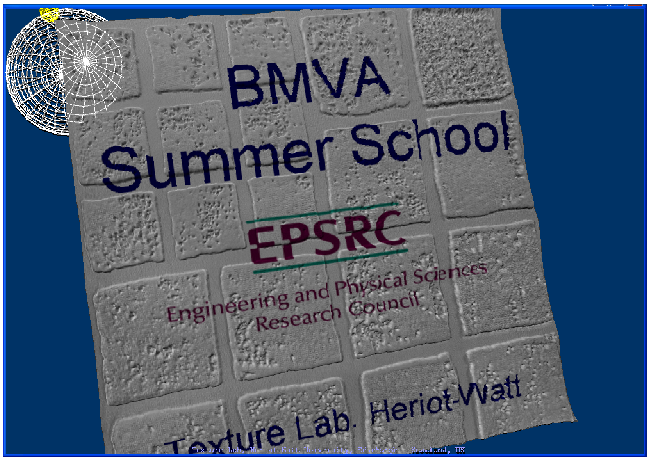


$$I(\omega,\theta) = \omega^2 \cos^2(\theta-\tau)\sin^2(\sigma)S(\omega,\theta)$$

$$f(\tau) = a + b\cos 2\tau + c\sin 2\tau$$

Other approaches

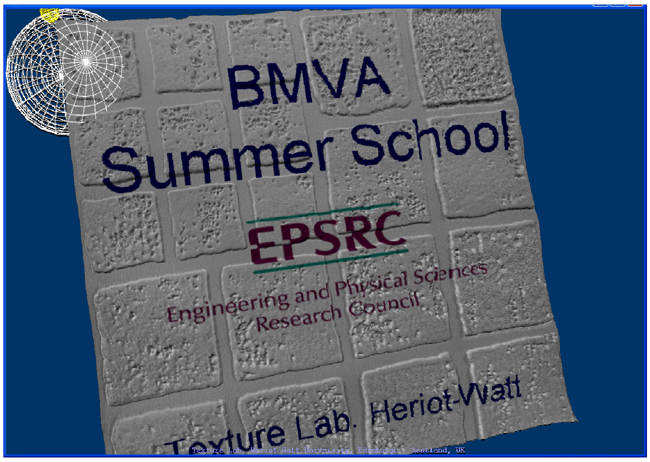
- **Varma & Zisserman**
  - Code book (vector quantisation) of local neigbourhoods
- **Petrou & Barsky**
  - Spatial domain model
  - Copes with albedo variation
- **Dana, Nayer, van Ginneken and Koenderink**
  - established the Columbia-Utrecht database of real world surface



Other applications

Other applications

- **Texture synthesis**
  - Quilting
    - Efros, Paget etc.
  - Parametric
    - Portilla & Simoncelli
- **Defect detection**
  - Xie & Mirmehdi.
- **Shape from Texture**
  - Blake
- **Modelling and Rendering**
  - Bidirectional Texture Functions
  - (per-pixel BRDF)



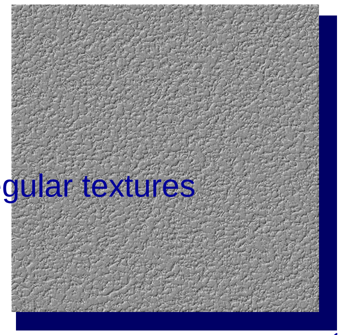
Summary

Summary

- What is Texture?
- Computer Vision ‘Texture’?
- Texture Features
- Linear Filter based Features
- Local Variance Estimation
- 1D Example & 2D Result
- Other Features
- Image Texture Problems
- Other Applications

Wots hot & wots not

- **Mature areas**
  – Classification/ segmentation using linear features
  – Texture synthesis via quilting
  – Shape from texture
  – Data driven classifiers under changing environmental conditions
- **Open Questions**
  – Phase related features
  – Psychophysics of texture
    - How many dimensions etc.
  – Parametric synthesis of non-regular textures
  – BTF compression



Texture, Mike Chantler, 28 June 2007

page 120

Thanks to

- **Funding**
  – EPSRC, CEC, SMI
- **Folk**
  – Ged McGunnigle, Andy Spence, Ondřej Drbohlav, Khem Emrith, Stefano Padilla, Patrick Green, Alasdair Clarke, Pratik Shah, Jerry Wu, Cristina Gullon, Laurie Linnett

Texture, Mike Chantler, 28 June 2007

page 121