iccMAX in Barbieri

Michele Conni
R&D Engineer at Barbieri Electronic
Ph.D. candidate at NTNU

Outline

1. Barbieri
2. What is texture?
3. Texture models
4. Roughness correction in iccMAX

© MICHELE CONNI

1. Barbieri

Provincia di Bolzano, Italy

15 employees
Service center in America, Europe and Asia

R&D, Assembling
Quality control

Advanced patented technology

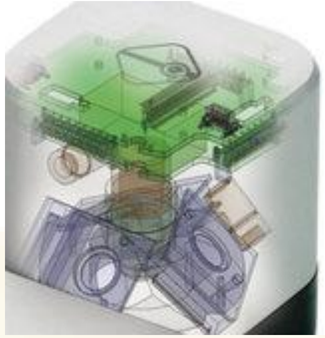
Products

Barbieri spectrophotometers

Spectro LFP qb for Large Format Printing

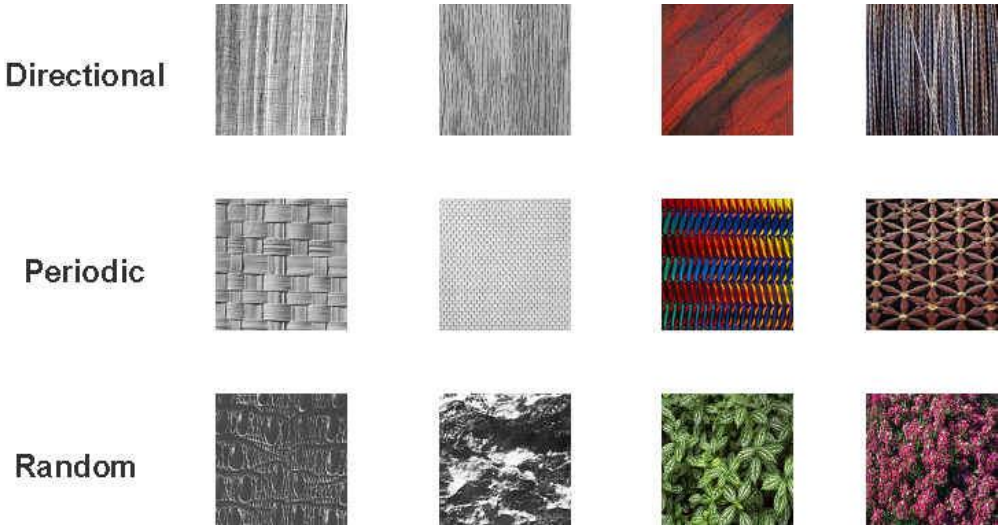
Spectro Pad as portable solution for roll-to-roll Format Printing

Spectro Swing for calibration in roll-to-roll-Format Printing



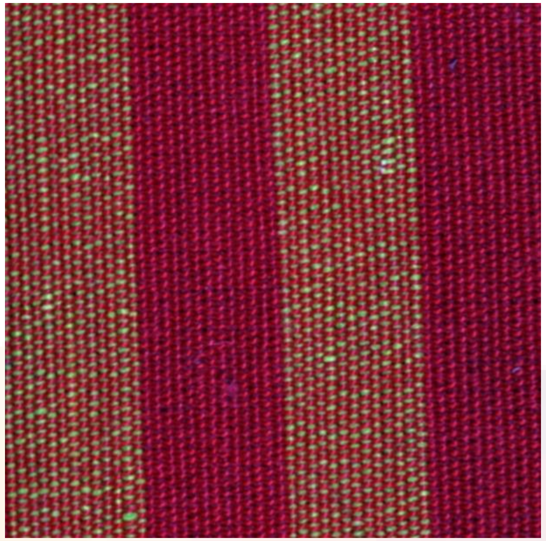
Customized measuring devices for OEM-manufacturers

2. What is texture?

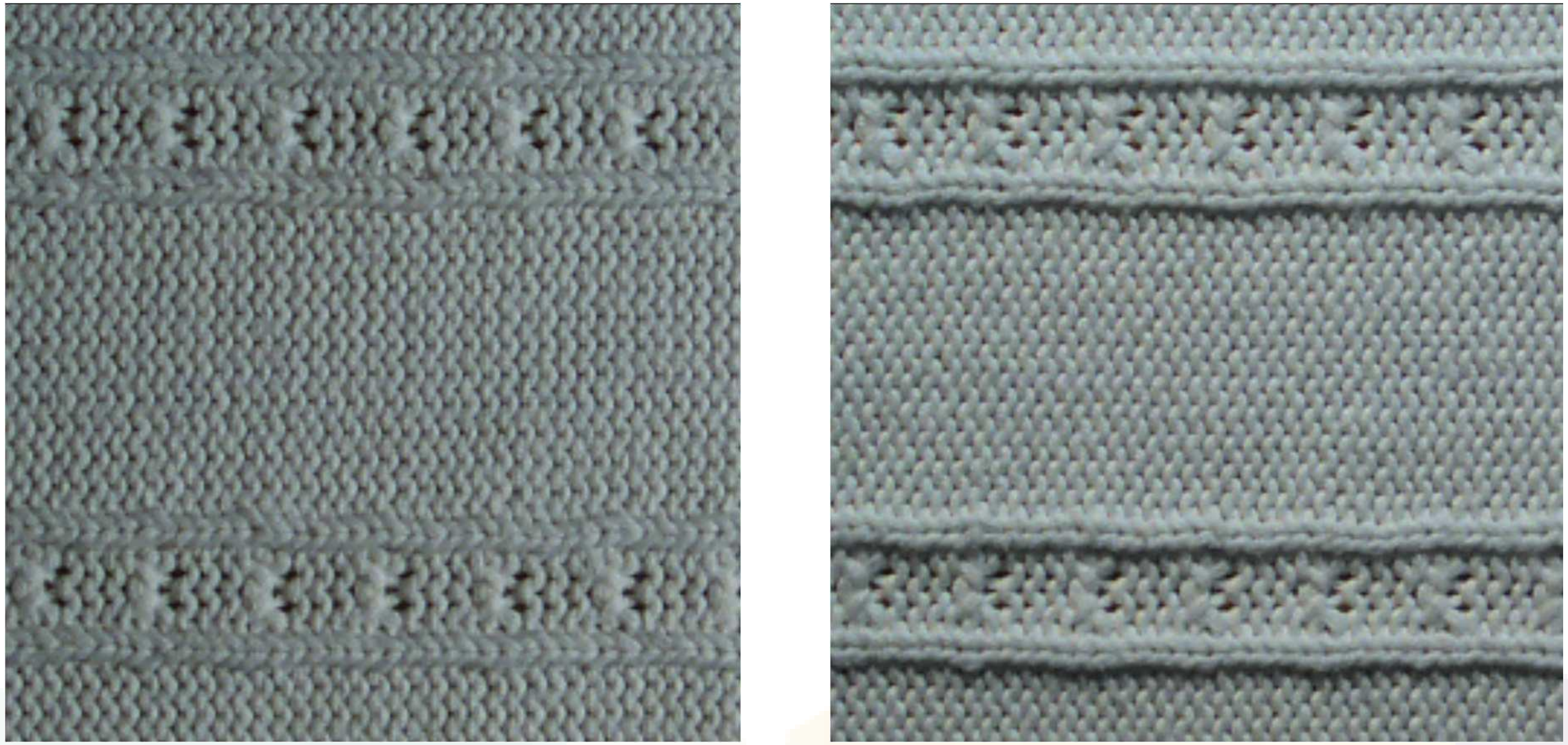


- "No formal definition of texture exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness and regularity." [Gonzalez, 2002]

- Usually refers to a scene taken from a single object/material characterized by spatial complexity



Surface texture

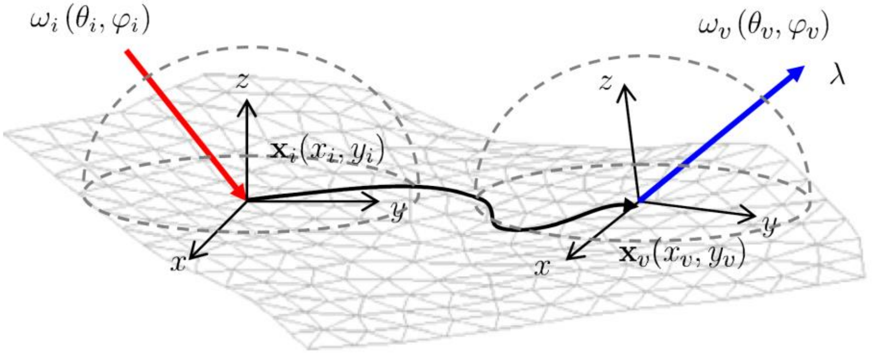


From [Dong, 2005]

© MICHELE CONNI

3. Texture models

- General Reflectance Function (GRF): 16 variables source, detector, collision, emission coordinates + time and frequency of generation and detection
- Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF): 9D (scattering)
- Bidirectional Texture Function (BTF): 7D (surface)
- Bidirectional Reflectance Distribution Function (BRDF): 5D (point)
- Drawbacks: lengthy and expensive processes, cumbersome data management

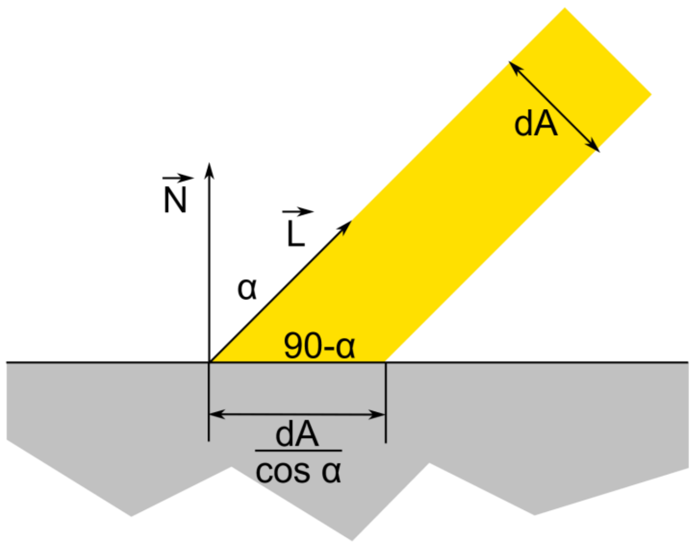


From [Haindl, 2013]

© MICHELE CONNI

Lambertian reflectance model

- Property of ideal diffusely reflecting surface
- Surface reflectance is isotropic
- It is impossible to tell where the incident light comes from
- No specular peak
- Real world examples: matte paper, flat paint, opal glass

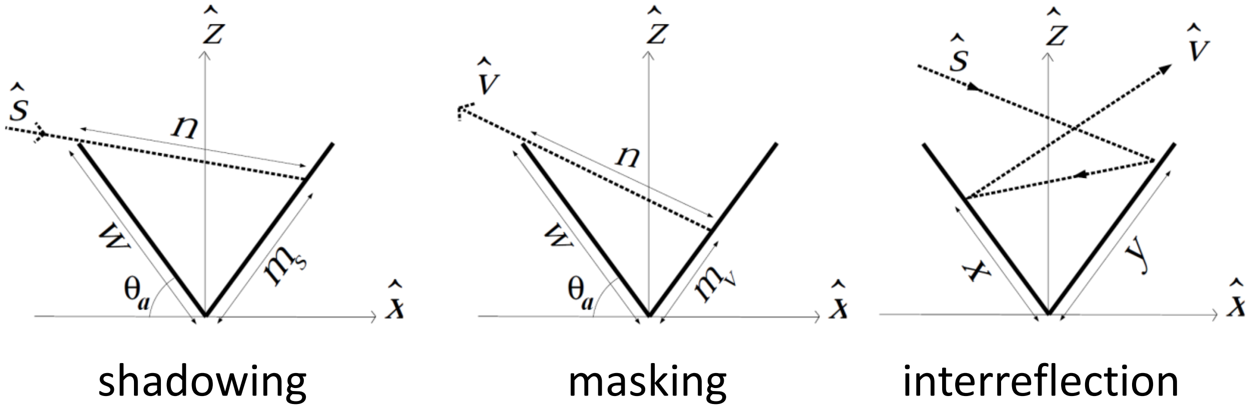


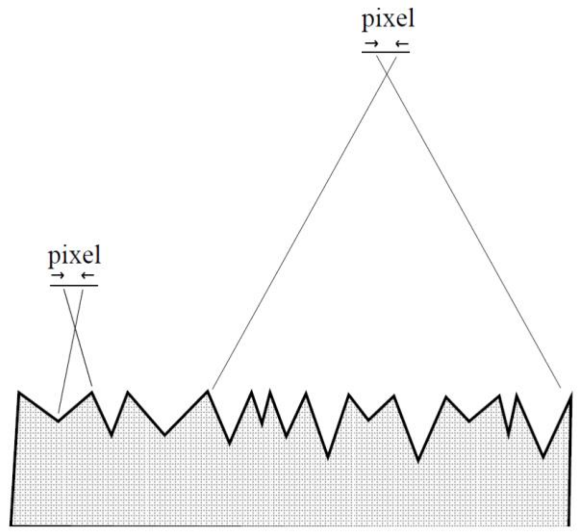
From [wikipedia]

© MICHELE CONNI

Oren–Nayar reflectance model

- Comprehensive model for body reflectance from surfaces with macroscopic roughness
- Accounts for complex geometric and radiometric phenomena (masking, shadowing, interreflections)
- Based on V-cavities
- Depends on the acquisition system (e.g. resolution of pixels)





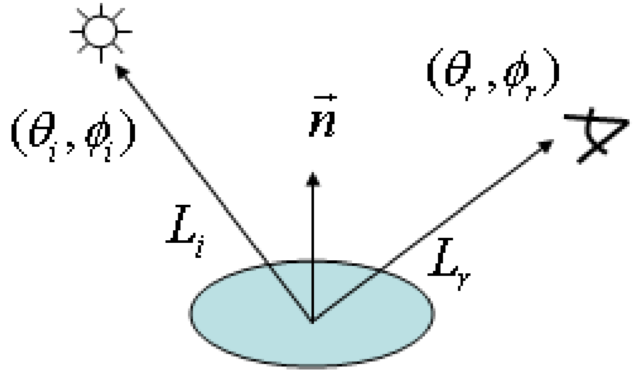
From [Oren, 1994]

© MICHELE CONNI

Oren–Nayar reflectance model

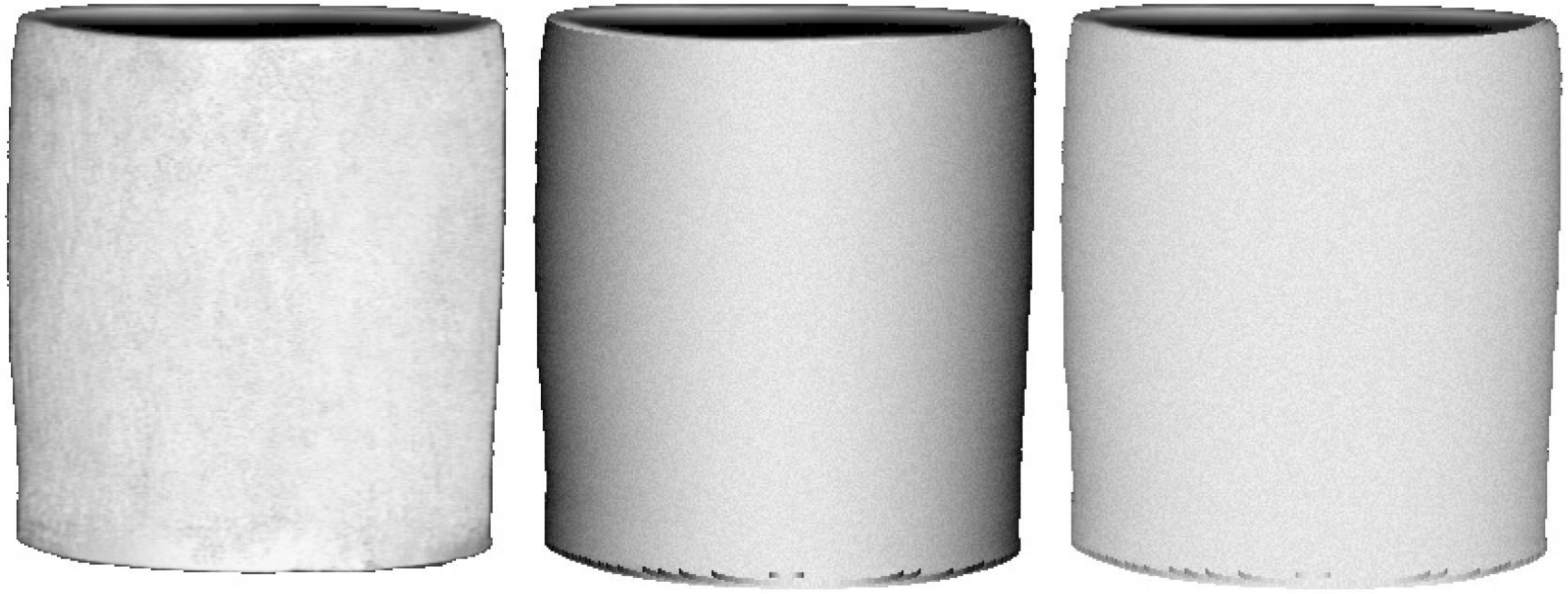
$$L_r = \frac{\rho}{\pi} \cos\theta_i [A(\sigma) + B(\sigma) \max(0, \cos(\phi_i - \phi_r)) \sin(\max(\theta_i, \theta_r)) \tan(\min(\theta_i, \theta_r))] E_0$$

- Statistical model
- Effective for rough diffuse surfaces, such as, plaster, sand, clay, and cloth
- $A(\sigma) = 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33}$
- $B(\sigma) = 0.45 \frac{\sigma^2}{\sigma^2 + 0.09}$



From [wikipedia]

Oren–Nayar reflectance model

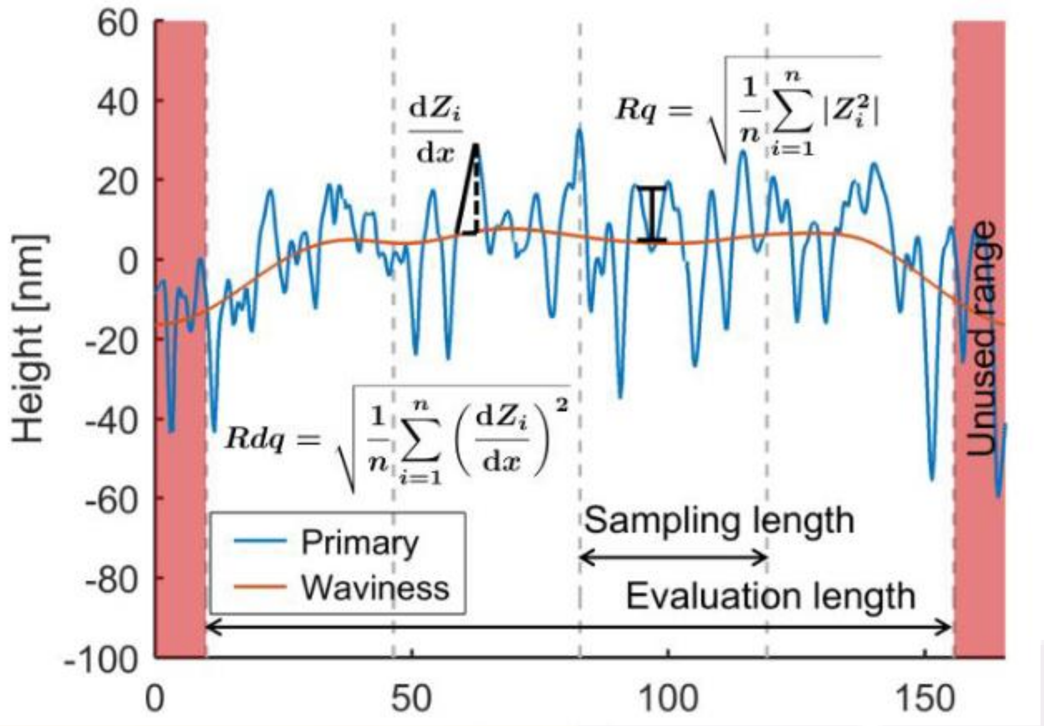


(a) Image (b) Lambertian (c) Model

From [Oren, 1994]

Roughness

- More models exist
- E.g., Principled BRDF
- $Rdq = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{dZ}{\mathrm{dx}}\right)_i^2}$
- Lambertian and Oren-Nayar models can be mixed

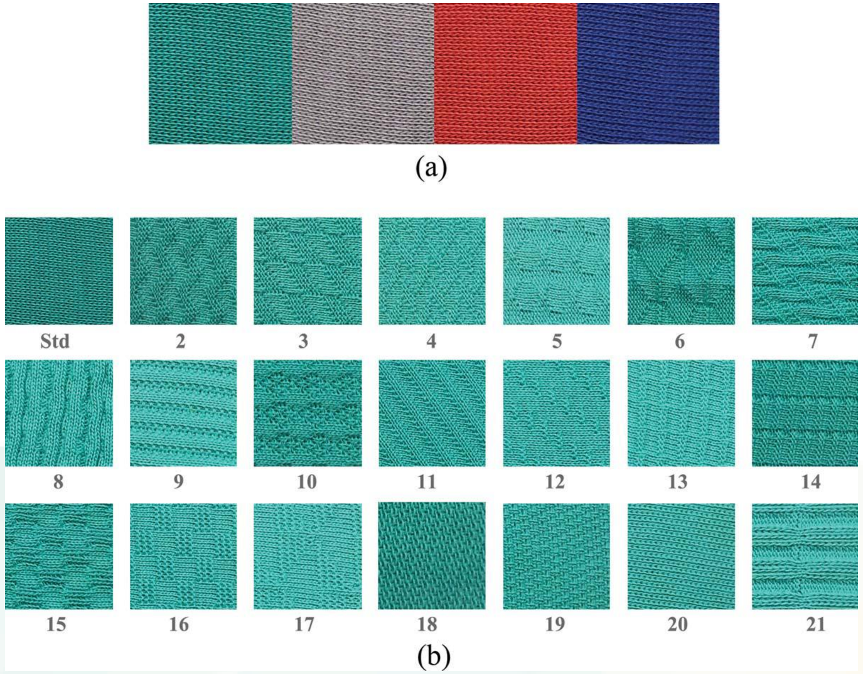


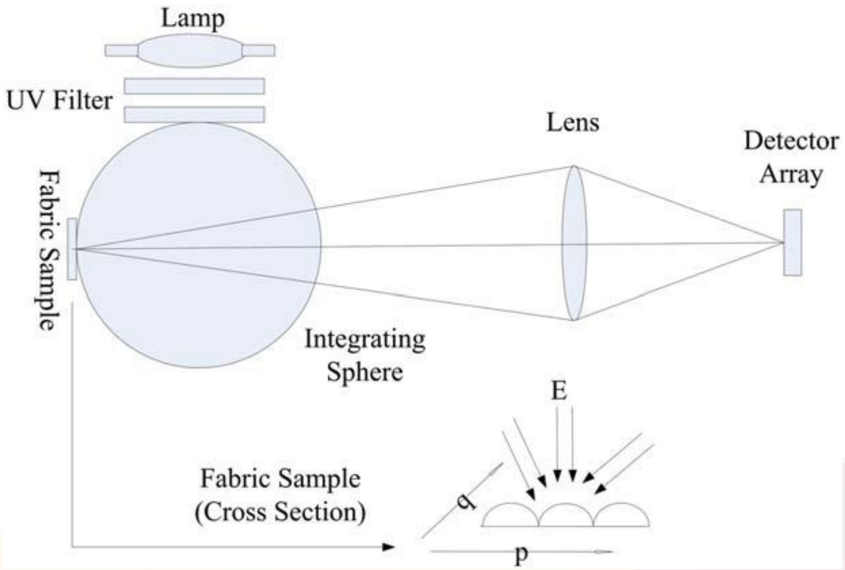
From [Feidenhans'l, 2015]

© MICHELE CONNI

Luo texture correction model

- Use reflectance model to correct colour measurements of textiles
- The correction must be done in CIEXYZ, given linearity with reflectance
- $X = \frac{\iint_{p,q} m_b(p,q)H(p,q)dpdq}{A_r} \int_\lambda E(\lambda)R(\lambda)\bar{x}(\lambda)d\lambda = C \cdot X_{norm}$



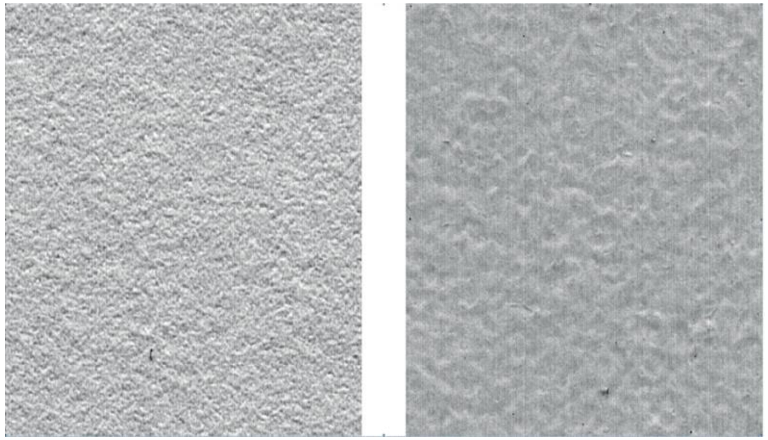


From [Luo, 2014]

© MICHELE CONNI

4. Roughness correction in iccMAX

- Extension of ICC v4
- v5 in header
- Backwards compatibility
- ISO 20677
- Novelties examples:
- Extended connection space (e.g. flexible PCS, illuminant, CMFs)
- multiProcessingElements (matrices, LUTs, CAM and Calc elements)
- Spectral and BRDF support
- Height/normal map can be stored but not used in profile
- Can be used for rendering



From [Specification ICC.2:2019]

© MICHELE CONNI

The Calc element

- MultiProcessElement
- Stack-based programming language
- Uses reverse polish notation: $2 * 4 \rightarrow 2\ 4\ mul$
- Structure of CalculatorElement:

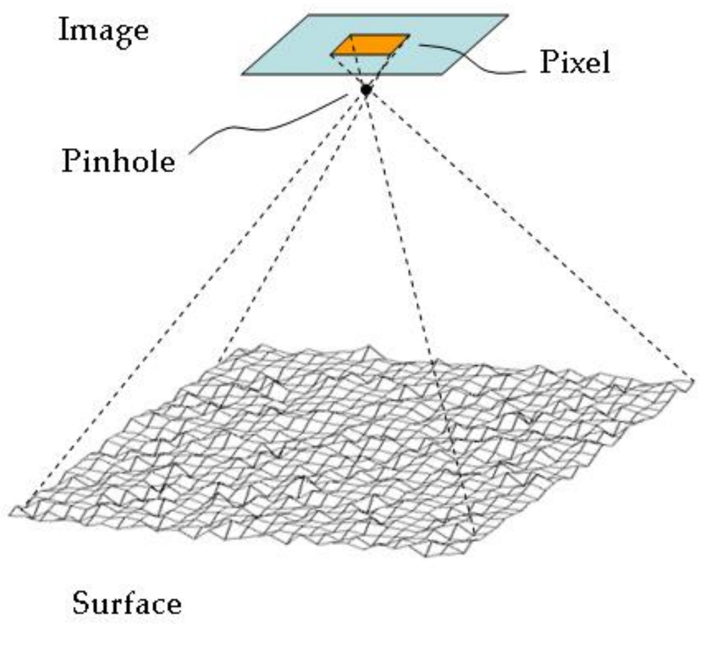
```
<CalculatorElement InputChannels="in" OutputChannels="out" InputNames="…" OutputNames="…">
        <Imports> … </Imports>
        <Variables> … </Variables>
        <Macros> … </Macros>
        <SubElements>… </SubElements>

        <MainFunction>Extended Textual Representation of Operations</MainFunction>
</CalculatorElement>
```

- The data is private to the MultiProcessElement
- Stack empty at the start

© MICHELE CONNI

Implementation

- Two texture correction models: Lambertian and Oren–Nayar
- Purpose: correction of colour measurement on complex surface
- $XYZ_{norm} = XYZ_{meas}/C$
- $C_{Lambert} = \mu(\cos(\theta_i))$
- $C_{Oren-Nayar} = \mu(\cos\theta_i[A(\sigma) + B(\sigma)\max(0, \cos(\phi_i - \phi_r))\sin(\max(\theta_i, \theta_r))\tan(\min(\theta_i, \theta_r))])$
- Single profile, CIEXYZ data colour space, CIEXYZ PCS
- Relative Colorimetric rendering intent
- ColorSpace ('spac') profile
- Correction implemented in multiProcessElement, A2B1 tag
- B2A1 tag implements $XYZ_{meas} = C \cdot XYZ_{norm}$



From [wikipedia]

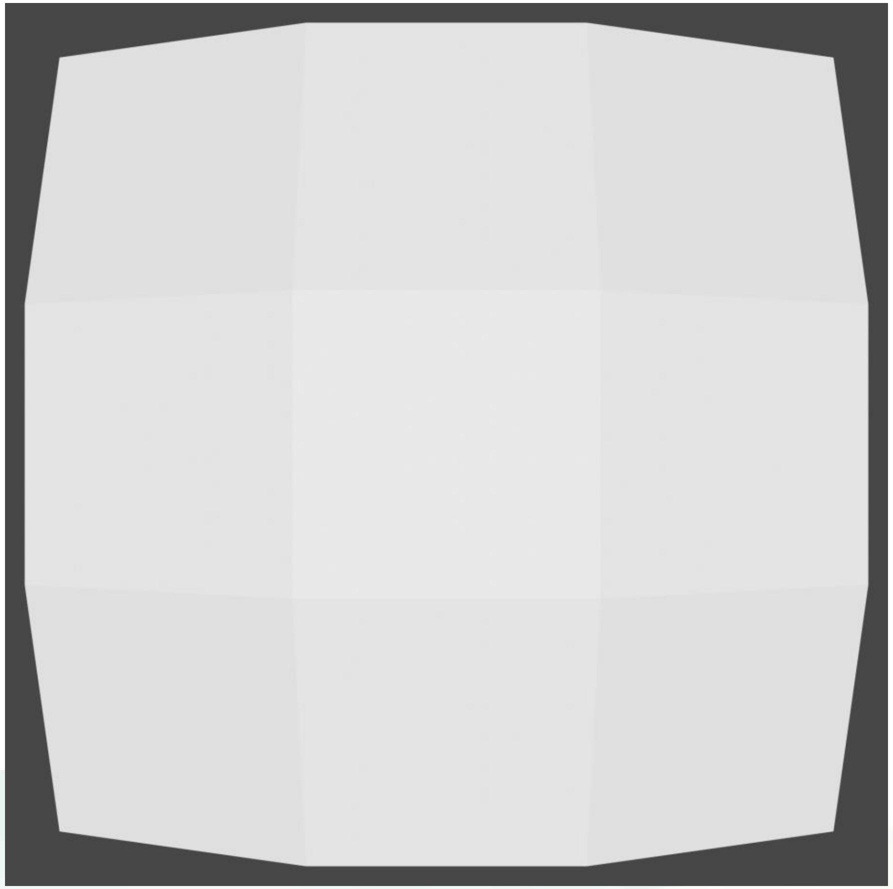
© MICHELE CONNI

Lambertian model correction

- $XYZ_{norm} = XYZ_{meas}/\mu(\cos(\theta_i))$
- Declarations:

```xml
<multiProcessElementType>
    <TagSignature>A2B1</TagSignature>
    <MultiProcessElements InputChannels="3" OutputChannels="3">
        <CalculatorElement InputChannels="3" OutputChannels="3">
            <Imports>
                <Import Filename="import_lamb.xml"/>
            </Imports>
            <Variables>
                <Declare Name="myVector" Size="9"/>
            </Variables>
            <Macros>
                <!-- Macro to convert degrees to radians -->
                <Macro Name="torad">3.14159265359 mul 180 div</Macro>
            </Macros>
        <MainFunction>
```

3D structure



Front

Side

© MICHELE CONNI

Lambertian model correction

- import_lamb.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<IccCalcImport>
    <Macros>
        <!-- Map of polar angles of incident light with each facelet -->
        <Macro Name="ti">
            44.14 44.14 44.14
            55.00 45.00 35.00
            44.14 44.14 44.14
        </Macro>
    </Macros>
</IccCalcImport>
```


© MICHELE CONNI

Lambertian model correction

- Main function:

```
        <MainFunction>
{
in(0,3)
% Load normal map
call{ti}
% Calculate cosine and save in myVector
call{torad} cos tput{myVector[0]}
call{torad} cos tput{myVector[1]}
call{torad} cos tput{myVector[2]}
call{torad} cos tput{myVector[3]}
call{torad} cos tput{myVector[4]}
call{torad} cos tput{myVector[5]}
call{torad} cos tput{myVector[6]}
call{torad} cos tput{myVector[7]}
call{torad} cos tput{myVector[8]}
% Calculate average cosine
tget{myVector} sum(9)
9 div
% Use it to correct XYZ
copy copy
div(3)
out(0,3)
}
        </MainFunction>
```

Oren-Nayar model correction

- $XYZ_{norm} = XYZ_{meas}/\mu\,(\cos\theta_i[A(\sigma) + B(\sigma)\max(0, \cos(\phi_i - \phi_r))\sin(\max(\theta_i, \theta_r))\tan(\min(\theta_i, \theta_r))])$
- Declarations:

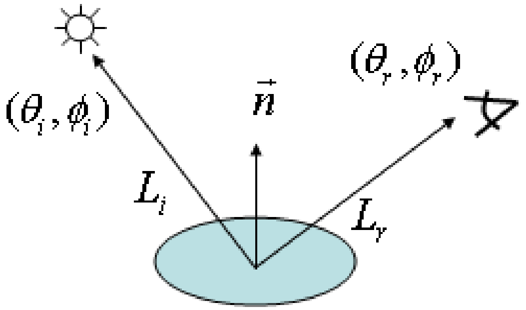
```
<multiProcessElementType>
    <TagSignature>A2B1</TagSignature>
    <MultiProcessElements InputChannels="3" OutputChannels="3">
        <CalculatorElement InputChannels="3" OutputChannels="3">
            <Imports>
                <Import Filename="import_oren.xml"/>
            </Imports>
            <Variables>
                <Declare Name="cos_ti" Size="9"/>
                <Declare Name="A"/>
                <Declare Name="B"/>
                <Declare Name="phi_i" Size="9"/>
                <Declare Name="phi_r" Size="9"/>
                <Declare Name="t_i" Size="9"/>
                <Declare Name="t_r" Size="9"/>
                <Declare Name="cos_diff" Size="9"/>
                <Declare Name="sin_alpha" Size="9"/>
                <Declare Name="tan_beta" Size="9"/>
            </Variables>
```

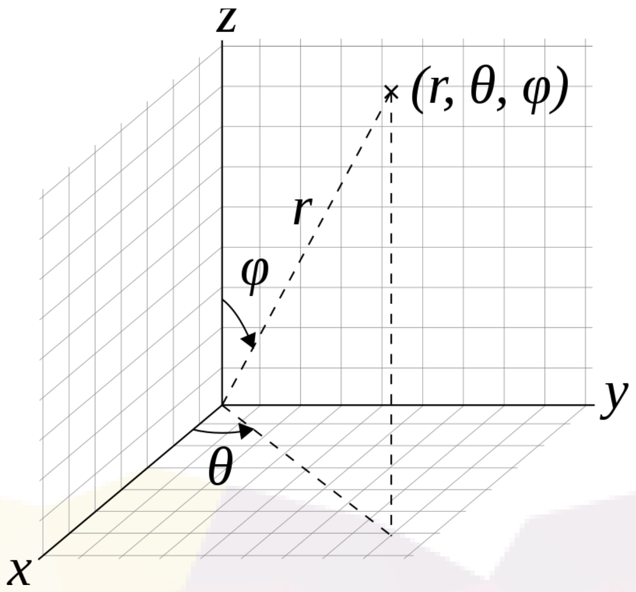

© MICHELE CONNI

Oren-Nayar model correction

- import_oren.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<IccCalcImport>
    <Macros>
        <!-- Here are stored the parameters of the Oren-Nayar correction model -->
        <!-- Std dev of the model -->
        <Macro Name="sigma">
            0.50
        </Macro>
        <!-- Map of polar angles of incident light with each facelet -->
        <Macro Name="ti">
            44.14 44.14 44.14
            55.00 45.00 35.00
            44.14 44.14 44.14
        </Macro>
        <!-- Map of azimuth angles of incident light with each facelet -->
        <Macro Name="phii">
            -125.15   9.85  54.85
               0.00   0.00   0.00
             144.85 350.15 -35.15
        </Macro>
        <!-- Map of polar angles of detector with each facelet -->
        <Macro Name="tr">
            80.00 10.00 80.00
            10.00  0.00 10.00
            80.00 10.00 80.00
        </Macro>
        <!-- Map of azimuth angles of detector with each facelet -->
        <Macro Name="phir">
             -45.00  90.00 135.00
               0.00   0.00 180.00
            -135.00 270.00  45.00
        </Macro>
    </Macros>
</IccCalcImport>
```





From [wikipedia]

Oren-Nayar model correction

- Conversion to radiants and $\cos(\theta_i)$:

```
<Macros>
    <!-- Macro to convert degrees to radians -->
    <Macro Name="torad">3.14159265359 mul 180 div</Macro>
    <Macro Name="calc_cos_ti">
        % Load illumination cosines map
        call{ti} tput{t_i}
        % Calculate cosine and save in cos_ti
        tget{t_i[0]} call{torad} cos tput{cos_ti[0]}
        tget{t_i[1]} call{torad} cos tput{cos_ti[1]}
        tget{t_i[2]} call{torad} cos tput{cos_ti[2]}
        tget{t_i[3]} call{torad} cos tput{cos_ti[3]}
        tget{t_i[4]} call{torad} cos tput{cos_ti[4]}
        tget{t_i[5]} call{torad} cos tput{cos_ti[5]}
        tget{t_i[6]} call{torad} cos tput{cos_ti[6]}
        tget{t_i[7]} call{torad} cos tput{cos_ti[7]}
        tget{t_i[8]} call{torad} cos tput{cos_ti[8]}
    </Macro>
```

Oren-Nayar model correction

- $A(\sigma)$ and $B(\sigma)$:

```
</Macro>
<Macro Name="calc_A">
    call{sigma} call{sigma} mul
    tsav(0,1)
    tget(0,1) 0.33 add
    div
    -0.5 mul
    1 add
    tput{A}
</Macro>
<Macro Name="calc_B">
    call{sigma} call{sigma} mul
    tsav(0,1)
    tget(0,1) 0.09 add
    div
    0.45 mul
    tput{B}
</Macro>
```

Oren-Nayar model correction

- $\max(0, \cos(\phi_i - \phi_r))$:

```
<Macro Name="calc_cos_diff">
    call{phii} tput{phi_i}
    call{phir} tput{phi_r}
    tget{phi_i[0]} tget{phi_r[0]} sub call{torad} cos 0 max tput{cos_diff[0]}
    tget{phi_i[1]} tget{phi_r[1]} sub call{torad} cos 0 max tput{cos_diff[1]}
    tget{phi_i[2]} tget{phi_r[2]} sub call{torad} cos 0 max tput{cos_diff[2]}
    tget{phi_i[3]} tget{phi_r[3]} sub call{torad} cos 0 max tput{cos_diff[3]}
    tget{phi_i[4]} tget{phi_r[4]} sub call{torad} cos 0 max tput{cos_diff[4]}
    tget{phi_i[5]} tget{phi_r[5]} sub call{torad} cos 0 max tput{cos_diff[5]}
    tget{phi_i[6]} tget{phi_r[6]} sub call{torad} cos 0 max tput{cos_diff[6]}
    tget{phi_i[7]} tget{phi_r[7]} sub call{torad} cos 0 max tput{cos_diff[7]}
    tget{phi_i[8]} tget{phi_r[8]} sub call{torad} cos 0 max tput{cos_diff[8]}
</Macro>
```


© MICHELE CONNI

Oren-Nayar model correction

- $\sin(\max(\theta_i, \theta_r))$:

```
<Macro Name="calc_sin_alpha">
    call{tr} tput{t_r}
    tget{t_i[0]} tget{t_r[0]} max call{torad} sin tput{sin_alpha[0]}
    tget{t_i[1]} tget{t_r[1]} max call{torad} sin tput{sin_alpha[1]}
    tget{t_i[2]} tget{t_r[2]} max call{torad} sin tput{sin_alpha[2]}
    tget{t_i[3]} tget{t_r[3]} max call{torad} sin tput{sin_alpha[3]}
    tget{t_i[4]} tget{t_r[4]} max call{torad} sin tput{sin_alpha[4]}
    tget{t_i[5]} tget{t_r[5]} max call{torad} sin tput{sin_alpha[5]}
    tget{t_i[6]} tget{t_r[6]} max call{torad} sin tput{sin_alpha[6]}
    tget{t_i[7]} tget{t_r[7]} max call{torad} sin tput{sin_alpha[7]}
    tget{t_i[8]} tget{t_r[8]} max call{torad} sin tput{sin_alpha[8]}
</Macro>
```


© MICHELE CONNI

Oren-Nayar model correction

- $\tan(\min(\theta_i, \theta_r))$:

```
<Macro Name="calc_tan_beta">
    tget{t_i[0]} tget{t_r[0]} min call{torad} tan tput{tan_beta[0]}
    tget{t_i[1]} tget{t_r[1]} min call{torad} tan tput{tan_beta[1]}
    tget{t_i[2]} tget{t_r[2]} min call{torad} tan tput{tan_beta[2]}
    tget{t_i[3]} tget{t_r[3]} min call{torad} tan tput{tan_beta[3]}
    tget{t_i[4]} tget{t_r[4]} min call{torad} tan tput{tan_beta[4]}
    tget{t_i[5]} tget{t_r[5]} min call{torad} tan tput{tan_beta[5]}
    tget{t_i[6]} tget{t_r[6]} min call{torad} tan tput{tan_beta[6]}
    tget{t_i[7]} tget{t_r[7]} min call{torad} tan tput{tan_beta[7]}
    tget{t_i[8]} tget{t_r[8]} min call{torad} tan tput{tan_beta[8]}
</Macro>
```


© MICHELE CONNI

Oren-Nayar model correction

- Main function:

```
        <MainFunction>
{
in(0,3)
% Calculate cosine of illumination and save in cos_ti
call{calc_cos_ti}
% Calculate coefficient A
call{calc_A}
% Calculate coefficient B
call{calc_B}
% Calculate cosine of difference
call{calc_cos_diff}
% Calculate sin(alpha)
call{calc_sin_alpha}
% Calculate tan(beta)
call{calc_tan_beta}
% Put everything together to get the correction coefficient
tget{B} tget{cos_diff[0]} mul tget{sin_alpha[0]} mul tget{tan_beta[0]} mul tget{A} sum tget{cos_ti[0]} mul
tget{B} tget{cos_diff[1]} mul tget{sin_alpha[1]} mul tget{tan_beta[1]} mul tget{A} sum tget{cos_ti[1]} mul
tget{B} tget{cos_diff[2]} mul tget{sin_alpha[2]} mul tget{tan_beta[2]} mul tget{A} sum tget{cos_ti[2]} mul
tget{B} tget{cos_diff[3]} mul tget{sin_alpha[3]} mul tget{tan_beta[3]} mul tget{A} sum tget{cos_ti[3]} mul
tget{B} tget{cos_diff[4]} mul tget{sin_alpha[4]} mul tget{tan_beta[4]} mul tget{A} sum tget{cos_ti[4]} mul
tget{B} tget{cos_diff[5]} mul tget{sin_alpha[5]} mul tget{tan_beta[5]} mul tget{A} sum tget{cos_ti[5]} mul
tget{B} tget{cos_diff[6]} mul tget{sin_alpha[6]} mul tget{tan_beta[6]} mul tget{A} sum tget{cos_ti[6]} mul
tget{B} tget{cos_diff[7]} mul tget{sin_alpha[7]} mul tget{tan_beta[7]} mul tget{A} sum tget{cos_ti[7]} mul
tget{B} tget{cos_diff[8]} mul tget{sin_alpha[8]} mul tget{tan_beta[8]} mul tget{A} sum tget{cos_ti[8]} mul
% Calculate average correction factor
sum(9) 9 div
% Use it to correct XYZ
copy copy
div(3)
out(0,3)
}
      </MainFunction>
```


© MICHELE CONNI

Input

- The .xml file can be converted in .icc profile with iccFromXml.exe
- Applied to image with iccApplyProfiles.exe, to named input with iccApplyNamedCMM.exe
- We used:

  iccApplyNamedCMM.exe input_oren.txt 3 0 oren_nayar_correction.icc 3

- Final encoding: icEncodeFloat
- Interpolation: Linear
- Rendering intent: Absolute
- Input file:

```
'XYZ'     ; Data Format
icEncodeFloat    ; Encoding

0.5 0.5 0.5
```

Results

- Output:

```
'XYZ '   ; Data Format
icEncodeFloat    ; Encoding

;Source Data Format: 'dXYZ'
;Source Data Encoding: icEncodeFloat
;Source data is after semicolon

;Profiles applied
; oren_nayar_correction.icc

   0.8689     0.8689     0.8689   ;    0.5000     0.5000     0.5000
```


© MICHELE CONNI

Thank you for your attention

Michele Conni
(R&D Engineer at Barbieri Electronic
Ph.D. candidate at NTNU)

michele.conni@barbierielectronic.com
michelco@stud.ntnu.no
www.barbierielectronic.com

© MICHELE CONNI

QUESTIONS?

© MICHELE CONNI

References

- http://www.cs.columbia.edu/CAVE/projects/btf/
- Dong, J., & Chantler, M. (2005). Capture and synthesis of 3D surface texture. International Journal of Computer Vision, 62(1-2), 177-194.
- Feidenhans, N. A., Hansen, P. E., Pilný, L., Madsen, M. H., Bissacco, G., Petersen, J. C., & Taboryski, R. (2015, October). Industrial characterization of nano-scale roughness on polished surfaces. In Optifab 2015 (Vol. 9633, p. 96330B). International Society for Optics and Photonics.
- Gonzalez, R. C., & Woods, R. E. (2002). Digital image processing [M]. Publishing house of electronics industry, 141(7).
- Haindl, M., & Filip, J. (2013). Visual texture: Accurate material appearance measurement, representation and modeling. Springer Science & Business Media.
- Luo, L., Tsang, K. M., Shen, H. L., Shao, S. J., & Xin, J. H. (2015). An investigation of how the texture surface of a fabric influences its instrumental color. Color Research & Application, 40(5), 472-482.

 © MICHELE CONNI

- Oren, M., & Nayar, S. K. (1994). Generalization of Lambert's reflectance model. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques (pp. 239-246).
- Specification ICC.2:2019 (Profile version 5.0.0 - iccMAX), Image technology colour management — Extensions to architecture, profile format and data structure
- https://en.wikipedia.org/wiki/Lambertian_reflectance

 © MICHELE CONNI