INTERNATIONAL STANDARD

ISO/IEC 15938-3

First edition
2002-05-15

AMENDMENT 3
2009-04-01

Information technology — Multimedia content description interface —

Part 3: **Visual**

AMENDMENT 3: Image signature tools

Technologies de l'information — Interface de description du contenu multimédia —

Partie 3: Visuel

AMENDEMENT 3: Outils de signature d'image

Reference number
ISO/IEC 15938-3:2002/Amd.3:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 3 to ISO/IEC 15938-3:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Multimedia content description interface —

Part 3: Visual

AMENDMENT 3: Image signature tools

Page 1, Scope

Replace 1.2 entirely with the following:

1.2 Overview of Visual Description Tools

This part of ISO/IEC 15938 specifies tools for description of visual content, including still images, video and 3D models. These tools are defined by their syntax in DDL and binary representations and semantics associated with the syntactic elements. They enable description of the visual features of the visual material, such as color, texture, shape, motion, localization of the described objects in the image or video sequence and also unique and robust identification of visual material. An overview of the visual description tools is shown in Figure 1.

The basic structure description tools include five supporting tools of visual descriptions defined in clauses 6-11. They are categorized into two groups, descriptor containers and basic supporting tools. The former consists of three datatypes, GridLayout providing efficient representations of visual features on grids, TimeSeries representing temporal arrays of several descriptions, GofGopFeature describes representative descriptions over video segment, and MultipleView describing a 3D object using several pictures captured from different view angles. The latter contains two tools, Spatial2DcoordinateSystem used to specify the 2D coordinate system and TemporalInterpolation indicating the interpolation method between two samples on a time axis.

The remaining description tools, except for the FaceRecognition and ImageSignature descriptors, are associated with visual features and are grouped into five feature categories: Color, Texture, Shape, Motion and Localization.

The color description tools include five color descriptors to represent different aspects of color features: representative colors (DominantColor), color distribution (ScalableColor), spatial distribution of colors (ColorLayout and ColorStructure) and perceptual feeling of illumination color (ColorTemperature). It also contains three supporting tools, ColorSpace and ColorQuantization used in DominantColor and IlluminationInvariantColor to extend four color descriptors, DominantColor, ScalableColor, ColorLayout and ColorStructure, to support illumination invariant similarity matching. An extension of ScalableColor to a group of frames or pictures (GoFGoPColor) is also included in this group. All the color descriptors can be extracted from arbitrarily shaped regions.

The texture description tools facilitate browsing (TextureBrowsing) and similarity retrieval (HomogeneousTexture and EdgeHistogram) using the texture of a still or moving image region. All the texture descriptors can be extracted from arbitrarily shaped regions.

The shape description tools include two descriptors that characterize different shape features of a 2D object or region. The RegionShape descriptor captures the distribution of all pixels within a region and the Contour

Shape descriptor characterizes the shape properties of the contour of an object. The extension of RegionShape is also defined as ShapeVariation to describe temporal variation of shape over video segment. The Shape3D and Perceptual 3D Shape descriptors provide 3-dimensional shape information; the former represents an intrinsic shape characterization of 3D mesh models, and the latter represents part-based representation of a 3D object.

The motion description tools include four descriptors that characterize various aspects of motion. The CameraMotion descriptor specifies a set of basic camera operations such as, for example, panning and tilting. The motion of a key point (pixel) from a moving object or region can be characterized by the MotionTrajectory descriptor. The ParametricMotion descriptor characterizes an evolution of an arbitrarily shaped region over time in terms of a 2D geometric transformation. Finally, the MotionActivity descriptor captures the pace of the motion in the sequence, as perceived by the viewer. All motion descriptors except for CameraMotion can be extracted from arbitrarily shaped regions.

The localization description tools can be used to indicate regions of interest in the spatial (RegionLocator) and spatio-temporal (SpatioTemporalLocator) domains.

The FaceRecognition descriptor and the Advanced Face Recognition descriptor are not associated with any particular visual feature and can be used to describe a human face for applications requiring the matching and retrieval of face images.

The ImageSignature descriptor provides a "fingerprint" of an image that uniquely identify it. The signature is robust (unchanging) across a wide range of common editing operations, but is sufficiently different for every item of "original" content to identify it uniquely and reliably – just like human fingerprints. The ImageSignature has no direct association with specific visual features such as colour, shape or texture.

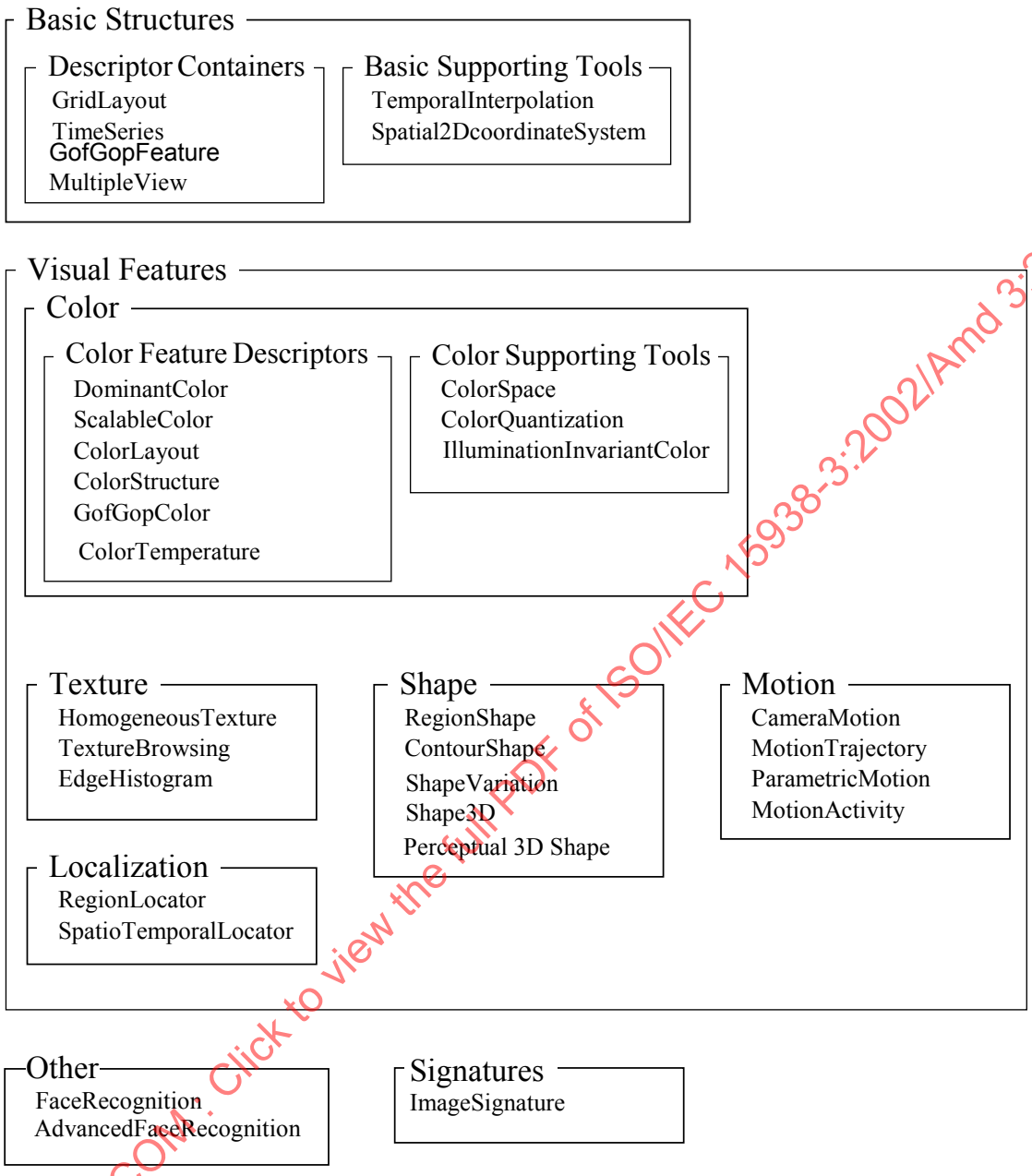

Figure 1 — Overview of Visual Description Tools

Replace 3.2 with:

3.2 Abbreviations

ART	Angular-Radial Transform
CSS	Curvature Scale Space
DDL	Description Definition Language
DS	Description Scheme
D	Descriptor
DCT	Discrete Cosine Transform
FOC	Focus of Contraction
FOE	Focus of Expansion
FFT	Fast Fourier Transform
GoF	Group of Frames
GoP	Group of Pictures
HMMD	Hue-Min-Max-Difference
HSV	Hue-Saturation-Value
RGB	Red-Green-Blue

Replace 4.2.2 with:

4.2.2 Generic binary representation

The use of the video-specific syntax is signalled using the codec configuration mechanism defined in ISO/IEC 15938-1. The following classification scheme is defined for this purpose.

```
<ClassificationScheme uri="urn:mpeg:mpeg7:cs:VisualDescriptorCodecCS:2001">
   <Term termID="1">
      <Name xml:lang="en">MPEG7CameraMotion</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Camera Motion
            Codec</Definition>
   </Term>
   <Term termID="2">
      <Name xml:lang="en">MPEG7ColorLayout</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Layout
            Codec</Definition>
   </Term>
   <Term termID="3">
      <Name xml:lang="en">MPEG7ColorQuantization</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Quantization
            Codec</Definition>
   </Term>
   <Term termID="4">
      <Name xml:lang="en">MPEG7ColorSpace</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Space
            Codec</Definition>
   </Term>
   <Term termID="5">
      <Name xml:lang="en">MPEG7ColorStructure</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Structure
            Codec</Definition>
   </Term>
   <Term termID="6">
      <Name xml:lang="en">MPEG7ContourShape</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Contour Shape
            Codec</Definition>
```

```
    </Term>
    <Term termID="7">
        <Name xml:lang="en">MPEG7DominantColor</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Dominant Color
            Codec</Definition>
    </Term>
    <Term termID="8">
        <Name xml:lang="en">MPEG7EdgeHistogram</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Edge Histogram
            Codec</Definition>
    </Term>
    <Term termID="9">
        <Name xml:lang="en">MPEG7FaceRecognition</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Face Recognition
            Codec</Definition>
    </Term>
    <Term termID="10">
        <Name xml:lang="en">MPEG7FoFGoPColor</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary GoFGoP Color
            Codec</Definition>
    </Term>
    <Term termID="11">
        <Name xml:lang="en">MPEG7GridLayout</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Grid Layout
            Codec</Definition>
    </Term>
    <Term termID="12">
        <Name xml:lang="en">MPEG7HomogeneousTexture</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Homogeneous Texture
            Codec</Definition>
    </Term>
    <Term termID="13">
        <Name xml:lang="en">MPEG7IrregularVisualTimeSeries</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Irregular Time Series
            Codec</Definition>
    </Term>
    <Term termID="14">
        <Name xml:lang="en">MPEG7MotionActivity</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Motion Activity
            Codec</Definition>
    </Term>
    <Term termID="15">
        <Name xml:lang="en">MPEG7MotionTrajectory</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Motion Trajectory
            Codec</Definition>
    </Term>
    <Term termID="16">
        <Name xml:lang="en">MPEG7MultipleView</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Multiple View
            Codec</Definition>
    </Term>
    <Term termID="17">
        <Name xml:lang="en">MPEG7ParametricMotion</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Parametric Motion
            Codec</Definition>
    </Term>
    <Term termID="18">
        <Name xml:lang="en">MPEG7RegionLocator</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Region Locator
            Codec</Definition>
    </Term>
```

```
    <Term termID="19">
       <Name xml:lang="en">MPEG7RegionShape</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Region Shape
              Codec</Definition>
    </Term>
    <Term termID="20">
       <Name xml:lang="en">MPEG7RegularVisualTimeSeries</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Regular Time Series
              Codec</Definition>
    </Term>
    <Term termID="21">
       <Name xml:lang="en">MPEG7ScalableColor</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Scalable Color
              Codec</Definition>
    </Term>
    <Term termID="22">
       <Name xml:lang="en">MPEG7Shape3D</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Shape 3D
              Codec</Definition>
    </Term>
    <Term termID="23">
       <Name xml:lang="en">MPEG7Spatial2DCoordinateSystem</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Spatial 2D Coordinate
              System Codec</Definition>
    </Term>
    <Term termID="24">
       <Name xml:lang="en">MPEG7SpatioTemporalLocator</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary SpatioTemporal Locator
              Codec</Definition>
    </Term>
    <Term termID="25">
       <Name xml:lang="en">MPEG7TemporalInterpolation</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Temporal Interpolation
              Codec</Definition>
    </Term>
    <Term termID="26">
       <Name xml:lang="en">MPEG7TextureBrowsing</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Texture Browsing
              Codec</Definition>
    </Term>
    <Term termID="27">
       <Name xml:lang="en">MPEG7GofGopFeature</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Gof Gop Feature
Codec</Definition>
    </Term>
    <Term termID="28">
       <Name xml:lang="en">MPEG7ColorTemperature</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Temperature
Codec</Definition>
    </Term>
    <Term termID="29">
       <Name xml:lang="en">MPEG7ShapeVariation</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Shape Variation
Codec</Definition>
    </Term>
    <Term termID="30">
       <Name xml:lang="en">MPEG7IlluminationInvariantColor</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Illumination Invariant
Color Codec</Definition>
    </Term>
```

```
    <Term termID="31">
        <Name xml:lang="en">MPEG7AdvancedFaceRecognition</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Advanced Face Recognition
Codec</Definition>
    </Term>
    <Term termID="32">
        <Name xml:lang="en">MPEG7Perceptual3DShape</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Perceptual 3D Shape
Codec</Definition>
    </Term>
    <Term termID="33">
        <Name xml:lang="en">MPEG7ImageSignature</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Image Signature
Codec</Definition>
    </Term>
</ClassificationScheme>
```

In 5.2.4, replace Table 1 with:

Table 1 — Assignment of IDs to descriptors

ID	Descriptor
0	Forbidden
1	CameraMotion
2	ColorLayout
3	ColorSpace
4	ColorStructure
5	ColorQuantization
6	ContourShape
7	DominantColor
8	EdgeHistogram
9	FaceRecognition
10	GoFGoPColor
11	GridLayout
12	HomogeneousTexture
13	IrregularVisualTimeSeries
14	MotionActivity
15	MotionTrajectory
16	MultipleView
17	ParametricMotion
18	RegionLocator
19	RegionShape
20	RegularVisualTimeSeries
21	ScalableColor
22	Shape3D
23	Spatial2DCoordinateSystem
24	SpatioTemporalLocator
25	TemporalInterpolation
26	TextureBrowsing
27	GofGopFeature
28	ColorTemperature
29	ShapeVariation
30	IlluminationInvariantColor
31	AdvancedFaceRecognition
32	Perceptual3DShape
33	ImageSignature
34-255	Reserved

After 11.2, add the following:

1.3 Image Signature

11.3.1 Introduction

Most visual descriptors are very useful when trying to find images with *similar* content however, such descriptions are intended to be general: finding *identical* content is not of special importance. The image signature descriptor is robust (unchanging) across a wide range of common editing operations, but is sufficiently different for every item of "original" content to identify it uniquely and reliably – just like human fingerprints. Therefore this descriptor is able to distinguish between two images which have similar content and two images which are the same.

There are three components within the image signature, the first two are global signatures of the complete image and the third is composed of a set of local image signatures.

11.3.2 DDL representation syntax

```
<complexType name=”ImageSignatureType” final=”#all”>
  <complexContent>
    <extension base=”mpeg7:VisualDType”>
      <sequence>
        <element name="GlobalSignatureA">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned1" />
              </simpleType>
              <length value="512" />
            </restriction>
          </simpleType>
        </element>
      </sequence>
      <sequence>
        <element name="GlobalSignatureB">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned1" />
              </simpleType>
              <length value="512" />
            </restriction>
          </simpleType>
        </element>
        <element name="LocalSignature">
          <complexType>
            <sequence>
              <element name="FeaturePointCount">
                <simpleType>
                  <restriction base="nonNegativeInteger">
                    <minInclusive value="32" />
                    <maxInclusive value="80" />
                  </restriction>
                </simpleType>
              </element>
              <element name="FeaturePoint" minOccurs="32" maxOccurs="80">
                <complexType>
                  <sequence>
```

```
                    <element name="XCoord" type="mpeg7:unsigned8"/>
                    <element name="YCoord" type="mpeg7:unsigned8"/>
                    <element name="Direction" type="mpeg7:unsigned4"/>
                    <element name="LocalSignature">
                      <simpleType>
                        <restriction>
                          <simpleType>
                            <list itemType="mpeg7:unsigned1" />
                          </simpleType>
                          <length value="60" />
                        </restriction>
                      </simpleType>
                    </element>
                  </sequence>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.3.3 Binary Representation Syntax

ImageSignature {	Number of bits	Mnemonics
GlobalSignatureA	512	bslbf
GlobalSignatureB	512	bslbf
FeaturePointCount	8	uimsbf
for(k=0; k<NumberOfPoints; k++) {		
Xcoord	8	uimsbf
Ycoord	8	uimsbf
Direction	4	uimsbf
LocalSignature	60	bslbf
}		
}		

NumberOfPoints = FeaturePointCount + 32

The value of FeaturePointCount is restricted to the range 0-48

11.3.4 Descriptor Component Semantics

GlobalSignatureA
This element represents the first global signature of the image normalised with the original aspect ratio maintained.

GlobalSignatureB
This element represents the second global signature of the image normalised by changing the aspect ratio to 1:1.

FeaturePointCount
This field specifies the number of feature points represented in the descriptor.

Xcoord
The normalized horizontal coordinate of the feature point, the coordinates are normalized to be in the range 0-255.

Ycoord
The normalized vertical coordinate of the feature point, the coordinates are normalized to be in the range 0-255.

Direction
The direction of the image intensity gradient at the feature point expressed as an angle between 0 and 360 degrees, quantized to 0-15. It is defined as shown in the following pseudo-code:

```
int quantize_angle(float f_angle) {
    int direction;
    /* quantize angle using uniform 4 bit quantization
    over 0-360 degrees */
    if((f_angle>=0)&&(f_angle<22.5)) direction=0;
    else if((f_angle>=22.5)&&(f_angle<45.0)) direction=1;
    else if((f_angle>=45.0)&&(f_angle<67.5)) direction=2;
    else if((f_angle>=67.5)&&(f_angle<90.0)) direction=3;
    else if((f_angle>=90.0)&&(f_angle<112.5)) direction=4;
    else if((f_angle>=112.5)&&(f_angle<135.0)) direction=5;
    else if((f_angle>=135.0)&&(f_angle<157.5)) direction=6;
    else if((f_angle>=157.5)&&(f_angle<180.0)) direction=7;
    else if((f_angle>=180.0)&&(f_angle<202.5)) direction=8;
    else if((f_angle>=202.5)&&(f_angle<225.0)) direction=9;
    else if((f_angle>=225.0)&&(f_angle<247.5)) direction=10;
    else if((f_angle>=247.5)&&(f_angle<270.0)) direction=11;
    else if((f_angle>=270.0)&&(f_angle<292.5)) direction=12;
    else if((f_angle>=292.5)&&(f_angle<315.0)) direction=13;
    else if((f_angle>=315.0)&&(f_angle<337.5)) direction=14;
    else if((f_angle>=337.5)&&(f_angle<360.0)) direction=15;

    return direction;
}
```

LocalSignature
This element represents the local signature of the content around the feature.

11.3.5 Global Signature Extraction

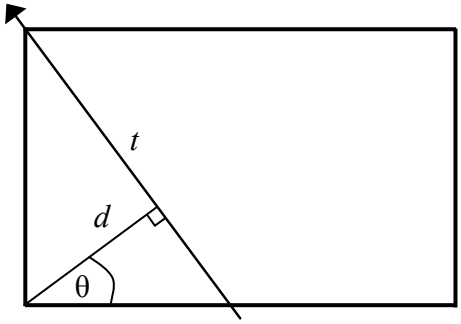

Figure 36 — Lines in the trace transform are parameterised by the angle θ and the distance *d*

The trace transform applies functionals over all of the lines in the image. A functional is a real-valued function on a vector space V, usually of functions. As shown in Figure 36, a line is parameterised by the distance d and the angle θ and points on the line by the parameter t. The result of applying a trace transform is a 2D function. A further functional can then be applied to the columns of the trace transform to give a vector of real numbers. This second functional is known as the diametrical functional and the resulting vector is known as the circus function. The properties of the circus function can be controlled by appropriate choices of the two different functionals (trace and diametrical).

11.3.5.1 Multi-Resolution Representation

A multi-resolution trace transform shall be created by sub-sampling an original trace transform in its *d* dimension. In the "trace-domain" sub-sampling of the *d*-parameter shall be performed by integrating over intervals along the columns, as in Figure 38. This corresponds to projecting strips of width Δd over the image during the trace transform, as shown in Figure 39.

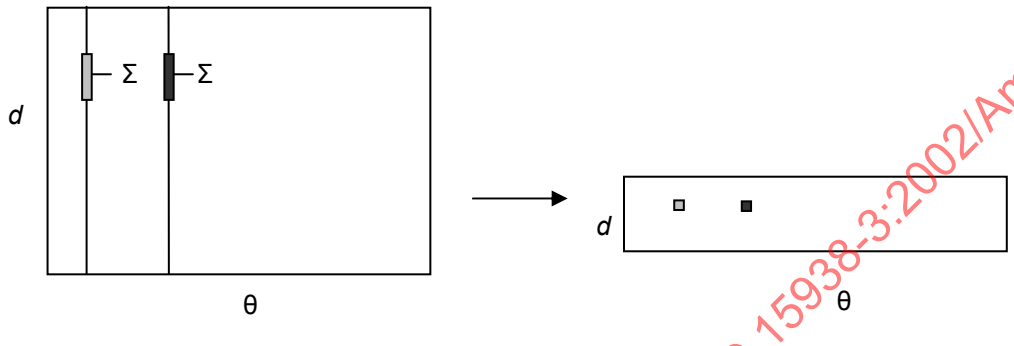

Figure 37 — The trace transform is decomposed by summing over bands in the *d*-parameter

Using this approach a pyramid of trace transforms shall be constructed. Multiple binary strings shall be obtained from one trace transform by using a multi-resolution decomposition, where the sub-sampling takes place over a range of different interval widths.

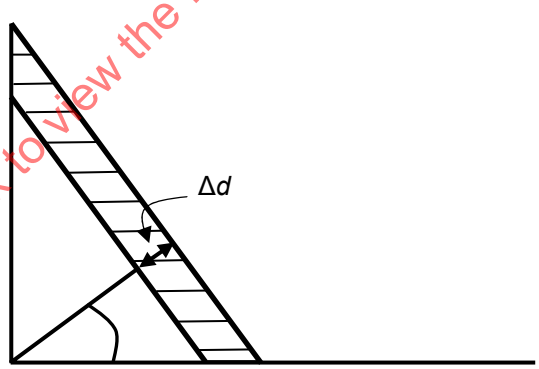

Figure 38 — Decomposing the trace transform in the d-parameter corresponds to projecting strips over the original image

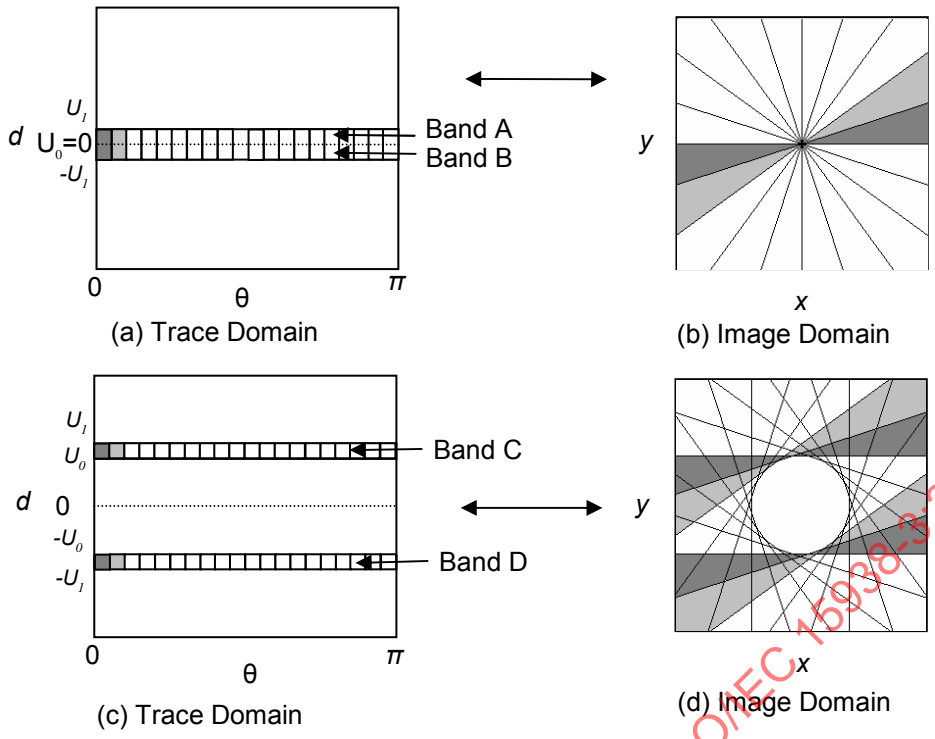

Figure 39 — Bands in the trace domain are equivalent to double-cones in the image domain

11.3.5.2 Trace-band Representation

The trace transform of an image contains a representation of every possible line across the image. Therefore the circus function and the basic identifiers described in 11.3.5 also contain information about every line across the image. This also applies to the multi-resolution trace transform described in 11.3.5.1. Extra representations shall be extracted from the trace transform by selecting pairs of bands. The bands in the trace domain correspond to double cones in the image domain as shown in Figure 39. We refer to the bands as trace-bands. The trace-bands shall be integrated to create a 1D band-circus function as shown in Figure 41.

11.3.5.3 Circular Information

Although the trace transform is concerned with extracting lines from an image it is possible to use the trace transform to extract some circular information from an image. A three-dimensional (3D) representation of the image shall be generated in terms of the parameters *d*, θ and *t* as shown in Figure 41. This new representation, which is a 3D volume, is referred to as the trace-cube. The trace-cube carries significantly more information then the conventional trace transform, including information related to non-linear regions in the image, such as circles or ellipses. A slice shall be taken from the trace-cube at d=0, which allows two types of circular information to be extracted.

The extraction process of the first type of information, shown in Figure 42, shall apply functionals along the θ parameter, which is equivalent to applying the functional around circles in the image domain. The result is a one dimensional function in terms of the position on the line *t*, in this case *t* is related to the radius of the circle. We will refer to this function as the trace-circular.

The second type of circular information concerns circular regions in the image domain as shown in Figure 43. By applying a functional over only part of the line in an interval $t_0 \le |t| < t_1$ we shall obtain a one dimensional signal in terms of the angle θ. The values of t_0 and t_1 define the radius of the concentric circles. Although Figure 42 shows $t_0 = 0$, this is not necessary, taking a larger value $t_0 > 0$, would give an annulus. An annulus is made up of two concentric circles. The values of t_0 and t_1 define the radius of the concentric circles. This function will be referred to as the trace-annulus.

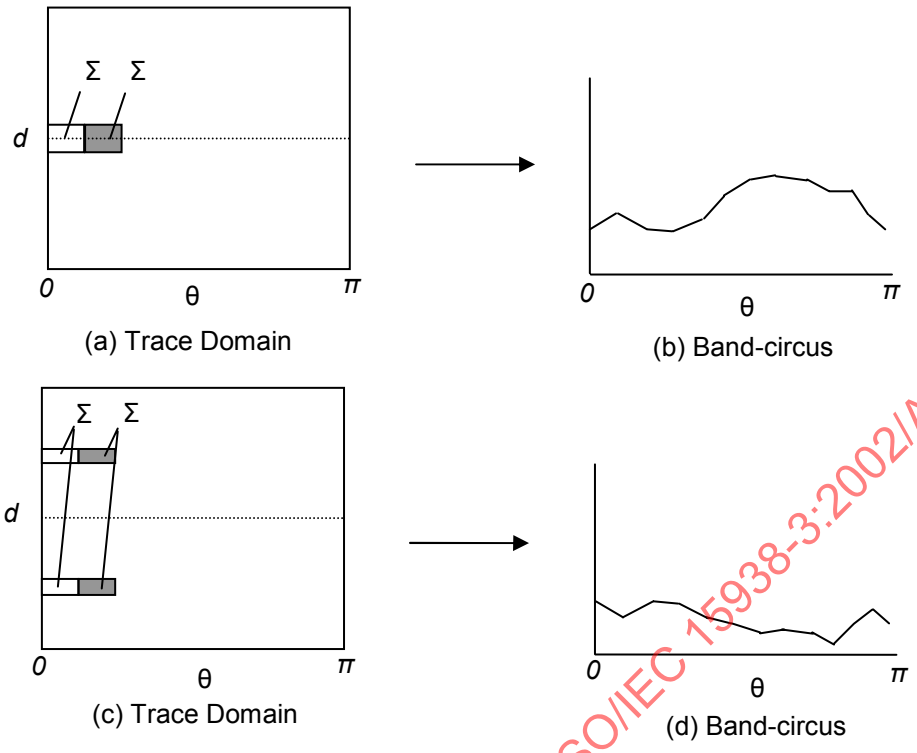

Figure 40 — Band-circus functions are extracted from the trace transform

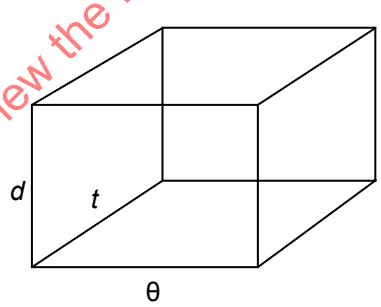

Figure 41 — A 3D representation of the image in terms of the angle θ, distance *d* and position on the line *t*. This representation is referred to as the trace-cube

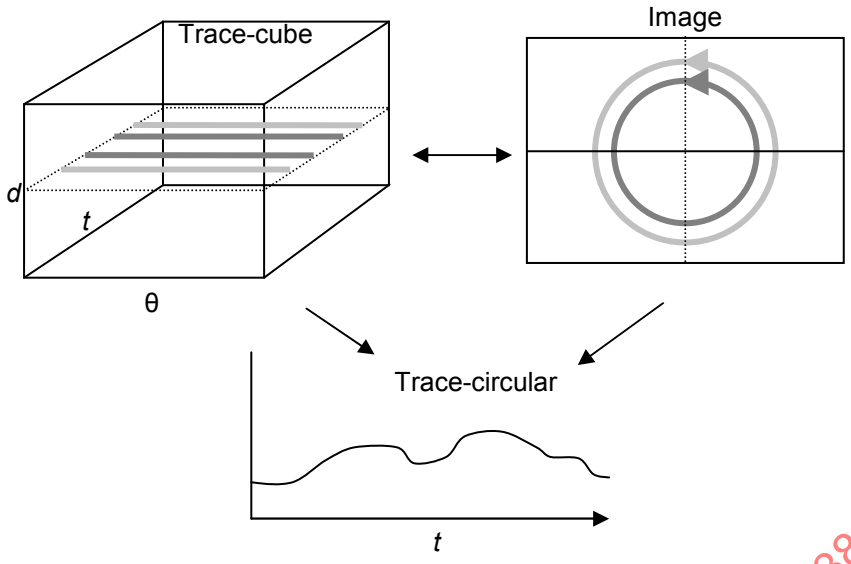

Figure 42 — Information can be extracted from the trace-cube that is equivalent to circular information in the image domain. Functionals can be applied around the circles to obtain a 1D signal in terms of the position on the line t which corresponds to the radius of the circle

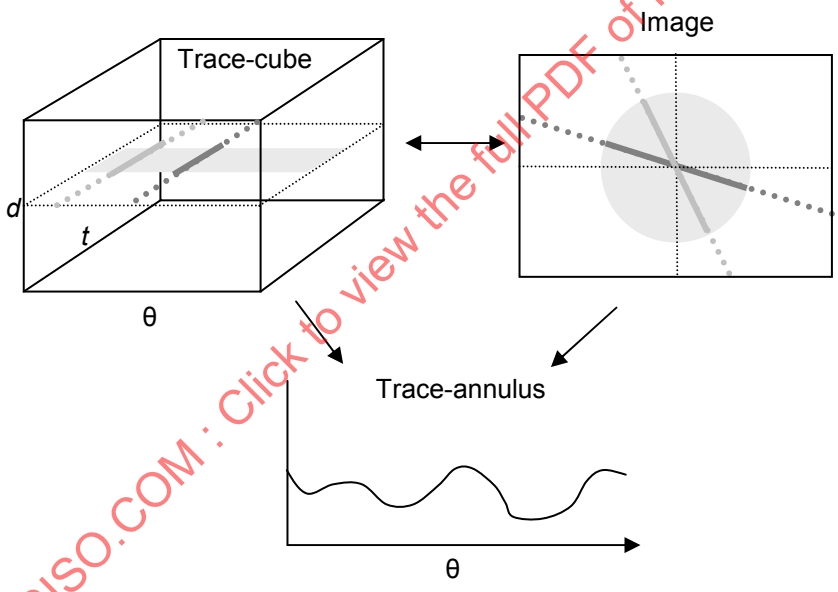

Figure 43 — Further circular information can be obtained from the trace-cube. By applying functionals along intervals in the line position *t* it is possible to obtain information about circular bands in the image

In practice with discrete data d=0 may not be available, in this instance two slices shall be used that correspond to +/-Δd. These two slices shall be combined by adding the resulting trace-circular or trace annulus functions together for the two slices.

11.3.5.4 Global Signature A Extraction Algorithm

The complete algorithm is given in two parts. The first part of the algorithm is a resampling and windowing stage, this includes a number of steps which where found to be beneficial to the overall performance of the algorithm. The resampling and windowing algorithm is presented in Steps 1-3.

- **Step 1** – Resize original image, maintaining aspect ratio, to 192xN or Nx192, where N ≥ 192.
- **Step 2** – Extract a tapered circular region from the centre of the resized image, the circle has a diameter of 192. The circular region has a linear taper of 7 pixels.
- **Step 3** – Filter the image with a two dimensional Gaussian kernel with size 3x3 and standard deviation of 0.95.

The two dimensional Gaussian kernel is

$$G(\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

where σ is the standard deviation.

The main algorithm extracts the four types of representations and then extracts binary identifiers from each representation. The complete extraction algorithm is presented in Steps 4-12.

- **Step 4** – Take the trace transform of the grey scale representation of the image f using the functional $\sum_t f(t)$, recalling that t defines points on a line in the image. That is summing over all lines in the image. 600 evenly spaced samples are taken from the angle θ, sampled from 0 to π and the distance d is sampled 384 times from $-191.5\times(1\div sqrt(2))$ to $191.5\times(1\div sqrt(2))$ with a sampling interval of $(1\div sqrt(2))$.
- **Step 5** – Extract 5 band-circus functions from the trace transform and then sub-sample these by a factor of 6. The bands are taken from intervals in the trace transform located at:

  $U_0 = 0, U_1 = \pm 1.5$

  $U_0 = \pm 17.5, U_1 = \pm 18.5$

  $U_0 = \pm 33.5, U_1 = \pm 34.5$

  $U_0 = 50.8, U_1 = \pm 51.5$

  $U_0 = 67.5, U_1 = \pm 68.5$
- **Step 6** – Extract 3 trace-annulus functions from the trace-cube at $|d| \le 1\div sqrt(2)$ corresponding to:

  $t_0 = 0$, $t_1 = 1\div 3t_{max}$

  $t_0 = 1\div 3t_{max}$, $t_1 = 2\div 3t_{max}$

  $t_0 = 2\div 3t_{max}$, $t_1 = t_{max}$

  and then sub-sample these by a factor of 6. Use the functional $\sum_t f(t)$ to extract the functions.
- **Step 7** – Extract 1 trace-circular function from the trace-cube at $|d| \le 1\div sqrt(2)$ using the functional $\sum_t f(t)$.
- **Step 8** – Obtain the first two basic circus functions by applying the following diametrical functionals to the columns of the 2D matrix resulting from step 4,

  a. $\sum_t |g(t)'|$, where $g(t)'$ denotes the gradient of $g(t)$ with respect to *t*, calculated as $g(t) - g(t+1)$

  b. $\max(g(t))$