MITK Channel-Konzept

Kwong Yung

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Channel, was ist das?

Beispiele um ein neues Image zu initialisieren:

```
virtual void Initialize(const mitk::PixelType& type,
 unsigned int dimension, unsigned int *dimensions,
 unsigned int channels = 1);
```

```
virtual void Initialize(const mitk::PixelType& type,
 const mitk::Geometry3D& geometry, unsigned int
 channels = 1, int tDim=-1);
```

Der Parameter **channels** taucht immer wieder auf, aber was bedeutet das eigentlich??

Definition

Kommentar aus mitk::Image.h

…Image organizes sets of slices (s x 2D), volumes (t x 3D) and channels (n x ND). Channels are for different kind of data, e.g., morphology in channel 0, velocities in channel 1. All channels must have the same Geometry! In particular, the dimensions of all channels are the same, only the pixel-type may differ between channels…

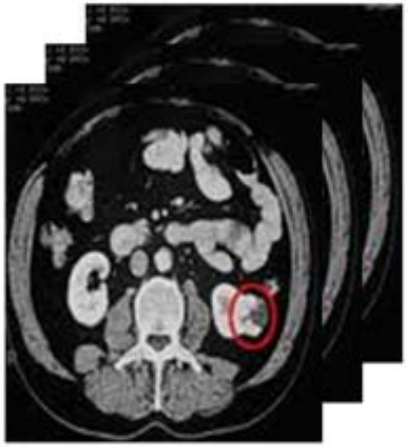
So wars mal angedacht, aber vorsicht, pixel-type muss gleich sein!

Slice / Volume

2D: Slice

3D: Slices / Volume

t=1

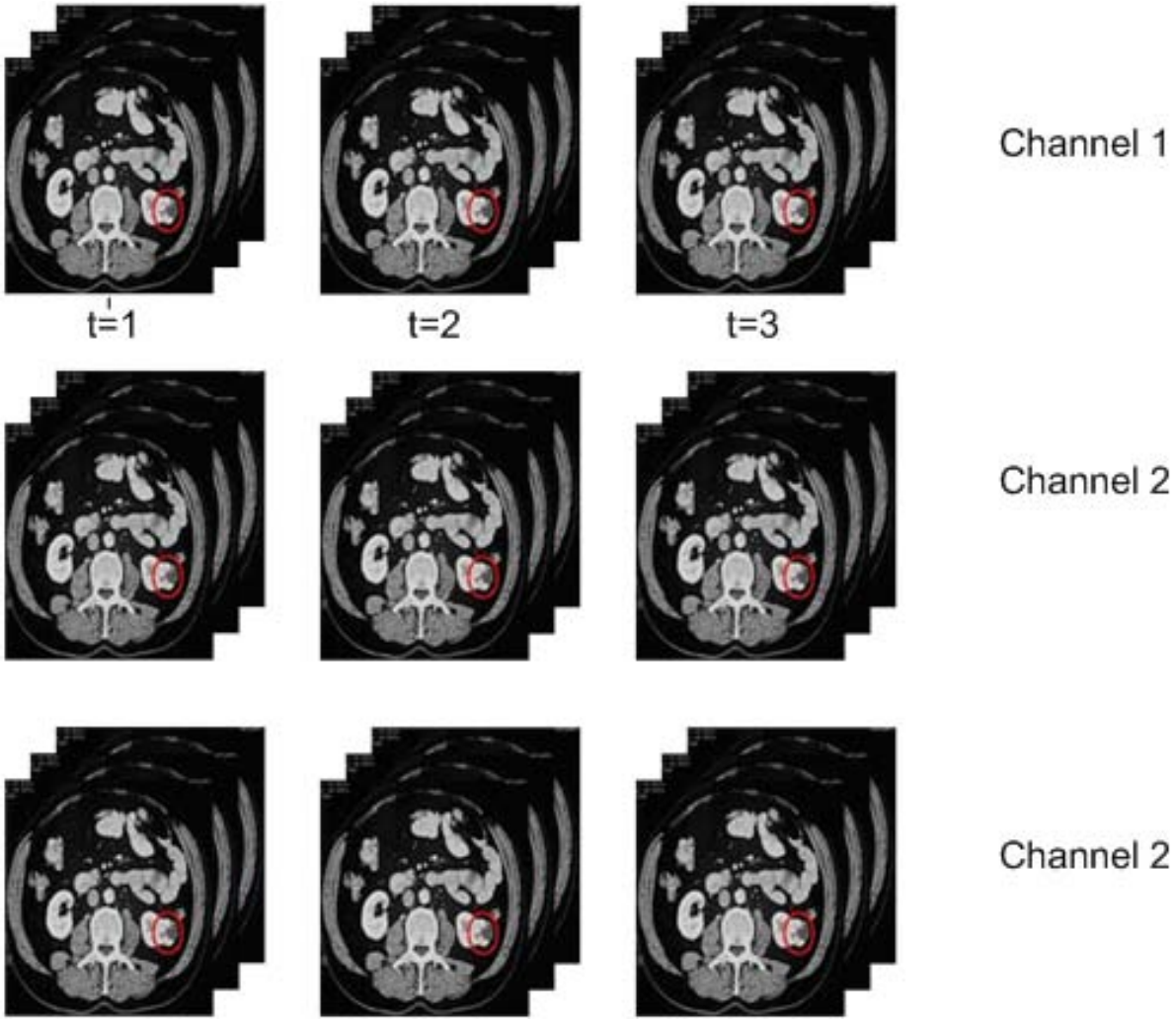


t=2

t=3

4D: Volume + t

Channel



SubImageSelector

```
mitk::ImageTimeSelector::Pointer timeSelector =
       mitk::ImageTimeSelector::New();
timeSelector->SetInput(image);
timeSelector->SetTimeNr(timestep);
timeSelector->UpdateLargestPossibleRegion();
mitk::Image::Pointer i= timeSelector->GetOutput();

mitk::ImageSliceSelector::Pointer sliceSelector =
       mitk::ImageSliceSelector::New();
sliceSelector->SetInput(image);
sliceSelector->SetSliceNr(slice);
sliceSelector->UpdateLargestPossibleRegion();
mitk::Image::Pointer i= sliceSelector->GetOutput();


mitk::ImageChannelSelector //wird nirgends verwendet
```

Zugriff auf Channeldaten

```
float* m_IntensityArray, m_DistanceArray, m_AmplitudeArray;
    …
m_ToFCameraDevice->GetAllImages(this->m_DistanceArray,
       this->m_AmplitudeArray, this->m_IntensityArray, …);
image->SetSlice(this->m_DistanceArray, 0, 0, 0);
image->SetSlice(this->m_AmplitudeArray, 0, 0, 1);
image->SetSlice(this->m_IntensityArray, 0, 0, 2);


void* dData, aData, iData;
dData = this->m_AllImage->GetSliceData(0, 0, 0)->GetData();
aData = this->m_AllImage->GetSliceData(0, 0, 1)->GetData();
iData = this->m_AllImage->GetSliceData(0, 0, 2)->GetData();
```

Image neu

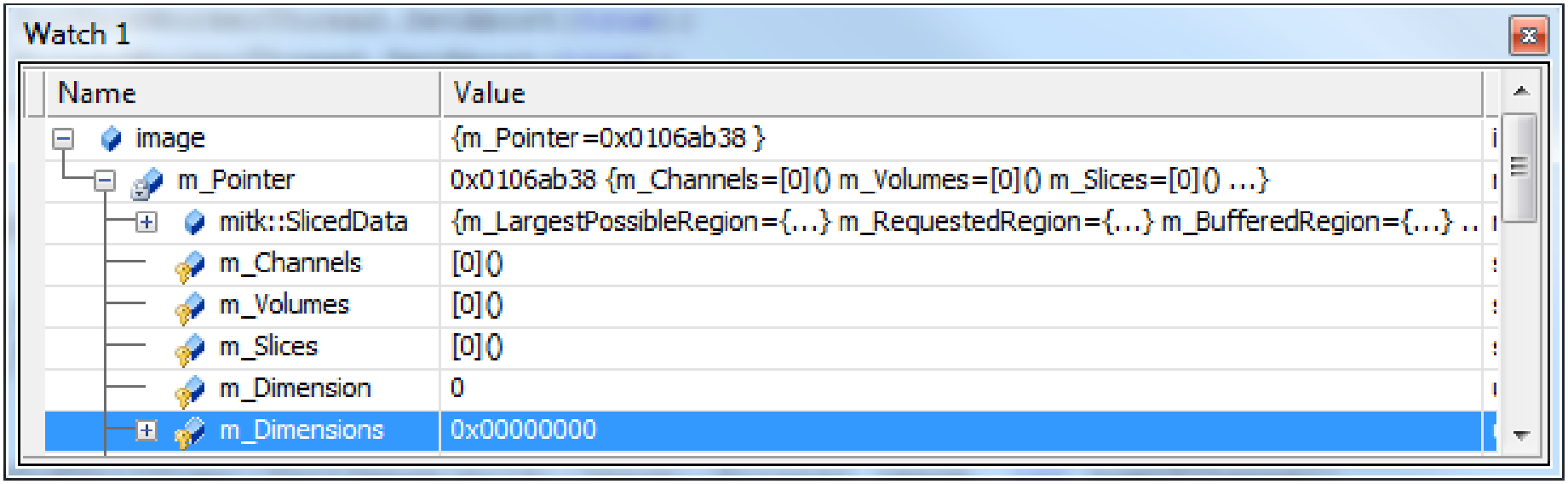

Watch 1
Name
Value
image
{m_Pointer=0x0106ab38 }
m_Pointer
0x0106ab38 {m_Channels=[0]() m_Volumes=[0]() m_Slices=[0]() ...}
mitk::SlicedData
{m_LargestPossibleRegion={...} m_RequestedRegion={...} m_BufferedRegion={...} ..
m_Channels
[0]()
m_Volumes
[0]()
m_Slices
[0]()
m_Dimension
0
m_Dimensions
0x00000000

Image nach Initialize mit Channel = 1

```
dim0] = 204; dim[1] = 204;
image->Initialize(mitk::PixelType(typeid(float)), 2, dim, 1);
```

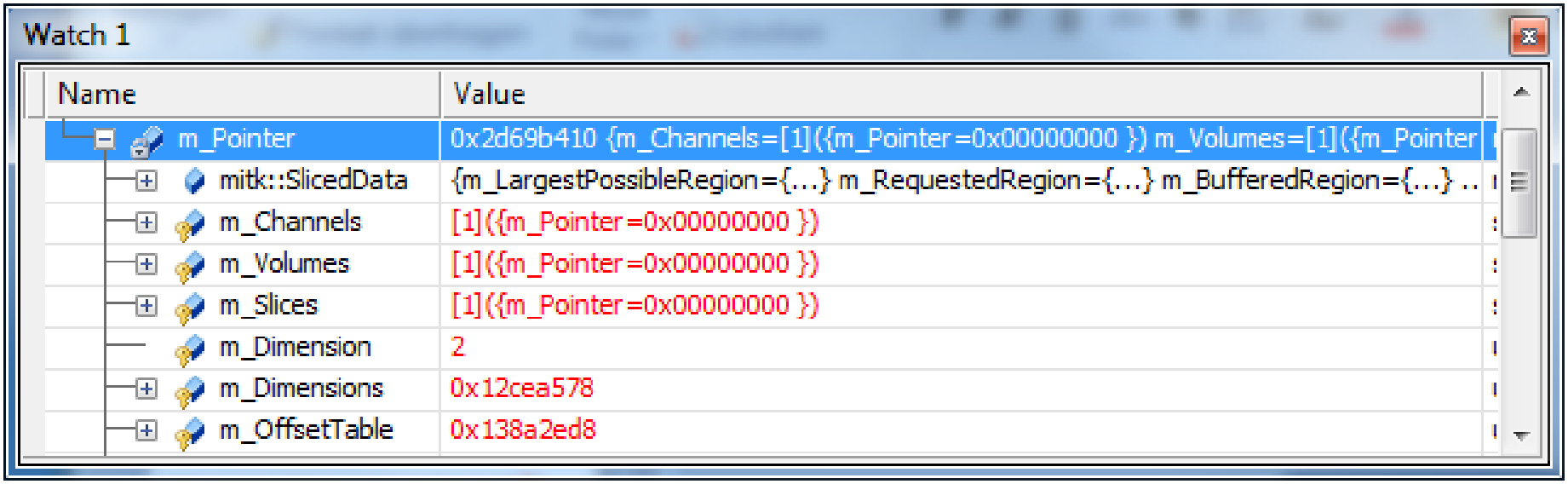


Image nach Initialize mit diversen Parameter

```
Dim[0] = 204; dim[1] = 204; Dim[2] = 2;
dim[3] = 3; // t
image->Initialize(mitk::PixelType(typeid(float)), 4, dim, 2);
```

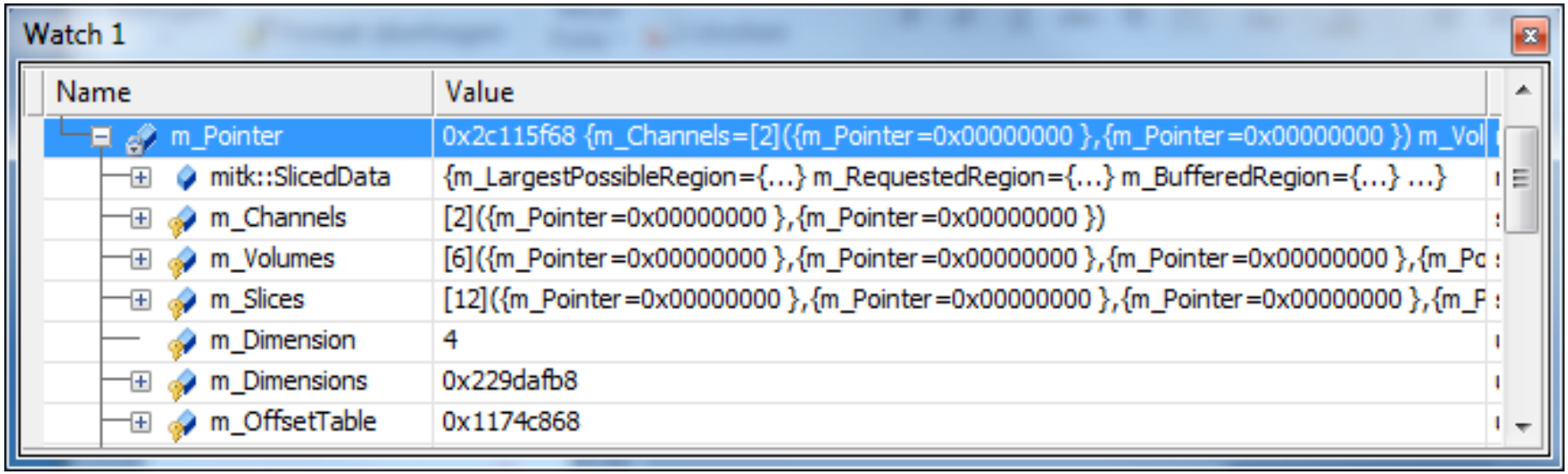


Image nach Initialize mit Channel = 3

```
Dim[0] = 204; dim[1] = 204;
image->Initialize(mitk::PixelType(typeid(float)), 2, dim, 3);
```

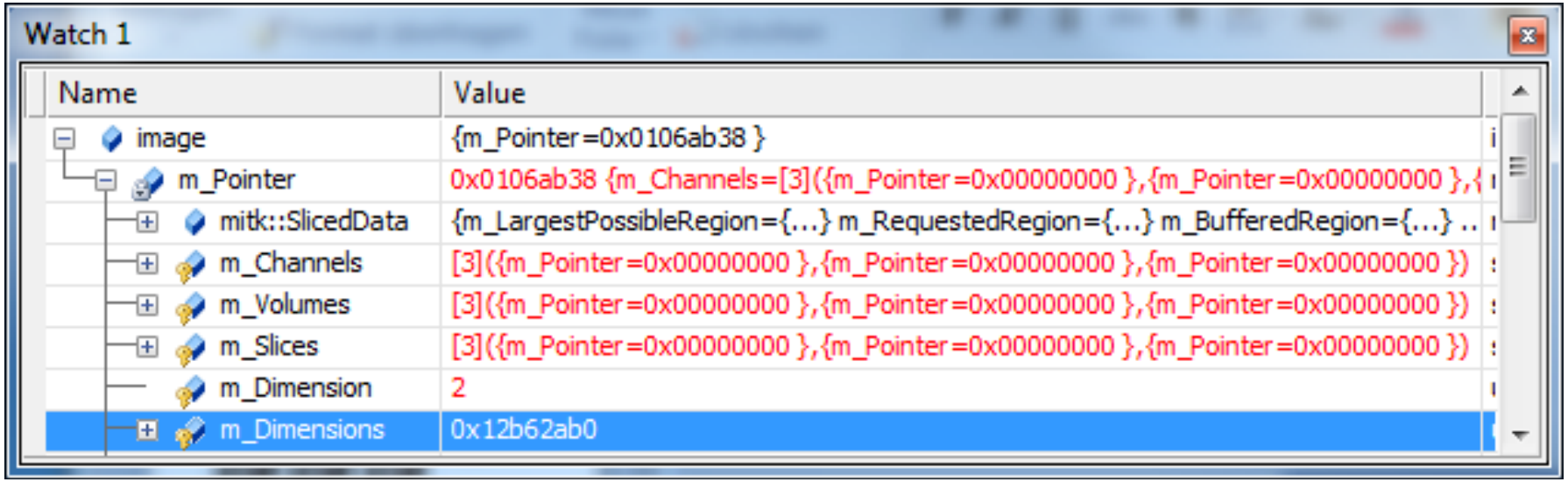
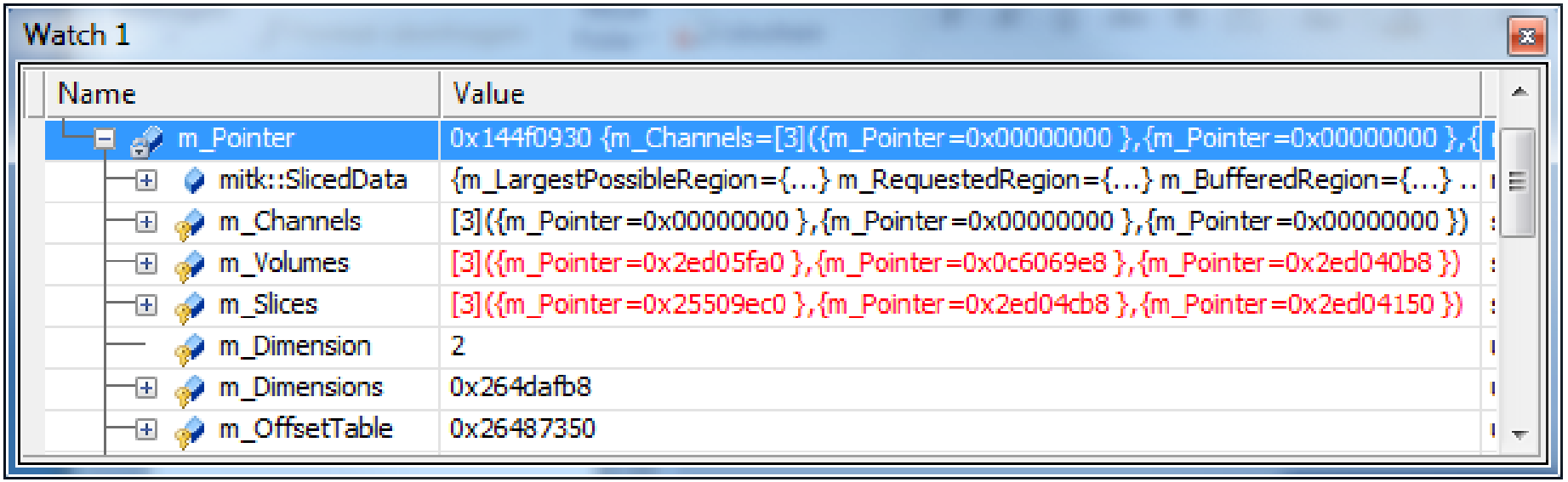


Image nach SetSlice

```
image->SetSlice(this->m_DistanceArray, 0, 0, 0);
image->SetSlice(this->m_AmplitudeArray, 0, 0, 1);
image->SetSlice(this->m_IntensityArray, 0, 0, 2);
```



Channels als 5. Dim?

```
typedef struct mitkIpPicDescriptor
{
  void *data; /* pointer to 'image' data */
  _mitkIpPicInfo_t *info; /* pointer to the PicInfo */
  mitkIpPicType_t  type; /* datatype of the data */
  mitkIpUInt4_t  bpe; /* bits per element */
  mitkIpUInt4_t  dim; /* number of dimensions */
  mitkIpUInt4_t  n[_mitkIpPicNDIM]; /* size of dim n[i] */
} mitkIpPicDescriptor;

#define _mitkIpPicNDIM 8
```

Zusammenfassung

- Channels nur mit Einschränkung nutzbar:
  - Gleicher Datantyp für alle Channels
  - Pic-Reader/Writer funktioniert (noch) nicht richtig
  - Keine GUI Unterstützung („Channel-Slider“ im Image Navigator)