Low-level Characterization of Astronomical DataSets - DRAFT

BY FABIEN CHÉREAU

Virtual Observatory System Department, ESO

April 11, 2008

Abstract

This document investigates the definition of a general characterization model for describing any astronomical observation at a detailed level. It should be read as a complement and a discussion of [3]. The main goals pursued by this document are:

- to define base concepts and definitions useful for the general low-level characterization of DataSets which could lead to a practical implementation of it.
- to use a bottom-up approach to provide feedbacks and refinements on the concepts defined in the high-level characterization document. This include in particular the definition of a strict and non ambiguous method for defining high-level meta data.

1 Introduction

We can split characterization of DataSet into two categories: high-level characterization and low-level characterization. High level characterization meta data are used to describe DataSets as a whole using simple intuitive descriptors such as central position, or maximum spectral resolution. On the other hand, we will call low-level characterization the most detailed statistical description of a DataSet that it is possible to produce. High-level descriptor could be considered as a lossy compression of the low-level characterization.

Many applications such as a search engine or SIA-like service will only use high-level characterization meta data as the ones described in [3]. On the other hand, a detailed science analysis tool such as a SED builder, an image cut-out service, or a data reduction pipe line will need to access and manipulate the lowest level of characterization to perform per-pixel operations with the highest possible precision.

To someone who knows how heterogeneous are most of the astronomical instruments in service in the world, defining a general model for low-level characterization of any astronomical DataSet may sound like an impossible task. However we will show in section 3 that, when reduced to its fundamental concepts, it is possible to transform the problem to something more homogeneous and logical. These concepts then naturally lead us in section 3 to adopt a bottom-up approach to clarify or improve the high-level characterization descriptors. Section 3 is a discussion on how the previous considerations can explain and solve some of the issues faced with the high-level model described in [3].

2 Base concepts

2.1 Scope

2.1.1 Characterization meta-data versus other meta-data

It is important to clearly differentiate characterization meta data from other meta data. In this document we call characterization the description of the physical content of a data set. This means that for example the name of the instrument or the date of release should not be considered as characterization while the spatial resolution, the PSF (Point Spread Function) or the time of acquisition is really a characterization.

2.1.2 Highly reduced data

In this document we investigate only the description of observed (or simulated) DataSets , such as an image or a spectrum. These data sets can be reduced in the sense that some known data processing may have been applied on it (e.g. flat fielding etc..). If we push this definition to the extreme, it is perfectly valid to consider that e.g. a star catalog derived from a set of images is also a highly reduced observation DataSet which can be characterized. However it will not be very efficient (and easy) to use it this way, because it would require that the mapping as described in section 2.2 contains the description of the whole processing performed to generate the star catalog. For this kind of usage, an astronomical object data model will be more appropriate.

2.1.3 Astronomical Frame and units

Astronomical information in the real world can be described in a large number of reference frame and units which can be converted from one to another. They are just different ways of representing the same data. The purpose of the lower level characterization model is not to describe all the possible frames and units that people use, neither to provide a way to convert between them, although this could be provided as a helping library provided with the implementation of the model. In our case, we only need to agree on a common reference system defined as precisely as possible and understood by everyone. Since we are working on astronomy DataSets, the ICRS is a good candidate. Similarly, the choice of a unit for each axis can of course be discussed, the only important point is to choose one and stick to it.

It is important to understand that using a single reference frame and unit is only an internal simplification and does not prevent to describe data sets which are more naturally described in a different frame. The description of the main mapping for each DataSet as described in section 2.2 just need to include the conversion from the native frame and unit to ICRS with the proper units. This part is hidden into the mapping, which allows to keep our model free of any specific conversion algorithm while being flexible and transparent to the user.

2.2 Definitions: DataSet, Spaces, Axes and Mapping

In this document we will use the term of **DataSet** very frequently. A DataSet is the base entity that we want to characterize. and it should be understood as a set of information observed from the real (or simulated) world using an acquisition device, like a telescope with a CCD camera, optionally followed by some data processing.

From a statistical signal processing point of view, a DataSet can be perceived as a signal coming from the real continuous **world space** (e.g. incoming photons) projected into a set of digital samples stored in the discrete **data space** (e.g. image pixel data)[1].

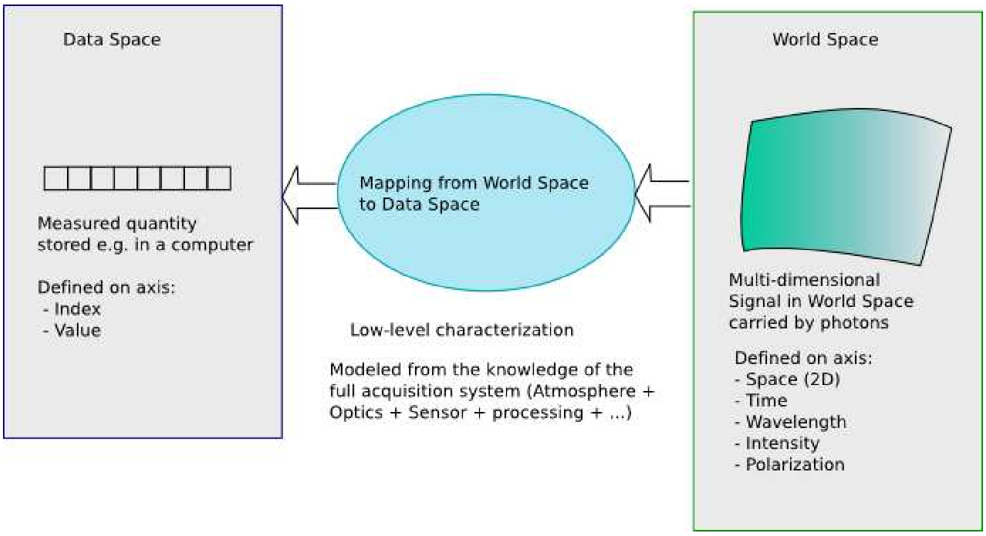
Data Space
Measured quantity stored e.g. in a computer
Defined on axis:
- Index
- Value
Mapping from World Space to Data Space
Low-level characterization
Modeled from the knowledge of the full acquisition system (Atmosphere + Optics + Sensor + processing + ...)
World Space
Multi-dimensional Signal in World Space carried by photons
Defined on axis:
- Space (2D)
- Time
- Wavelength
- Intensity
- Polarization

Figure 1. Representation of the world and data space

[1] We will assume in this document that a DataSet is stored in a digital way although it should be possible to extend it to analogical data

In the world space we assume that information is carried by incoming particles (photons) whose flux is defined on a set of (fixed once for all) axes: **space**, **time**, **wavelength**, **intensity**, **polarization**[2,3]. In the data space, the only two axes are the samples indice and their values (e.g. a C array of integers). See tables 1 and 2 for a detailed description of the axes in both spaces.

Axis	Unit	Description
Space (2D)	deg	(α, δ), ICRS orientation from which comes the photons*
Time	JDay	t, time at which the photons arrive on the sensor
Wavelength	m	λ, wavelength
Intensity	W	I, intensity of the photon flux
Polarization	-, rad, rad	(p, ψ, χ), degree of polarization, azimuth angle, ellipticity angle**

Table 1. Axes of the world space. Value on all axes are continuous.
* ICRS orientation is equivalent to BCRS or GCRS orientations. The orientation is implicitly defined at the observer position and speed which should therefore be provided in the high-level meta data.
** An alternative would be to use the Stokes Parameters Q, U and V

Axis	Type	Description
Sample Index (1D,2D,..)	integer	uni or multi-dimensional index of the data element
Sample Value	various types	scalar value of the data element

Table 2. Axes of the data space. Elements have no physical units since the data space doesn't directly represent something physical.

This set of axes must be defined and consistent for all astronomical DataSets because it defines the common reference frame in which we can compare and manipulate all of them. It is very important to understand that the (arbitrary) choice of the axes defining our world space practically restricts the model to observations of (incoherent) source of photons only. For characterization of more exotic DataSets such as gravitational waves, another world space has to be defined with different axes, and it will not be possible to manage them in a consistent way with photon-based DataSets because they cannot be expressed in a consistent space.

The acquisition device can be modeled by a projection function, or **mapping**, which transforms an input signal in world space to an output signal in the data space. The input signal is a multi-dimensional continuous function defined on each axes of the world space. The output signal is a discrete probability distribution defined on the data space. It forms a k-dimensional random vector X = (X0, X1, ..., Xk -1) which is a collection of random variables, one for each of the k samples of the data set. The joint distribution of this random vector can be described by a k-dimensional probability mass function (pmf) giving the probability that the samples of the DataSet take the corresponding values for a given input signal.

The complex mapping function should ideally contain everything we know about the DataSet acquisition chain. However, in practice, it does not necessarily need to be perfect or complete but only as good and detailed as we need or can. The lack of knowledge of the acquisition system is naturally reflected in the statistical dispersion (variance) of the pmf of X for a given input signal[4].

[2] We want to describe here only the information coming into the sensor in an objective way. Therefore axes such as distance, redshift or other astrometric parameters cannot be defined in this space because they are not directly observed, but derived from a (subjective) interpretation of the observation, which goes beyond the scope of the characterization. Therefore the space motion of the source, parallax, light deflection aberration etc.. are at this point not extracted from the signal.

[3] The utility of the phase axis has to be discussed. It may have a meaning for coherent flux of photons.

[4] And we will see in section 3 that it is itself an inseparable element to take into account for computing high-level characterization meta data.

We just saw that providing a mapping function for a DataSet allows to compute the statistical distribution of the projection of a defined input signal into the data space. Similarly, the reversed mapping, or backward transformation allows to compute the statistical back-projection of a given DataSet into the world space. In this case, the input of the reverse mapping is a vector in the data space and the output a random process in the world space. The reverse mapping can be computed using the basic inverse image formula although it may be very tricky in practice. These two considerations lead to the conclusion that the knowledge of a mapping function for a DataSet is a full model of the acquisition device and constitutes the lowest level characterization of a DataSet.

3 A bottom-up approach for deriving high-level meta data

For many usage such as SIA/SSA like services, we are only interested in high-level meta-data, such as the position, the spatial resolution or the spectral coverage of a DataSet. However there is today no standard agreed upon way of defining them. For example, the spatial central position of a DataSet will be in some case given as the position in the sky of the central pixel of the data, sometimes as its barycenter, and even sometimes as the position of the observed source (in the case of spectrum). To this confusion must also be added the one caused by the use of many different coordinate systems and units. As a result of this complexity, it is today still very difficult to use high-level descriptors in a program in a fully automatic way and with guarantees of validity and completeness.

Because the low-level characterization provides the most detailed description of a DataSet, it should be possible to use it for generating automatically and generically a set of high-level descriptors. A DataSet described by such a low-level characterization would then guarantee to have high-level descriptors which are consistent and complete because computed strictly the same way. The choice of which high-level descriptors to define and the full and non-ambiguous definition of them could be one of the main task of the characterization working group. Using fixed units and reference frames as defined in 2.1.3 would also much simplify the processing by avoiding all the programs to re-implement all kind of unit and frame conversions. In this section we will attempt to give a formal definition of more or less intuitive high-level characterization concepts such as coverage, resolution and error.

3.1 Coverage

Intuitively, it is pretty easy to imagine what the coverage of a DataSet means on all the axes. It is usually broadly understood as “the region in the world space for which the DataSet contains some valuable information”. However, we will see that although this notion is quite intuitive, it is not easily formally described.

To understand the problem, let us consider the simple example of a 2D image as shown on figure 2. The rectangle on the figure represents the intuitive “image border” obtained by back-projecting the pixels position in the sky using a WCS-like transformation. We see that the sky position B (Ra2, Dec2) is clearly “inside” the “image borders”. We also see that there is a bright star at point A (Ra1, Dec1) which is slightly outside the “image borders” and that the Point Spread Function (PSF) is overlapping the image. In this case, if we know a model of the PSF, we can in theory compute an estimate of the position and brightness of the bright star from the image pixels. We therefore see that the DataSet really contains some valuable information about a point which lays “outside” the “image border”. We also see that the spatial distance to the “image border” from which a source still can be partially observed depends on the shape of the PSF and of the brightness of this source, i.e. on the flux axis. We can even go further if we imagine that our PSF model also depends on the wavelength and on the time. We then see that the coverage becomes a complex function of many axes.

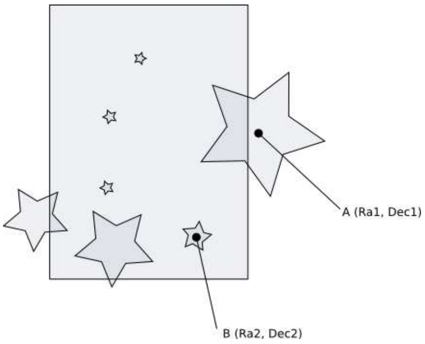

Figure 2. The definition should say whether A and B are included into the coverage support or not.

In the general case, what we need is a function giving for each point of the world space a scalar "sensitivity" value. This **sensitivity** function could be assumed to be the multi-dimensional equivalent to the sensitivity as defined in [3]. Then, by triggering, bounding and averaging it, it also automatically provides the multi-dimensional equivalent to the **support**, **bounds** and **location** as defined in [3]. From these multi-dimensional descriptors it is also easy to define the mono-dimensional (or bi-dimensional for the spatial axis) support, bounds and location for each axis by simply projecting them on each axis. The **filling factor** can also be defined by dividing the support area by the bound area.

We have now seen that the coverage descriptors on all axis can all be simply derived from the multi-dimensional sensitivity function. This sensitivity function is by nature a multi-dimensional function of all the axes of the world space. It should reflect the value or quality of the information that the DataSet contains for each of these points. In the next section, we will attempt to formally define the sensitivity using the concepts from section 2.

3.2 Sensitivity, Resolution and Error

We intuitively feel that the sensitivity function should be totally independent of the actual pixel content of the DataSet. Indeed, it should describe the sensitivity of the acquisition device for any input signal from the world space which obviously does not depends on what was observed on a specific DataSet. This is a hint that it must be possible to define it from the mapping only. Also, beyond what has been said in the previous section, the sensitivity function must have specific properties so that it corresponds to the intuitive idea that we have from coverage. Notably, the sensitivity value associated to a very faint signals totally lost into random noise or to a saturated signal should be close to zero because the DataSet contains no valuable information about it. Similarly the sensitivity value associated to signal which spatially lays far outside the field of view should also be close to zero, and will bound the coverage on the spatial axis. Another example is the case of a fully saturated detector, e.g. when observing a very bright object, which should have a sensitivity exactly equal to zero, and will actually give the higher bound of the coverage on the flux axis.

These considerations let us think that the SNR (Signal to Noise Ratio) should be taken into account for the definition of the sensitivity. However, the last example shows that the SNR taken as-is is not enough because the SNR of a saturated signal is very high (because the variance is very low) while the sensitivity should be null. We propose in the next section an idea for a function defining the sensitivity.

3.2.1 A sensitivity function

Let a mapping M defining our acquisition chain and a multi-dimensional Dirac signal δ_x defined in the world space, as well as $\delta_{x+\mathrm{dx}}$, being the same signal to which is added a small delta on the x axis (for example the x axis could be the spatial or time axis). We can call $S = E[M(\delta_x)]$ the mean of the distribution of the projection of δ_x and $S' = E[M(\delta_{x+\mathrm{dx}})]$ the one of the projection of $\delta_{x+\mathrm{dx}}$.

An example of a sensitivity function could be the sum of a differential signal to noise ratio on all the pixels for all the axis.

$$\text{Sensitivity} = \sum_{x \in \text{axes}} \sum_i \left(\frac{\Delta \text{Sx}_i}{\sigma_i}\right)^2 \tag{1}$$

with $\Delta \text{Sx}_i = |S_i' - S_i|$ the difference between the value of S and S' at pixel i and σ_i the variance at pixel i when δ is varying on axis x. With this formula, we see that in the case the distribution of the projections of δ_x and $\delta_{x+\text{dx}}$ are similar, $M(\delta_x) = M(\delta_{x+\text{dx}})$ we obtain $S_{\text{ensitivity}} = 0$. This will happen as wished when the signal is completely dominated by noise (which happens when the signal is fully "outside" the coverage of the DataSet), and also when the signal is fully saturated for all samples, i.e when the value on the flux axis is 'above" the coverage bound (in this case the joint distribution is a Dirac with all pixels at the saturation value).

This is just an example of a sensitivity function and could probably be defined differently. The advantage of using only the mean and variance of the distribution in the data space is that the values can be practically computable. Other sensitivity functions could be derived using for example a more information-theory based approach, by computing for example the Kullback-Leibler divergence of the distributions.

3.2.2 Resolution and Error

As suggested by previous studies reported in [1], the measure of this differential SNR can in fact also be considered as a measure of the **resolution** as well as of the **error** for a given axis. Work by Falconi [2] showed that the angular precision with which a single target position can be measured is equal to the ratio of some constant (of the order of the Rayleigh limit), which is called the resolution scale, to the SNR. With SNR defined as in equation 1 by

$$\text{SNR}_x = \sum_i \left(\frac{\Delta \text{Sx}_i}{\sigma_i}\right)^2 \tag{2}$$

We could therefore define the resolution limit on a given axis as the value of the small delta |dx| for which the $\text{SNR}_x = 1$, i.e. when the small change is likely to be detected.

Of course such a definition of resolution is quite different from the classical Rayleigh resolution but it has the big advantage to be generical enough to be consistently defined on all the axes. It has notably the property of depending **only** on the random error (SNR) **and** on the accuracy of the knowledge of the mapping which introduces systematics. Consequently, the actual spatial resolution of a DataSet can be much higher than the Rayleigh one[5].

It is also important to note that this definition of resolution inherently includes the noise caused by **sampling** and quantization (which can be modeled by a convolution by a sinc function), and therefore renders the one defined as a high-level descriptor in [3] useless.

3.3 Summary of high-level descriptors

Table 3 show a summarized list of all the high-level descriptors and how to compute them from the low-level characterization.

[5] This can be understood if we take the example of a noiseless data set for which we know perfectly the mapping. In this case, a maximum likelihood method could be used to reconstruct the input signal with a precision as good as we want, i.e that the resolution is infinite.

Descriptor	How to Compute
axis:Location	Barycenter of Sensitivity
axis:Bounds	Min/Max box for area where sensitivity>threshold
axis:Support	Polygons where sensitivity>threshold
axis:Filling Factor	Bounds/Support
axis:Resolution Limit*	Resolution limit for the given axis as defined in section 3.2.1
Observer Position	pos and speed in ICRS

Table 3. List of high-level descriptors. The ones starting with "axis:" are defined for all of the axis of the World Space. The sensitivity function itself can not be stored into the high-level descriptor and is therefore only implicitly contained in the low-level descriptor.
* With the definition of resolution given in section 3.2.1, Minimum Error and Resolution Limit are in fact the same thing.

4 Features of a low-level characterization and implications on high-level descriptors

In [3] as well as in email exchanges, a number of problematic points have been mentioned which seem not to fit nicely into the high-level characterization model. We show in this section that some of these issues can be solved when considering a low-level characterization model based on the concepts explained in section 2 and deriving from it the high-level descriptors as described in section 3.

4.1 Dependency between axes

One of the main issue mentioned in [3] is the handling of dependent axes. For example an optical system with optical aberration presents a dependence between the spatial and the wavelength axes. This means that at a certain level of precision, there is no correct formula projecting the position (in world space) of an incident light ray to a position in the data space (e.g pixel in an image) if this formula doesn't take into account the wavelength as an input parameter. If we extend this analogy to the other axes, we realize that in the general case we should not assume that any axis is independent of the others. The fact that this is sometimes the case for a given precision should therefore just be considered as a specialization (or an optimization for faster computing). In our lower level characterization, this fact is naturally taken into account because the main mapping function which project from world space to data space takes into input a function of all the axes. What is done internally in the mapping can or not include inter-axis dependency without breaking the model.

The high-level descriptors obtained in section 3 take naturally this fact into account because they are derived from a multi-dimensional low-level characterization.

4.2 Characterize group of DataSet as well as subset of DataSet

Another important issue is the validity of the model for a group of DataSet as well as for any arbitrary subset of a DataSet . If the model is correct, it should indeed be able to treat a collection of DataSet or even 1 single pixel of a DataSet exactly the same way since both can be viewed as DataSets as well. To understand how this can be easily expressed using the concepts of the previous section, let's imagine that we have 2 DataSets and that for each of them we know the main mapping from world space to data space. Now if we want to consider both DataSets as a single composite one, it is enough to append the samples from one to the other, and to create a new global mapping by encapsulating the 2 mappings into the new one (see figure 3). Each of the 2 previous mappings will apply the projection function to the related subpart of the data space. Similarly, it is possible to create a new DataSet from the subset of another one simply by using its main mapping applied to a subset only of the data space.

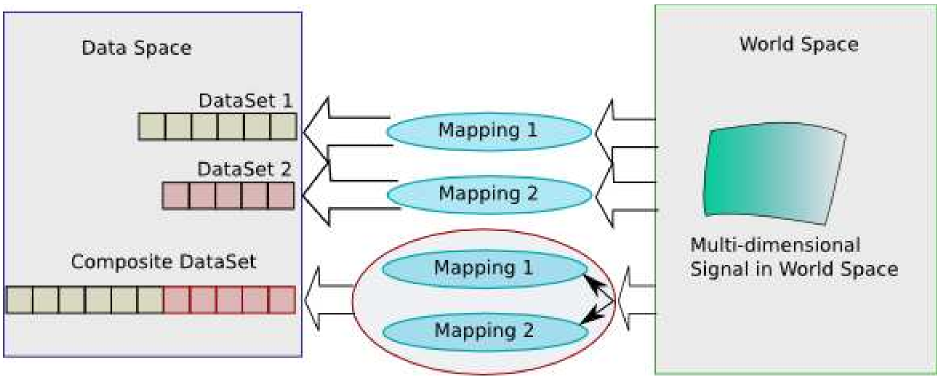

Figure 3. Creating a new DataSet by composing 2 other

4.3 Accuracy, errors and completeness of calibration

Error management is one of the fundamental point of the characterization. We saw in the first section that the output of the mapping in the data space is not a single value for each sample, but a discrete probability joint distribution (it is a random vector). This means that for a given input signal, we know for each sample of the data space the probability that it takes a given scalar value. This probability is given by the pmf of the distribution. This distribution function reflects not only the modeled random error caused e.g. by thermal noise, but also the lack of accuracy in the description of the observing device. The main mapping for a DataSet therefore contains the modeling of all errors and knows how to propagate them into the data space. It can therefore be considered as the lowest level characterization of the errors for a DataSet.

To make things clearer, let's suppose for example that we have an image described by its associated main mapping. If we know the astrometric calibration of the image, we know that a point source in the sky will be projected at a given position P in the data space. However the calibration is never perfect and follows in reality an error distribution. If the error is Gaussian, the mapping function will describe that the projection of the point source is a Gaussian probability distribution centered on P. In this case the values of the samples around P in the data space will have their joint pmf modified accordingly. We therefore see that an random estimation error on the spatial axis is reflected as another random error on the value and indice axes of the data space. It is also very important not to assimilate this distribution as the one of the intensity (flux) axis. This axis exist only in the world space, but we are now describing distributions functions in the data space.

Because we assume that our observed signal comes from the world space, it exists by definition on all of the axes. If the observing device ignores some of the axes (like an imaging device usually ignores the polarization axis), it is reflected in the main mapping by the fact that the inputs for these axes has no incidence on the projected result in the data space. Furthermore, if the calibration of DataSet is partially known for some of the axes, this must also be reflected in the main mapping by increasing the resulting standard deviation of the value and indice axes in the projected data space. If some part of the calibration is totally missing (which is normal if we don't need it for science purposes) the mapping should simply introduce infinite errors. The last point is very important to ensure completeness of the lower level characterization for all described DataSets. Having a complete characterization with a valid error distribution model ensures that algorithms can safely use all the DataSet in a generic way without making assumptions on the value of missing data.

Bibliography

[1] A. J. den Dekker and A. van den Boss. Resolution: a survey. *J. Opt. Soc. Am. A*, 14:547–557, 1997.

[2] O. Falconi. Maximum sensitivities of optical direction and twist measuring instruments. *J. Opt. Soc. Am.*, 54:1315–1320, 1964.

[3] IVOA Data Model Working Group. Data model for astronomical dataset characterisation, version 1.12. *IVOA Recommendation*, 2007.