

SNAP Geocoding Implementation Update

Motivation

TODO

Definitions

This chapter specifies and defines expressions used throughout this document to establish a common vocabulary.

Expression	Definitions
ForwardCoding	A geocoding transformation that converts pixel positions in (x/y) pixel raster to geo-locations in (lon/lat) coordinate space.
GeoLocation	A point on the Earth surface defined by longitude and latitude values.
InverseCoding	A geocoding transformation that converts geo-locations in (lon/lat) coordinate space to pixel positions in (x/y) pixel raster.
PixelLocation	A point in the measurement pixel raster defined by x and y location. A pixel location uses fractional accuracy to take the pixel reference location into account. The integer part of the pixel location defines the raster position, the fractional part the reference location in a single pixel. An offset of (0.0, 0.0) denotes the upper left corner, (0.5, 0.5) the pixel centre, and so forth.
TiePoint	In a data product where the resolution of the geo-location data is lower than the resolution of the measurement data, a TiePoint denotes a raster position with a defined geo-location. Intermediate raster position locations need to be interpolated.

Framework

- ComponentGeoCoding as interfacing class to the user/API.
- Separate subclasses for forward and inverse coding functionality.
- Forward coding: get (lon/lat) for (x/y), i.e. retrieve geo-location from pixel-position. Can be realised by either direct access to the geo-location data (pixel geo-coding) or by interpolating in a sub-sampled geo-location raster (tie-point geo-coding).
- Inverse coding: get (x/y) for (lon/lat), i.e. retrieve pixel-position from geo-location. Is realised by either a clever minimum distance search (pixel geo-coding) or an interpolation on a sub-sampled geo-location raster (tie-point geo-coding).
- ComponentGeoCoding performs common analysis tasks: detection of anti-meridian crossings and detection of pole locations. These checks can be skipped, if it is known that the data will never cross the anti-meridian or the sensing geometry will never allow a pole to be seen.

- Anti-meridian detection: step around the longitude raster and calculate the longitude-difference between two border-pixel longitudes. If a jump larger than 270 deg. is detected, an anti-meridian crossing is detected.
- Pole detection: based on the pixel ground resolution, a delta angle to the pole is calculated. Then the latitude raster is checked for pixels exceeding this threshold latitude. For each of the candidate locations, the surrounding longitude raster is checked for anti-meridian crossings. If an odd number of crossings is detected, the candidate location is considered to be a pole.

Forward Codings

PixelForward

Simple forward implementation for per-pixel based geo-codings. This forward coding does not take any sub-pixel offsets into account, instead it truncates the fractional pixel location to the integer part and returns the geo-location stored at this raster position.

TiePointBilinearForward

A geocoding for tie-point based geolocation products. This forward coding uses bilinear interpolation to calculate geo-locations for pixels between the tie point raster support points.

If the product intersects with the anti-meridian, the implementation internally switches to sine/cosine based interpolation rasters, which introduces small offsets at the tie-point locations. This, in turn, introduces small interpolation errors even on tie-point locations in the range of some centimetres – which in most cases can be neglected.

TiePointSplineForward

A geocoding for tie-point based geolocation products. This forward coding uses spline interpolation to calculate geo-locations for pixels between the tie point raster support points, which better approximates the ellipsoid surface than the bilinear interpolation.

The runtime performance is comparable to the TiePointBilinearForward implementation, while the measured interpolation errors are reduced by approximately a factor of 2.

Inverse Codings

PixelQuadTreeInverse

Inverse geo-coding for pixel based geo-location data. This implementation uses a quad-tree based search algorithm to identify the closest (x/y) position for the given geo-location. This implementation does not take fractional accuracy into account, it merely returns the closest (in a geodetic sense) full pixel position.

Indication: fast initialization, good look-up time. Use for single pixel or transects data-extraction.

PixelGeoIndexInverse

Inverse geo-coding for pixel based geo-location data. This implementation uses a two step search algorithm identify the closest (x/y) position for the given geo-location. In a first step, the inverse coding performs a geo-index lookup to identify a small candidate vicinity (approx. 3x3 pixels). This step is followed by a local minimum-distance search in the target region. The closest (x/y) position is returned as result.

This implementation does not take fractional accuracy into account, it merely returns the closest (in a geodetic sense) full pixel position.

Indication: slow initialization, increased memory requirements, very fast look-up time (approx. 4 times faster than quad-tree). Use for systematic processing of complete raster.

TiePointInverse

- Inside/outside detection strategies for inverse codings

Java API

ComponentGeoCoding

Constructors

ComponentGeoCoding(GeoRaster geoRaster, ForwardCoding forwardCoding, InverseCoding inverseCoding)

Constructs a GeoCoding with given GeoRaster, ForwardCoding and InverseCoding. No GeoChecks will be performed during initialize phase. Defaults to WGS84 CRS. Forward and/or Inverse coding can be null.

ComponentGeoCoding(GeoRaster geoRaster, ForwardCoding forwardCoding, InverseCoding inverseCoding, GeoChecks geoChecks)

Constructs a GeoCoding with given GeoRaster, ForwardCoding, InverseCoding and GeoCheck definition to be executed during initialize phase. Defaults to WGS84 CRS. Forward and/or Inverse coding can be null.

ComponentGeoCoding(GeoRaster geoRaster, ForwardCoding forwardCoding, InverseCoding inverseCoding, CoordinateReferenceSystem geoCRS)

Constructs a GeoCoding with given GeoRaster, ForwardCoding, InverseCoding and CRS. No GeoChecks will be performed during initialize phase. Forward and/or Inverse coding can be null.

ComponentGeoCoding(GeoRaster geoRaster, ForwardCoding forwardCoding, InverseCoding inverseCoding, GeoChecks geoChecks, CoordinateReferenceSystem geoCRS)

Constructs a GeoCoding with given GeoRaster, ForwardCoding, InverseCoding, GeoChecks to be performed during initialization and CRS. Forward and/or Inverse coding can be null.