



PixScape 1.2

User manual

Gilles Vuidel, Yohan Sahraoui and Jean-Christophe Foltête

2019-11-29

Contents

1	Introduction	3
1.1	About PixScape	3
1.1.1	Authors	3
1.1.2	Terms of use	3
1.2	System requirements	3
1.3	Installation and launching PixScape	4
2	Graphical user interface (GUI)	5
2.1	Project creation	5
2.2	Data management	6
2.2.1	Additional layers	6
2.2.2	Multiscale	6
2.3	Visibility calculation	7
2.3.1	Viewshed	8
2.3.2	Tangential view	10
2.3.3	Multi-viewshed	12
2.3.4	Eyesight bounds	15
2.4	Visibility metric calculation	16
2.4.1	View parameters	16
2.4.2	Sampling	17
2.4.3	Metric parameters	17
2.4.4	Results	18
2.5	Tools and options	21
2.5.1	Add eyesight attributes	21
2.5.2	Options	22
3	Command line interface (CLI)	24
3.1	Starting	24
3.1.1	Launch PixScape in CLI mode	24
3.1.2	Syntax	24
3.2	Commands	25
3.2.1	-help : show help screen	25
3.2.2	-metrics : show metrics	25
3.2.3	-project : load project	26
3.2.4	-landmod : land use changes	26
3.2.5	-viewshed : planimetric view or viewshed	26
3.2.6	-viewtan : tangential view	27
3.2.7	-multiviewshed : multiple viewshed	28
3.2.8	-planmetric : metric in planimetric view	29
3.2.9	-tanmetric : metric in tangential view	29
3.2.10	-toobject : point to point visibility	30
3.3	Metric parameters	31

3.3.1	Land use	31
3.3.2	Distances	31
3.4	Options	32
3.4.1	Parallelization : -mpi, -proc, -cuda	32
3.4.2	View settings : -zeye, -zdest, -bounds, -earth	32
3.4.3	Sampling : -sampling	32
3.4.4	Multiscale : -multi, -mono	33
3.4.5	Location results : -resdir	33
3.5	Examples	33
3.5.1	Tangential metrics with point sampling	34
3.5.2	Planimetric metrics with grid sampling	34
4	Description of visibility calculations	35
4.1	Planimetric view	35
4.1.1	Reverse planimetric view	36
4.2	Tangential view	37
4.3	Multiscale	38
4.4	Eyesight bounds	39
5	Metrics description	41
5.1	Common metrics	41
5.1.1	Area: A	41
5.1.2	Shannon diversity index: S	42
5.1.3	Aggregation index: AG	42
5.1.4	Contagion: CONTAG	42
5.1.5	Edge density: ED	42
5.1.6	Patch density: PD	43
5.1.7	Patch Mean Size: PMS	43
5.1.8	Distance distribution: DIST	43
5.2	Planimetric metrics	43
5.2.1	Perimeter: P	43
5.2.2	Compactness: C	44
5.2.3	Fractal dimension: FD	44
5.3	Tangential metrics	44
5.3.1	Skyline: SL	44
5.3.2	Shannon depth: SD	44
5.3.3	Depth line: DL	45
6	Performance tuning	46
6.1	Parallelism to speed up execution	46
6.1.1	One computer : threads	46
6.1.2	Graphics card : CUDA	47
6.1.3	Computer cluster : mpi	47
6.2	Memory management	48
6.2.1	Graphical user interface	48
6.2.2	Command line interface	48

Chapter 1

Introduction

1.1 About PixScape

PixScape software is dedicated to the analysis of the landscape visibility from raster data. This software integrates the main functionalities available in standard GIS in this domain and offers other original features such as tangential view and multi-resolution analysis.

1.1.1 Authors

PixScape has been developed at [ThéMA](#) laboratory ([Université de Franche-Comté – CNRS](#)) by Gilles Vuidel in collaboration with Jean-Christophe Foltête, Daniel Joly, Yohan Sahraoui et Samy Youssoufi. The PixScape logo has been designed by Xavier Girardet.

1.1.2 Terms of use

PixScape is distributed in open source, under the [GPL](#) license.

Source code can be downloaded from the sourcesup git repository : [git://git.renater.fr/pixscape.git](https://git.renater.fr/pixscape.git).

Users must cite the following reference in their publications :

Sahraoui Y., Vuidel G., Joly D, Foltête J-C., 2018. [Integrated GIS software for computing landscape visibility metrics](#), Transactions in GIS, 22 (5), 1310-1323.

1.2 System requirements

PixScape runs on any computer supporting Java 8 or later (PC under Linux, Windows, Mac, ...). However, when dealing with large datasets, the amount of memory (RAM) will limit the maximum area that can be processed. In addition, processing power (CPU) will determine the speed of computing. For details, see section [Performance tuning](#).

For GPU computing, PixScape supports only Nvidia graphics card with CUDA. By default, PixScape is compiled with CUDA 6.5. If you cannot install CUDA 6.5, you will need to install the JCuda library corresponding to your CUDA version and launch PixScape in command line (see [Graphics card : CUDA](#)). JCuda binaries are available here: www.jcuda.org/downloads/downloads.html

1.3 Installation and launching PixScape

PixScape is available for downloading at : <http://sourcesup.renater.fr/pixscape>.

- Download and install Java 8 or later (java.com). It is best to install 64 bits version of Java.
- Download pixscape-1.2.jar
- Launch pixscape-1.2.jar

PixScape can be used through 2 user interface : the graphical user interface (GUI) and the command line interface (CLI).

The graphical user interface allows to create project, visualize the data, calculate interactively viewsheds and visibility metrics, and perform exhaustive visibility metric calculation on local computer. The command line interface allows to run the same calculations on remote computers as well as on computer clusters.

The following chapter (2) describes all of the software features available from the GUI. Chapter 3 details how to use PixScape on the command line. Chapter 4 explains how visibility calculations are implemented in PixScape. The set of available metrics is detailed in Chapter 5. And Chapter 6 reviews the parallelization modes, performance and memory management.

Chapter 2

Graphical user interface (GUI)

Once PixScape is launched, the first step is to create a project based on a DTM (Digital Terrain Model) covering the studied area.

2.1 Project creation

To create a new project, use the menu File / New project. User must enter a name and a path for the project as well as a DTM in Tiff or AsciiGrid format. **The coordinate system of the DTM must be projected and in meter unit. If this is not the case, the visibility calculations will be erroneous.** The resolution of the altitudes (Z) can be specified if it is not in meter. For example, if the altitudes are given in centimeter, enter 0.01 in "Z resolution".

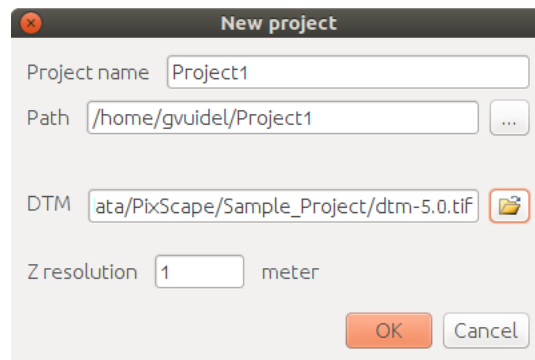


Figure 2.1: Creating a project

The created project stores in its folder, the DTM converted in Tiff format with altitudes in meter, and the XML file containing the project settings. For opening a project, simply select the corresponding XML file from the File / Load Project menu.

The DTM is sufficient to perform visibility calculations, but it can be more interesting to add additional information to perform more accurate analyses.

2.2 Data management

For importing raster data, PixScape supports 2 file formats : Tiff and AsciiGrid. For Tiff format, the coordinate system can be given by 2 ways : directly in the Tiff file with the GeoTiff extension, or an additional text file with the same name and extension .tfw.

2.2.1 Additional layers

Two additional raster layers can be added to an existing project: the DSM (Digital Surface Model) and the land use.

DSM

In PixScape, the DSM corresponds to the height of the object (building, trees, ...) on the ground. The addition of the DTM and the DSM makes it possible to obtain the total elevation of the elements of the landscape. The unit of heights must be expressed in meters. To load a DSM, simply select a raster file in Tiff or AsciiGrid format from the Data / Load DSM menu. After loading, the DSM is stored in the project directory

Land use

The land use layer is used by several landscape configuration metrics (S, IJI, CONTAG, etc.). The codes for land use categories must be between 0 and 255. To load a land use layer, simply select a raster file in Tiff or AsciiGrid format from the Data / Load Land Use menu. After loading, the layer is stored in the project directory.

The colors associated with the land use categories are set randomly, but can be changed in the style of the layer. Color changes are automatically saved in the project's XML file to be kept from one opening to the next.

These two layers must have exactly the same geometry as the DTM: the same spatial extent and the same resolution.

2.2.2 Multiscale

It is possible to add data at coarser resolutions to speed up visibility calculations. This data can be generated directly in PixScape from the Data / Multi scale / Generate menu or imported from the Data / Multi scale / Add scale menu.

Generate multiscale data

Generation of a multiscale database can be done directly in PixScape. User must give the required resolutions separated by commas. By default the software proposes 4 resolutions separated by a factor of 3. If the resolution of the initial data is 1 meters, the software will propose the resolutions 3m, 9m, 27m and 81m.

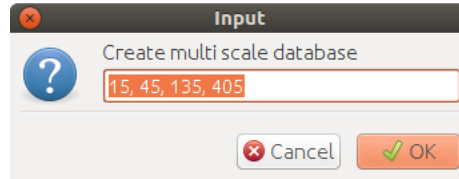


Figure 2.2: Multiscale database generation

It is recommended to use resolutions that follow a geometric sequence of a factor 2, 3 or 4. PixScape will create a new DTM for each resolution, as well as the DSM and land use if these layers are present in the project. For the DTM and the DSM the pixels are aggregated by the arithmetic mean, for the land use, the pixels are aggregated by the mode textit ie. the dominant land use category. At the end of the processing, all the new layers are saved in Tiff format in the project directory and are added in the layer tree of the main window.

Import data

The multiscale database can be imported for each scale. At least DTM must be provided, and DSM plus landuse should be provided if these layers are already presents for the initial scale. As with the creation of the project, the resolution of the altitudes of the DTM can be indicated.

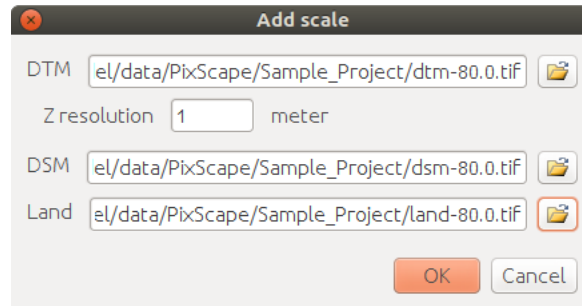


Figure 2.3: Import layers for one scale

The resolution of the DSM heights must be in meters and the land use codes must be between 0 and 255. For land use, codes may vary from one scale to another to differentiate objects that do not have the same meaning according to the resolution, for example a tree and a forest, a building and a village. The three layers must have exactly the same geometry: the same spatial extent and the same resolution. Finally, the spatial extent must cover at least the spatial extent of the initial data.

When the project contains multiscale data, it can be used in visibility calculations by activating the multiscale setting in the Options window (see [Options](#)).

2.3 Visibility calculation

Two visibility calculation methods are available from the Visibility menu: the viewshed and the tangential view. For each method, the user can view the result for a given observation point by clicking on the map.

2.3.1 Viewshed

The viewshed corresponds to the surface on the plane (x, y) containing the set of points visible from the observation point O .

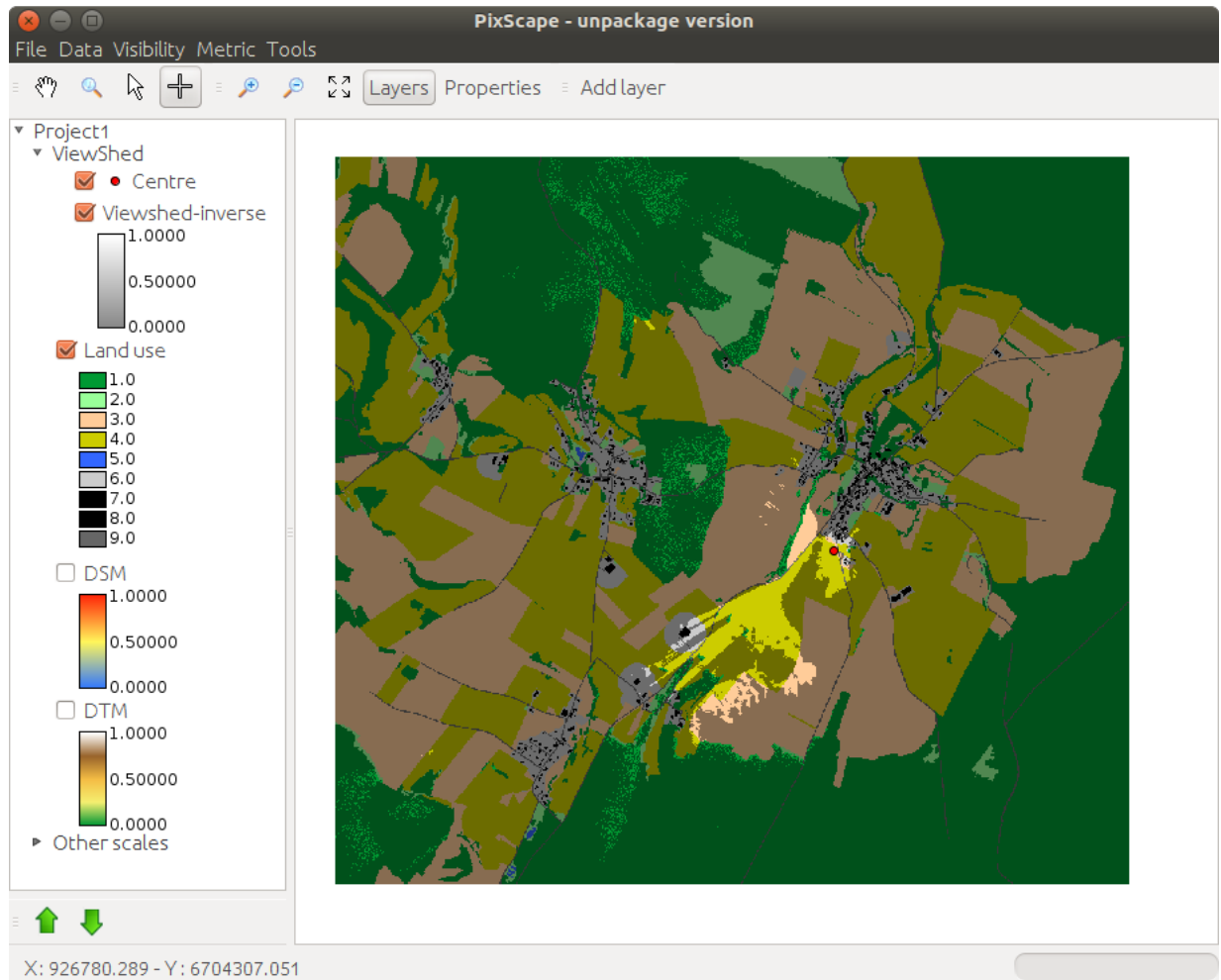


Figure 2.4: Resulting viewshed from the red point. The viewshed is displayed by transparency.

The calculation of a viewshed can be accessed from the Visibility / Viewshed menu. A window appears showing all the parameters that are relevant for the calculation of a viewshed (figure 2.5).

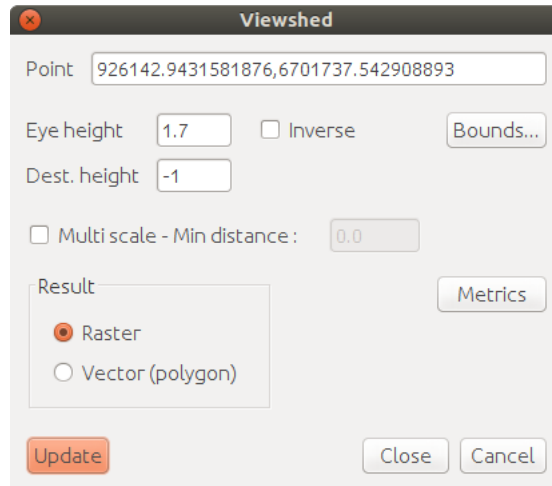


Figure 2.5: Viewshed settings window

Parameters

- Point: coordinates of the observation or observed point O. It can be entered manually or given by clicking on the map.
- Eye height (h1): height in meter of the observer eye
- Dest. height (h2): height in meters of observed point(s). If set to -1, the height of the DSM is used. If the project does not contain a DSM, the height is zero.
- Inverse: unchecked by default, point O is the observation point *ie.* the eye of the observer (figure 2.6). The viewshed corresponds to all the points visible from O. Conversely, if the box is checked, the point O becomes the observed point (figure 2.7). The viewshed then corresponds to the set of points from which the observer sees the point O.
- Bounds: this window allows you to restrict the eyesight of the observer in the 3 dimensions (see [Eyesight bounds](#))
- Multiscale: enables or disables multiscale calculation. If enabled, the minimal distance in meter before the change of scale must be entered.

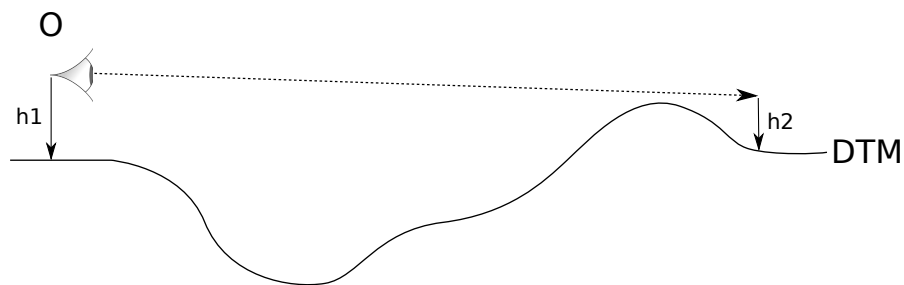


Figure 2.6: Default mode: O is the observation point, h1 is the height of the eye and h2 is the destination height or the DSM height if Dest. height = -1

Dest. height (h2) parameter is often used with inverse mode to analyze the impact of a construction (pylon, wind turbine, etc). The resulting viewshed then corresponds to the set of points from which the new construction of a height h2 will be visible (figure 2.7).

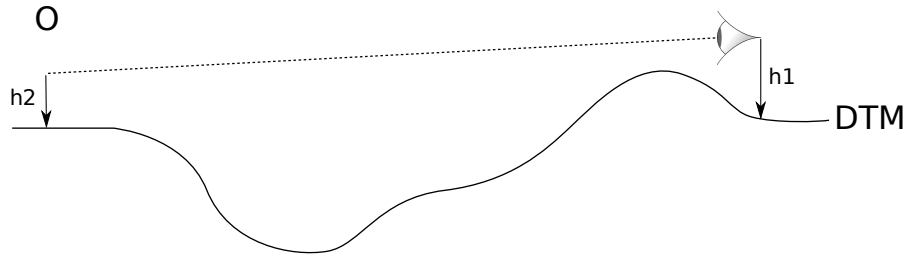


Figure 2.7: Inverse mode: O is the observed point, h1 is the height of the eye and h2 is the destination height (defined by Dest. height or the DSM height if Dest. height = -1)

Results

The viewshed can be displayed on the map in raster or vector form according to the choice made in the Result panel. In raster, the bright areas correspond to the visible areas. Conversely, in vectorial, the dark areas correspond to the visible areas. In both cases, the result layer can be exported by right clicking on the layer.

Metrics can be selected from the Metrics button to be calculated on the current viewshed. For more information on metrics, see the section [Metrics description](#).

Metric	Value
A - Area [0.0, Infinity]	[665100.0]
P - Perimeter	[176450.0]
C - Compactness	[61.03420004151587]
S - Shannon land use [0.0, Infinity]	[0.5558536324047149]
FD - Fractal Dimension	[1.434925916843142]
IJI - Interspersion and juxtaposition	[58.367509503540745]
CONTAG - Contagion	[68.20850430001212]
DIST - Distances distribution	[26604.0, 2.9335811061260957E7, 11...

Add... Remove

Figure 2.8: Metrics calculated from the current viewshed

After modifying a parameter, simply click on the Update button to update the viewshed on the map as well as the metrics if they are displayed.

2.3.2 Tangential view

The calculation of a tangential view can be accessed from the Visibility / Tangential View menu. A window appears showing the settings for the calculation of the view as well as for displaying the result (figure 2.9).

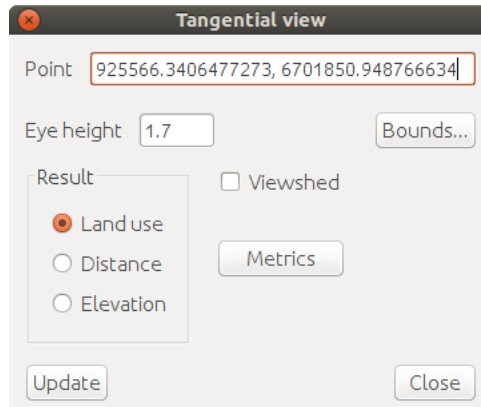


Figure 2.9: Tangential view settings

Parameters

- Point: coordinates of the observation or observed point O. It can be entered manually or given by clicking on the map.
- Eye height (hl): height in meter of the observer eye
- Bounds: this window allows you to restrict the eyesight of the observer in the 3 dimensions (see [Eyesight bounds](#))
- Viewshed: if checked, the viewshed is displayed on the map.
- Result: choice of the result displayed (see [Results](#)).

The angular resolution parameter and the use of the multiscale database cannot be set in this window but only on the Options window (see [Options](#)).

Results

The result is displayed in a new window representing the tangential view (figure 2.10). The coordinates of this image are in degrees from -180° to $+180^\circ$ horizontally and -90° to $+90^\circ$ vertically, the north being at 0° in the middle of the image.

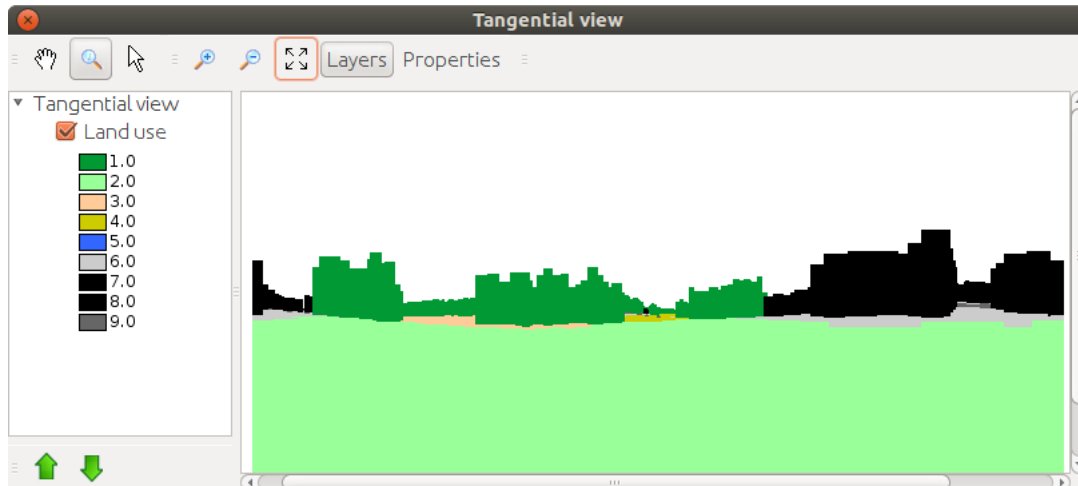


Figure 2.10: Tangential view. The colors represent the land use.

By default, colors represent the land use categories (figure 2.10). By selecting Distance in the result panel, the tangential view displays the distances to the observation point (figure 2.11).

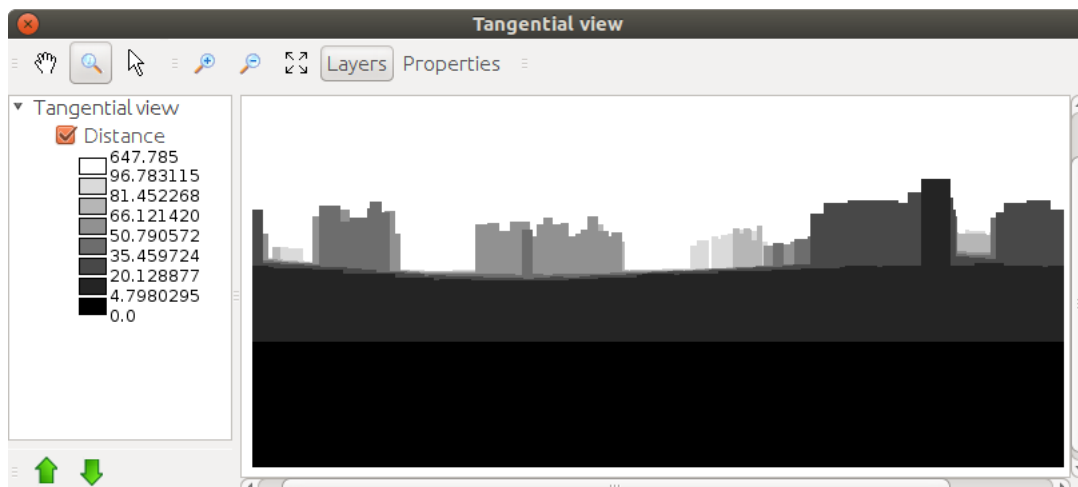


Figure 2.11: Tangential view. The colors represent the distance to the observation point.

Metrics can be selected from the Metrics button to be calculated on the current view. For more information on metrics, see the section [Metrics description](#).

After modifying a parameter, simply click on the Update button to update the view as well as the metrics if they are displayed.

2.3.3 Multi-viewshed

The Visibility / Multi Viewshed menu allows you to calculate several viewsheds from a shapefile containing observation points.

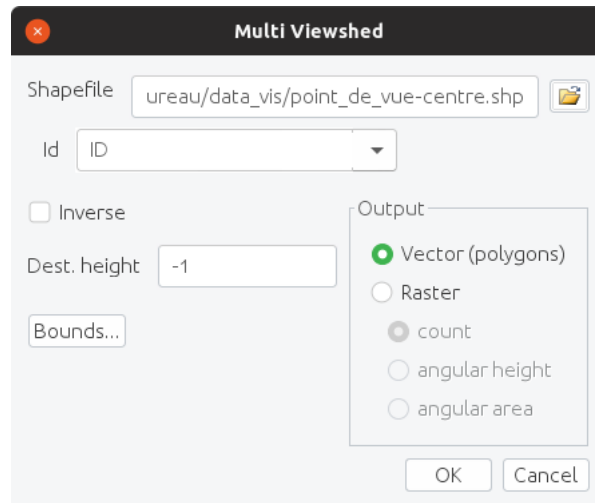


Figure 2.12: Multi-viewshed settings

Parameters

- Shapefile : shapefile containing observation points.
- Id : shapefile attribute used as identifier. This parameter is useful only for vectorial result.
- Inverse: unchecked by default, the shapefile points represent the observation points *ie.* the eye of the observer. Conversely, if the box is checked, the shapefile points becomes the observed points. For details see the viewshed [Parameters](#).
- Dest. height (h2): height in meters of observed point(s). If set to -1, the height of the DSM is used. If the project does not contain a DSM, the height is zero.
- Bounds: this window allows you to restrict the eyesight of the observer in the 3 dimensions for all observation points (see [Eyesight bounds](#)). If the shapefile contains the attributes (dmin, dmax, zmin, zmax, orien et amp), they will be used in place of these settings.
- Output : format of the result layer : vector or raster.
- Raster : for raster output, the result may be the number of visible points (count), the total visible area (angular area) or the total visible height(angular height).

The eye height (h1) parameter and the use of the multiscale database cannot be set in this window but only on the Options window (see [Options](#)).

Result

The resulting viewsheds can be displayed on the map in raster or vector format according to the selected option on Output panel. In both cases, the result layer is saved in the project directory.

In vector, the layer contains a multi-polygon for each viewshed (figure [2.13](#)). Each multi-polygon has an Id attribute allowing to link the shape with the corresponding observation point.

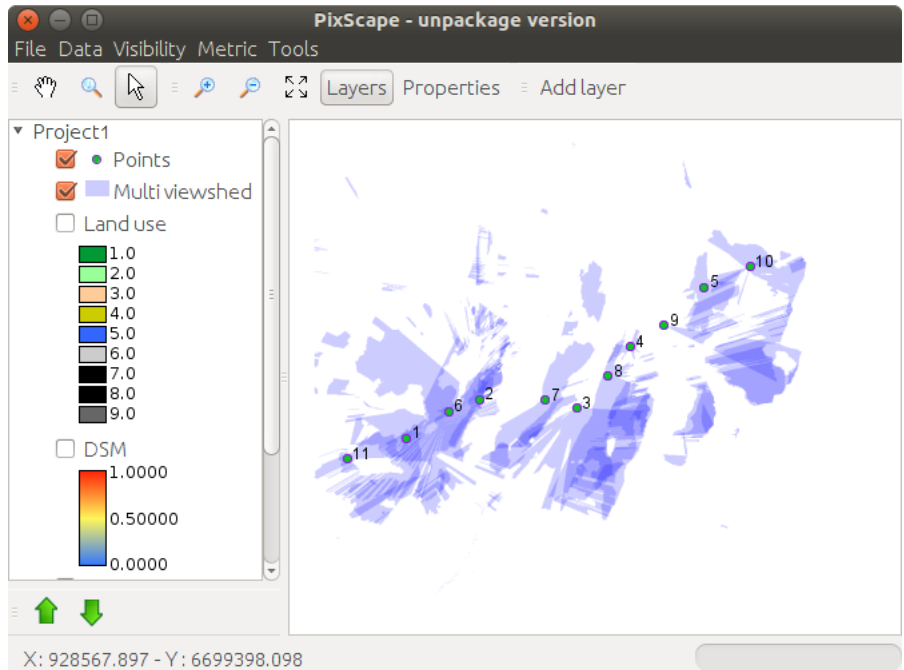


Figure 2.13: Multi-viewshed in vector format

In raster, the value of the pixel corresponds to the number of times the pixel is contained in a viewshed *ie.* the number of observation points seeing the pixel or, in inverse mode, the number of observation points seen from the pixel (figure 2.14). If "angular area" or "angular height" is selected, the pixel value is the total area or height in degree (figure 2.15).

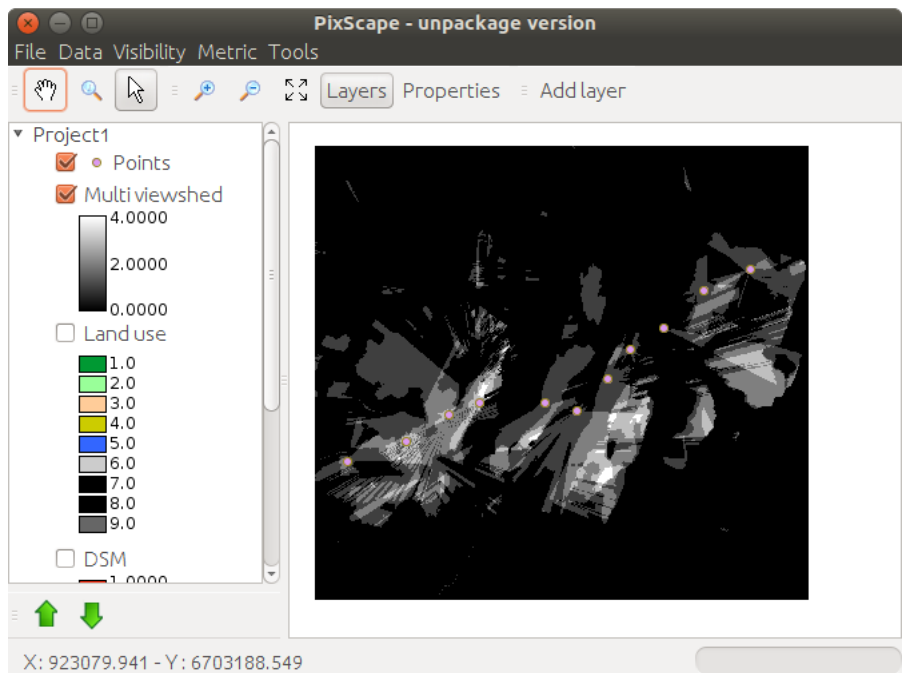


Figure 2.14: Multi-viewshed in raster format - Dest. height (h2) = -1 - count option

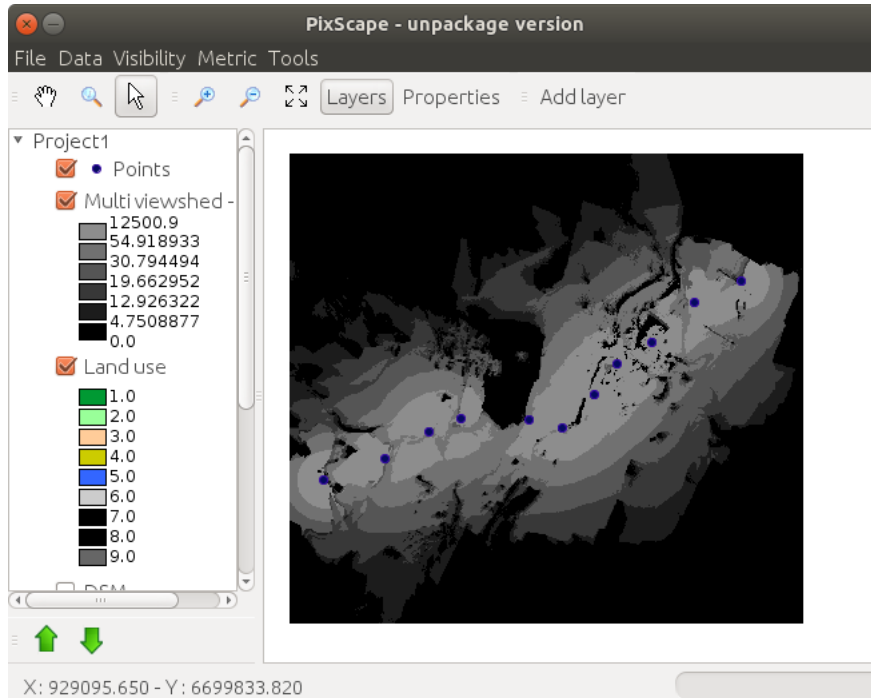


Figure 2.15: Multi-viewshed in raster format - Dest. height (h2) = 200 - angular area option

2.3.4 Eyesight bounds

For all visibility calculations into PixScape, the eyesight can be restricted in all 3 dimensions. Details are described in the section [Eyesight bounds](#).

These bounds can be set from the Bounds window (figure 2.16). When the observation points come from a shapefile, the setting of the eyesight can be differentiated for each point by adding the 6 parameters (dmin, dmax, zmin, zmax, orien, amp) in the shapefile attributes table. The "Set point attributes" menu allows you to automatically add them to an existing shapefile (see [Add eyesight attributes](#)).

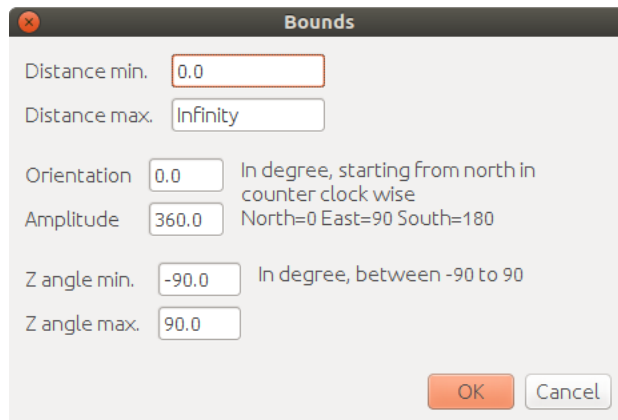


Figure 2.16: Settings of the eyesight. By default the eyesight bounds are unlimited.

2.4 Visibility metric calculation

PixScape can aggregate the result of a viewshed or tangential view by several indicators, or metrics. The available metrics are detailed in the chapter [Metrics description](#).

The Visibility menu already allows you to calculate several metrics on the current view. But this "manual" operation is only possible for a few observation points.

The Metric menu allows you to launch a visibility calculation (planimetric or tangential) and the calculation of the metrics for a set of observation points in a single execution.

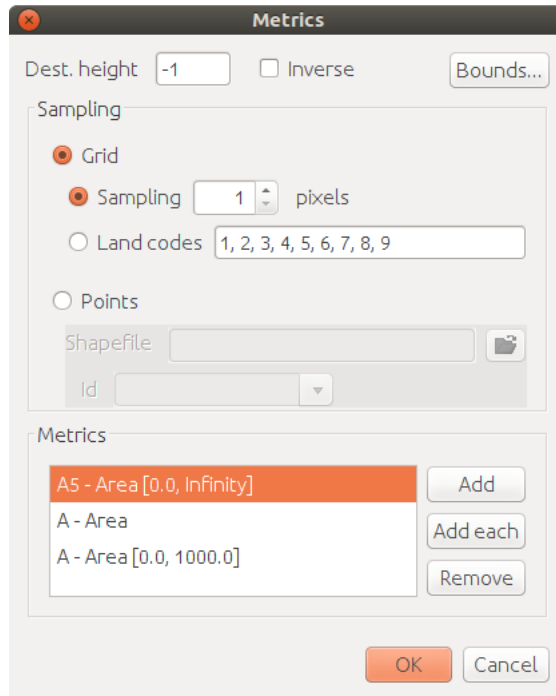


Figure 2.17: Settings for metrics calculation

2.4.1 View parameters

View-specific parameters are displayed at the top of the window (figure 2.17) and depend on the type of view selected in the Metric menu: planimetric (for viewshed) or tangential.

- Inverse (planimetric only): unchecked by default, the shapefile points represent the observation points *ie.* the eye of the observer. Conversely, if the box is checked, the shapefile points becomes the observed points. For details see the viewshed [Parameters](#).
- Dest. height (h2) (planimetric only): height in meters of observed point(s). If set to -1, the height of the DSM is used. If the project does not contain a DSM, the height is zero.
- Bounds: this window allows you to restrict the eyesight of the observer in the 3 dimensions for all observation points (see [Eyesight bounds](#)). If the selected sampling is Points and the shapefile contains the attributes (dmin, dmax, zmin, zmax, orien et amp), they will be used in place of these settings.

The other parameters for the view (*eg.* eye height, angular resolution, ...) must be set before in the Options window (see [Options](#)).

2.4.2 Sampling

Observation points can be given in raster (Grid option) or vector (Points option). The format of the result (raster or vector) is related to the type chosen.

Grid sampling (raster)

With grid option, the sampling of the observation points is done from the grid of the DTM at the finest resolution. Two options are possible: regular sampling on the grid or land use category.

By default, the sampling is regular and set at 1 pixel, which corresponds to an exhaustive run of the set of pixels of the image *ie.* a visibility calculation will be executed from each pixel. If we increase the sampling to 3, only one pixel out of 3 will be used in each dimension, so we will compute a view only for 1 pixel out of 9 of the grid (figure 2.18).

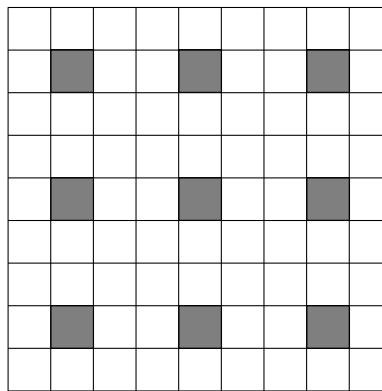


Figure 2.18: Grid sampling every 3 pixels on each dimension. 9 views will be calculated from the 9 shaded pixels.

If the sampling is done by land use category, the views will be calculated only from the pixels belonging to one of the listed categories. Land use codes must be separated by commas.

The result will be one or more raster layers. It will contain one layer for each metric and for each distance range defined in the metric setting.

Points sampling (vector)

Instead of using the raster grid sampling, you can provide a set of points from a shapefile.

The result will be in the form of a point layer corresponding to the input shapefile. The result of the metrics will be stored in the attribute table of the shapefile.

2.4.3 Metric parameters

For each observation point, a set of metrics can be computed. The Metrics panel allows you to select them and to set distances and landuse categories if needed. The same metric can be added several times with different settings.

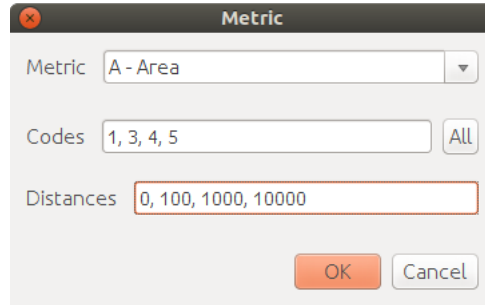


Figure 2.19: Parameters of metric A

Figure 2.19 shows the settings of the metric A. In this example, metric A will only be calculated on the landuse classes 1, 3, 4 and 5 for 3 distance intervals [0-100[, [100-1000[and [1000-10000[. The result will contain 3 values for each observation point, one for each distance interval.

"Add each" button adds the selected metric several times: one for each land-use category.

Several landuse categories can be grouped in one class with the dash character "-" (figure 2.20). This makes it possible to keep a detailed land-use and at the same time carry out some analysis on more general classes.

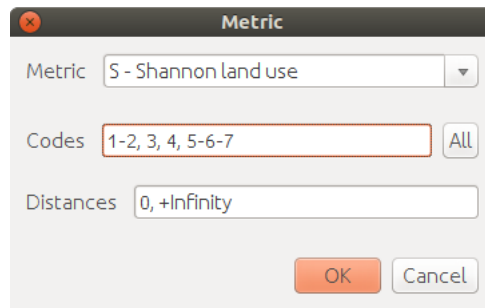


Figure 2.20: Parameters of metric S. Land-use category are grouped in 4 classes.

The grouping of land-use categories is not useful for metric A. Indeed, this metric does not differentiate the categories in each class. For example, 1,3,4,5 and 1-3,4-5 will give the same result for the metric A.

2.4.4 Results

The result of the calculation of metrics is displayed on the map by one or more layers in raster or vector format depending on the input format used for sampling. Each layer can be exported by right clicking on the layer name and the "Export ..." menu.

Raster

For raster, the result is a set of raster layers, one for each metric and each distance interval. Each layer can be exported in Tiff or AsciiGrid format.

For the settings of the figure 2.19, PixScope adds 3 raster layers (figure 2.21) for the 3 distance intervals.

If the sampling is greater than 1, the pixel size is increased proportionately as the sampling factor.

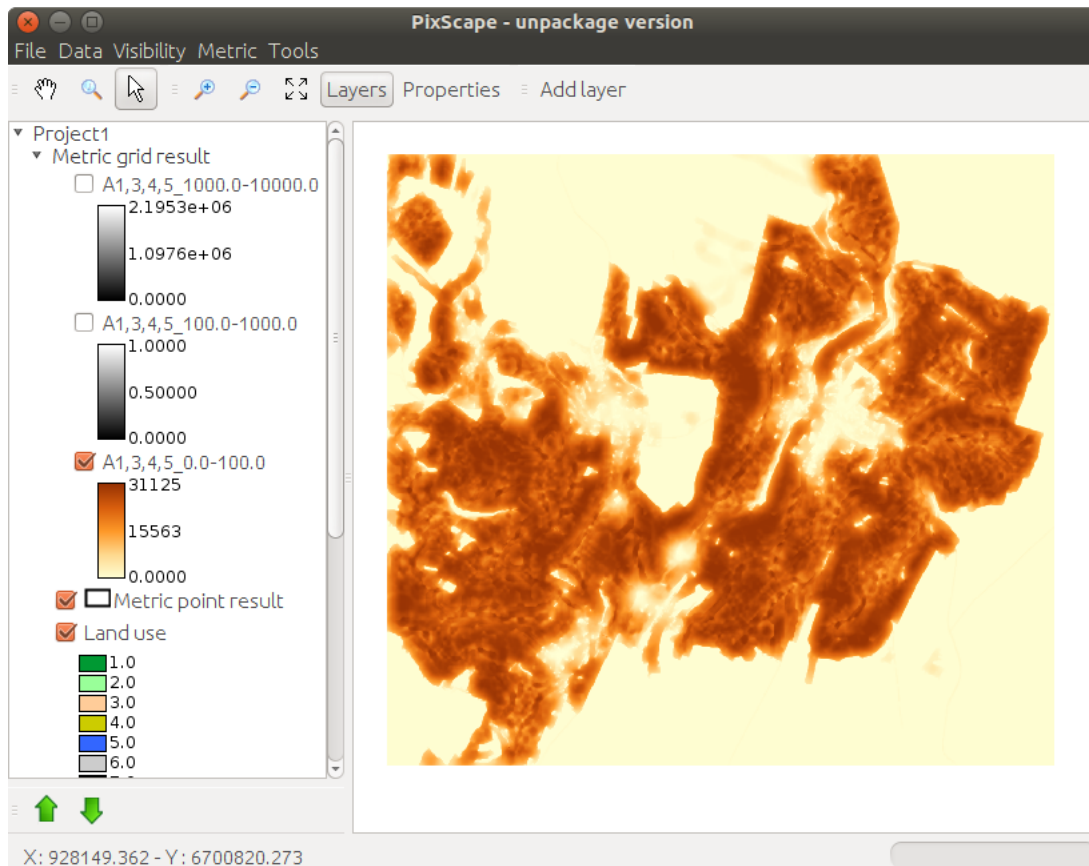


Figure 2.21: Metric result with grid sampling.

Vector

In vector, the result is a point layer containing the metrics values in the attribute table (figure 2.23). By default the first metric is displayed with proportional circle (figure 2.22). To display another metric, simply change the attribute selected in the "Circle" tab of the layer style window.

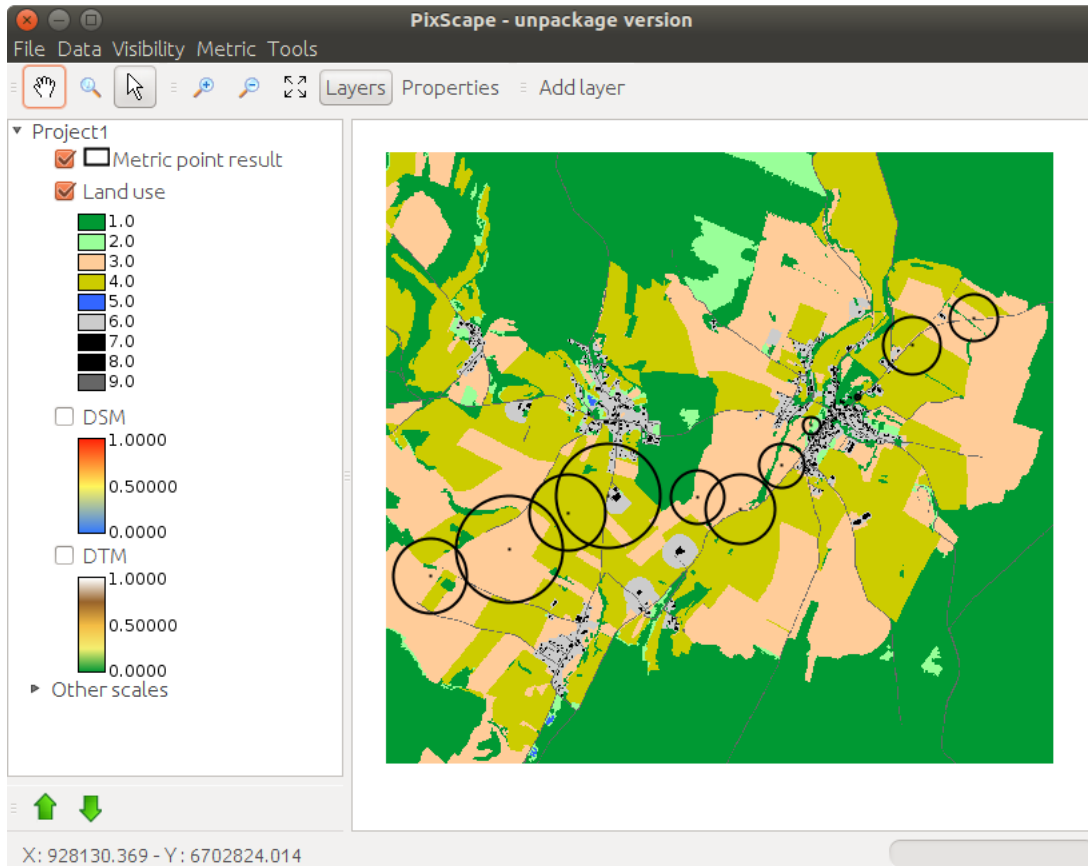


Figure 2.22: Metric result from a point layer. The first metric is displayed by circles proportional to the value of the metric.

The layer "Metric point result" can be exported in shapefile or text format by right clicking on the layers name and the menu "Export ...".

Metric point result									
id	dmin	dmax	orien	amp	zmin	zmax	A_0.0-Infinity		
1	0	∞	0	360	-90	90	1,376,025		
2	0	∞	0	360	-90	90	1,309,200		
3	0	∞	0	360	-90	90	579,475		
4	0	∞	0	360	-90	90	34,600		
5	0	∞	0	360	-90	90	403,550		
6	0	∞	0	360	-90	90	688,425		
7	0	∞	0	360	-90	90	328,175		
8	0	∞	0	360	-90	90	237,000		
9	0	∞	0	360	-90	90	225		
10	0	∞	0	360	-90	90	270,375		
11	0	∞	0	360	-90	90	662,975		

Figure 2.23: Attribute layer of the result layer.

The attribute table of the result layer is accessible by right clicking on the layer name and the "Attribute table" menu. It contains, in addition to the metrics, the 6 parameters used for limiting the eyesight, allowing to keep these parameters with the results.

2.5 Tools and options

2.5.1 Add eyesight attributes

For all visibility calculations, the eyesight can be restricted by the Bounds window (figure 2.16). This solution is useful if each observation point has the same limitations of the eyesight. The Tools / Set point attributes menu is useful for specifying different restrictions for each observation point. This function adds the 6 attributes to a shapefile with the default values defined by the Bounds window. You can then with your favorite GIS modify the restriction values for each observation point. The output shapefile can be used for calculation of metrics (2.4) or multi viewshed (2.3.3).

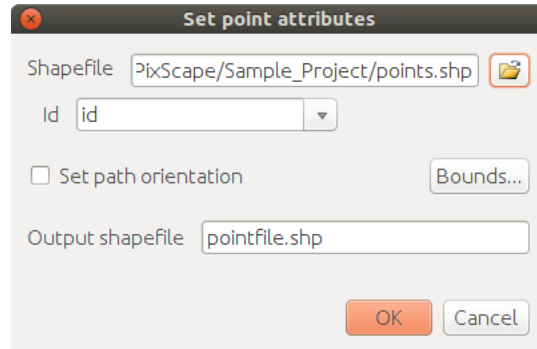


Figure 2.24: Add eyesight bound attributes to a shapefile

Eyesight along a path

The check box "Set path orientation" calculates the orientation of each point depending on the next one. When this option is checked, the orientation parameter of the "Bounds" window is ignored and each point may have a different orientation.

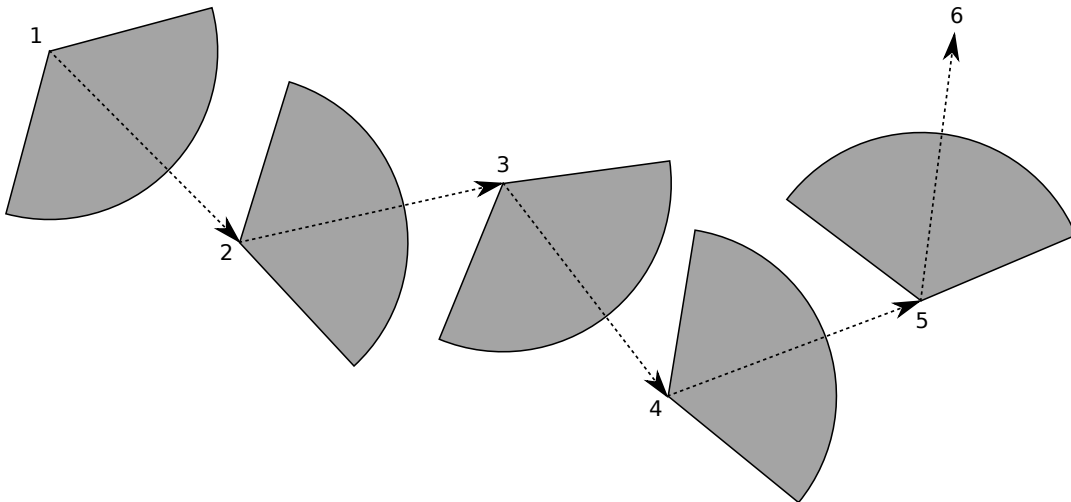


Figure 2.25: Eyesights along a path. The orientation is calculated by the next point. The amplitude is constant.

The points order is defined by "id" attribute which is sorted with natural ordering. This option is useful only with an amplitude less than 360°.

2.5.2 Options

The general project options are available from the Tools / Options menu. These options are saved automatically in the xml file of the project. They are used for all the visibility calculations except when an option is also present in the settings window of a processing. For example, the height of the eye can be changed directly in the settings window of a planimetric or tangential view, but it is not available for calculating metrics. In the latter case, the height of the observer's eye will be set from the general options window.

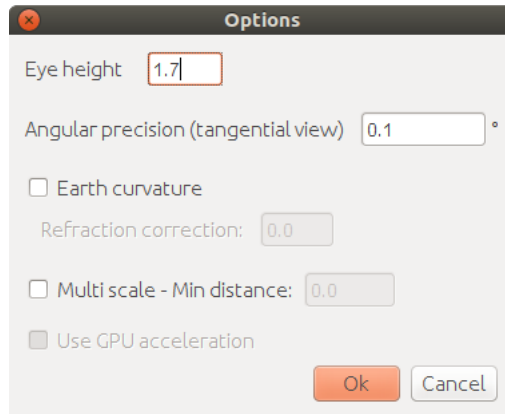


Figure 2.26: General options

Height of the observer's eye

Eye height parameter (h_1) sets the height of the observer from the DEM. The height is in meters. For more details, see [Description of visibility calculations](#) and the figure 4.2.

Angular precision

Angular precision parameter (α) sets the resolution in degrees of the tangential view. It is set by default to 0.1° , which corresponds to 3600 rays if the amplitude of the view is 360° . For more details, see [Tangential view](#) and the figure 4.5.

Earth curvature

The visibility calculations in PixScope can take into account the curvature of the Earth, which can play a role in the case of a view greater than several tens of kilometers. The principle of computation is simple: the more the distance to the observer increases, the more the height of the visible objects diminish. The height taking into account the curvature of the Earth z'_i of a point i is calculated as follows:

$$z'_i = z_i - \frac{d_i^2}{D}$$

with z_i the elevation of the DTM at point i , d_i the distance between the point i and the observer and D the Earth diameter (12 740 000 meters).

The Earth curvature may be compensated by the light refraction in the atmosphere. In PixScope, when the Earth curvature is taken into account, the refraction parameter can be specified. It is set to 0.13 by default. The calculation of the height becomes

$$z'_i = z_i - (1 - \rho) \frac{d_i^2}{D}$$

with ρ the refraction coefficient between 0 and 1. If $\rho = 0$, we return to the preceding formula without refraction.

Multiscale

When the project contains data with several resolutions, the visibility calculations can be done in multiscale by activating this option. The minimum distance parameter must be defined. It ensures a minimum distance in meters before changing resolution.

For more details on the multiscale calculations, see [Multiscale](#).

GPU acceleration

This parameter is used to perform visibility calculations on the graphics card processor (GPU) instead of the central processor. Runtime gains can be significant for Nvidia Tesla graphics cards, designed for high-performance computing (see [Performance tuning](#)). On the other hand, not all metrics are optimized for the GPU. Metrics A and S are the only ones to be optimized. If you use other metrics, the time savings will be minimal or non-existent.

The parameter is disabled if PixScape can not use GPU acceleration. The causes can be multiple: no NVidia graphics card, no CUDA support, not the right version of CUDA (6.5).

GPU acceleration cannot be used for multiscale calculations. If multiscale option is enabled, GPU acceleration will not be used even if it is enabled.

Chapter 3

Command line interface (CLI)

Once a project has been created from the GUI, it is possible to use PixScape on the command line. This mode is useful for running the software on a remote computer without a graphical interface, automatically launching several processes, or submitting executions on a cluster.

3.1 Starting

3.1.1 Launch PixScape in CLI mode

First you have to open a terminal window. Then, go to the directory of the Graphab program with *cd* command. Finally, type the following command to display the PixScape help screen :

```
java -jar pixscape.jar --help
```

Result

Usage :

```
java -jar pixscape.jar --metrics
```

...

...

You're ready to use PixScape in CLI mode.

3.1.2 Syntax

Definition

Commands always start with a double dash: `--project`, `--metrics`, ...

Global options start with only one dash: `-proc`, `-bounds`, ...

Parameters does not have dash: `dmin`, `inverse`, ...

Character separator

Blank spaces are used to separate commands and parameters. You cannot have a name containing blank spaces.

Help screen syntax

In the help screen, elements enclosed in brackets are optional. Therefore, elements not in brackets are mandatory. The character | separates the possible options.

3.2 Commands

3.2.1 -help : show help screen

Command:

```
java -jar pixscape-1.2.jar --help
```

Result:

Usage :

```
java -jar pixscape.jar --metrics
java -jar pixscape.jar [-mpi | -proc n | -cuda n]
--project project_file.xml
[--landmod zone=filezones.shp id=fieldname code=fieldname dsm=file.tif [selid=id1,...,idn]]
[-zeye val] [-zdest val] [-resdir path]
[-bounds [dmin=val] [dmax=val] [orien=val] [amp=val] [zmin=val] [zmax=val]]
[-sampling n=val | land=code1,..,coden | points=pointfile.shp id=fieldname]
[-multi dmin=val | -mono]
[-earth flat|curved [refrac=val]]
commands
```

Commands list :

```
--viewshed [inverse] x y [resfile=raster.tif]
--viewtan [prec=deg] x y [resname=name]
--multiviewshed format=vector|raster [inverse] [degree=area|height] [resname=name]
--planmetric [inverse] metric1[[code1,...,coden]][_d1,...,dm] ... metricn[[code1,...,coden]][_d1,...,dm]
--tanmetric [prec=deg] metric1[[code1,...,coden]][_d1,...,dm] ... metricn[[code1,...,coden]][_d1,...,dm]
--toobject [degree=area|height] [agreg=eye|object] objects=pointfile.shp id=fieldname [resname=name]
```

3.2.2 -metrics : show metrics

This command lists all available metrics with their short name and full name.

Commande :

```
java -jar pixscape-1.2.jar --metrics
```

Résultat :

```
==== Metrics ====
A - Area
S - Shannon land use
P - Perimeter
C - Compactness
FD - Fractal Dimension
AG - Aggregation index
CONTAG - Contagion
ED - Edge Density
PD - Patch Density
```

PMS - Patch Mean Size
DIST - Distances distribution
SL - Skyline ratio
SD - Shannon depth
DL - Depth line

3.2.3 `-project` : load project

This command sets the path to the project xml file.

```
java -jar pixscape-1.2.jar --project path2myproject/myproject.xml
```

This command loads the project *myproject* contained in the *path2myproject* folder.

The `--project` command can be used only once and must be the first command. All of the following commands below require a loaded project

3.2.4 `-landmod` : land use changes

```
--landmod zone=filezones.shp id=fieldname code=fieldname dsm=file.tif [selid=id1,...,idn]
```

Required parameters

- `zone=filezones.shp` : polygon shapefile containing land use changes.
- `id=fieldname` : field name of the shapefile used for identifying polygons. If values are not unique, polygons with the same identifier will be applied in a single change.
- `code=fieldname` : field name of the shapefile storing the new land use category of the polygon.
- `dsm=file.tif` : raster storing the new values of the DSM used for each land use change. The geometry of the raster must be exactly the same as the project DEM.

Optional parameter

- `selid=id1,...,idn` : list of polygon identifiers to process. If this parameter is not set, all polygons of the shapefile are processed.

Description

This command must be placed before the visibility calculation commands. It will duplicate the project for each polygon of the shapefile and change the landuse covered by the polygon and the DSM. The commands following this one, will be executed on each modified project.

The newly created projects will be named by the identifier of the polygon(s) and stored in the project directory.

3.2.5 `-viewshed` : planimetric view or viewshed

```
--viewshed [inverse] x y [resfile=raster.tif]
```

Required parameters

- **x** : abscissa of the observation point in the coordinate system of the DTM
- **y** : ordinate of the observation point in the coordinate system of the DTM

Optional parameters

- **inverse** : reverse mode, the point O becomes the observed point
- **resfile=raster.tif** : allow to specify another name for the file storing the result

Description

The `--viewshed` command allows to calculate the viewshed from an observation point. The result is stored in a raster file of the same size as the DTM in Tiff format. In this raster, all pixels set to 1 represents the viewshed, other pixels are set to 0.

Examples

The example below calculates the viewshed for an observer located at the point (925560, 6702495). The result is stored in the projet folder in a file named : `viewshed-925560,6702495.tif`.

```
--viewshed 925560 6702495
```

This second example calculates the viewshed in reverse mode (ie. the given coordinates represent the observed point) and stored the result in the file `bassin1.tif`.

```
--viewshed inverse 925560 6702495 resfile=bassin1.tif
```

3.2.6 `--viewtan` : tangential view

```
--viewtan [prec=deg] x y [resname=name]
```

Required parameters

- **x** : abscissa of the observation point in the coordinate system of the DTM
- **y** : ordinate of the observation point in the coordinate system of the DTM

Optional parameters

- **prec=deg** : angular precision in degrees. If this parameter is not set, the angular precision stored in the project will be used.
- **resname=name** : allow to specify another name for the files storing the result

Description

The `--viewtan` command allows to calculate the tangential view from an observation point. Results are stored in 2 or 3 raster files in Tiff format depending on whether the project contains a land use layer.

- `elev` : elevation of the DTM
- `dist` : 2D distance from the observation point
- `land` : land use, if present in the project

The rasters size depends on the angular precision. For a precision of 0.1° and an amplitude of 360° , the rasters size will be 3600x1800 pixels. Their coordinate system unit is set in degrees.

Examples

The example below calculates the tangential view for an observer located at the point (925560, 6702495). The result is stored in the projet folder in a files named: `viewtan-925560,6702495-elev.tif`, `viewtan-925560,6702495-dist.tif` and `viewtan-925560,6702495-land.tif` if the project contains a land use layer.

```
--viewtan 925560 6702495
```

The second example calculates the tangential view with an angular precision of 0.05° and stored the result in the files `view1-elev.tif`, `view1-dist.tif` and `view1-land.tif` if the project contains a land use layer.

```
--viewtan prec=0.05 925560 6702495 resname=view1
```

3.2.7 `--multiviewshed` : multiple viewshed

```
--multiviewshed format=vector|raster [inverse] [degree=area|height] [resname=name]
```

Required parameter

- `format=vector|raster` : result format (shapefile or tiff)

Optional parameters

- `inverse` : reverse mode, the point O becomes the observed point
- `degree` : for raster format, cumulate visible area or height in degree instead of the number of visible points
- `resname=name` : allow to specify another name for the file storing the result

Description

The `--multiviewshed` command allows to calculate several viewsheds from a set of observation points. The observation points must be given by the `-sampling` option. The result is stored in a raster file or a shapefile depending on the selected format.

In raster, the image has the same size as the DTM in Tiff format. In the raster, each pixel counts the number of times it is seen from the observation points and in reverse mode the number of points seen from this pixel. If the `degree` option is used, each pixel represents the total visible area or height in degree of this pixel seen

from all observation point. In reverse mode, each pixel contains the total visible area or height from this pixel to all observation points.

In vector, the result is a shapefile storing the viewshed of each observation point.

Example

The example below calculates the viewsheds of the observation points from the shapefile `points.shp` and stores the result in a shapefile in the project folder named: `multiviewshed.shp`.

```
-sampling points=points.shp id=fid --multiviewshed format=vector
```

3.2.8 `--planmetric` : metric in planimetric view

```
--planmetric [inverse] metric1 [[code1,...,coden]] [_d1,...,dm] ...
```

Required parameters

- `metric1[[code1,...,coden]][_d1,...,dm]` : planimetric metric to calculate with its parameters (cf. [Metric parameters](#)).
- ...

Optional parameter

- `inverse` : reverse mode, the observation points become the observed point

Description

The `--planmetric` command allows to calculate a set of metrics in planimetric view. The observation points are defined before with the global option `-sampling`. If this global option is not set, the default sampling is used, corresponding to all the pixels of the DTM ($n = 1$).

Results are stored in several rasters for a grid sampling and in one shapefile for a vector sampling.

Examples

The example below calculates the A metric without parameter. If the sampling is on the grid, the result will be stored in the file `A-.tif`.

```
--planmetric A
```

The example below calculates the A metric in reverse mode for the land use category 1 and 2. If the sampling is on the grid, the result will be stored in the files `A1-.tif` and `A2-.tif`.

```
--planmetric inverse A[1] A[2]
```

3.2.9 `--tanmetric` : metric in tangential view

```
--tanmetric [prec=deg] metric1 [[code1,...,coden]] [_d1,...,dm] ...
```

Required parameters

- `metric1[[code1,...,coden]][_d1,...,dm]` : tangential metric to calculate with its parameters (cf. [Metric parameters](#)).
- ...

Optional parameter

- `prec=deg` : angular precision in degrees

Description

The `--tanmetric` command allows to calculate a set of metrics in tangential view. The observation points are defined before with the global option `-sampling`. If this global option is not set, the default sampling is used, corresponding to all the pixels of the DTM ($n = 1$).

Results are stored in several rasters for a grid sampling and in one shapefile for a vector sampling.

Examples

The example below calculates the A metric without parameter. If the sampling is on the grid, the result will be stored in the file `A-.tif`.

```
--tanmetric A
```

The example below calculates the tangential views with an angular precision of 0.05° . The A metric is calculated twice: one for the land use category 1 and another time for the category 2. If the sampling is on the grid, the result will be stored in the files `A1-.tif` and `A2-.tif`.

```
--tanmetric prec=0.05 A[1] A[2]
```

3.2.10 `--toobject` : point to point visibility

```
--toobject [degree] [agreg=eye|object] objects=pointfile.shp id=fieldname [resname=name]
```

Required parameters

- `objects=pointfile.shp` : shapefile of observed points
- `id=fieldname` : shapefile field containing an identifier

Optional parameters

- `degree` : angular result
- `agreg=eye|object` : aggregate results by observation point (`eye`) or by observed point (`object`)
- `resname=name` : allow to specify another name for the file storing the result

Description

This command calculates the visibility between 2 sets of points. The observation points must be defined by the general option `-sampling points=...`. The observed points are defined by the parameter `object`. The result is stored by default in the file `toobject.csv`.

Without `verb|agreg|` parameter, the result contains the list of points couples (observer, observed) where observer sees the observed point. If the `degree` option is used, the column `SurfView` of the result file contains the angular surface.

With `agreg` parameter, the result is aggregated by observer or observed point and the column `SurfView` contains the sum of the grouped values.

Example

This example calculates the intervisibility of the points given in `points.shp` and stores the result in the file `toobject.csv` in project directory. In this example, observation and observed points are the same.

```
-sampling points=points.shp id=id --toobject objects=points.shp id=id
```

3.3 Metric parameters

The calculation of metrics may be restricted to some land use categories or to some distance intervals. Not all metrics support these settings. To know which metric supports which settings, see [Metrics description](#).

3.3.1 Land use

The calculation of metrics may be restricted to some land use categories by listing the desired categories between brackets. The example below calculates the area of the categories 1,3,4 and 5 in the current view.

```
A[1,3,4,5]
```

For metrics differentiating land use categories (metric S, IJI, and CONTAG), the categories can be grouped into a class by the character `-`. The example below brings together the 6 land use categories in 3 classes: 1-2-3, 4 and 5-6.

```
S[1-2-3,4,5-6]
```

3.3.2 Distances

The calculation of metrics can also be restricted to some distance intervals with the character `_`. The example below will calculate the metric A for 3 distance intervals: 0 to 100m, 100 to 1000m and over 1000m. It is therefore necessary to give 4 distances to obtain 3 intervals.

```
A_0,100,1000,+Infinity
```

For metrics A and S, it is possible to restrict by land use and by distance interval. In this case, the land use categories must be given before the distances:

```
A[1,2,3]_0,100,1000,+Infinity
```

In any case, there is no space in the definition of a metric and its parameters!

3.4 Options

3.4.1 Parallelization : `-mpi`, `-proc`, `-cuda`

`-mpi` | `-proc n` | `-cuda n`

PixScape supports 3 modes of parallelization on the command line: per thread for a computer containing several cores (`-proc`), on graphics card (`-cuda`) or on cluster (`-mpi`).

For more details, see [Parallelism to speed up execution](#) section.

If none of the 3 options is specified, thread parallelization will be used with the number of cores defined in the graphical interface.

3.4.2 View settings : `-zeye`, `-zdest`, `-bounds`, `-earth`

`-zeye`

`-zeye val`

The `-zeye` option sets the height of the observer's eye ($h1$). If it is not set, the height of the eye recorded in the project will be used instead.

`-zdest`

`-zdest val`

The `-zdest` option sets the height of the observed pixels ($h2$). If it is not set, the height of the DSM will be used instead.

`-bounds`

`-bounds [dmin=val] [dmax=val] [orien=val] [amp=val] [zmin=val] [zmax=val]`

The `-bounds` option allows to limit the eyesight in the 3 dimensions by 6 parameters: `dmin`, `dmax`, `zmin`, `zmax`, `orien`, `amp`. For more details, see [Eyesight bounds](#) section.

`-earth`

`-earth flat|curved [refrac=val]`

The `-earth` option is used to define whether the Earth curvature may be taken into account or not. The `flat` parameter disables the curvature calculation and the `curved` parameter enables it. In the latter case, the refraction coefficient of the atmosphere can be given by the parameter `refrac`.

For more details on the curvature calculation, see [Earth curvature](#) section.

If this option is not set, the settings recorded in the project will be used instead.

3.4.3 Sampling : `-sampling`

`-sampling n=val` | `land=code1,...,coden` | `points=pointfile.shp id=fieldname`

The sampling defining the observation points can be given in raster (grid sampling) or in vector form (point sampling). For more details on how sampling works, see [Sampling](#).

If this option is not set and a command needs a sampling, the exhaustive sampling on the grid will be used:
`-sampling n=1`.

Grid sampling (raster)

`-sampling n=val | land=code1,...,coden`

In raster, 2 sampling are available: regular (`n`) or by landuse category (`land`).

The grid sampling can be used only by `--planmetric` and `--tanmetric` commands.

Point sampling (vector)

`-sampling points=pointfile.shp id=fieldname`

Point sampling needs a shapefile of points with an identifier field. The result of `--planmetric` and `--tanmetric` commands is stored in a shapefile named `metrics-pointfile.shp`.

The point sampling can be used by `--planmetric`, `--tanmetric` and `--multiviewshed` commands.

3.4.4 Multiscale : `-multi`, `-mono`

`-multi dmin=val | -mono`

The multiscale calculation can be enabled or disabled with the option `-multi` and `-mono` respectively. The `-multi` option requires the `distmin` parameter to set the minimum distance before changing scale. To use this option, the project must contain multiscale data.

If none of the 2 options is defined, the settings stored in the project will be used instead.

3.4.5 Location results : `-resdir`

`-resdir path`

By default, the results are saved in the project directory. The `-resdir` option allows you to change the default directory. If the given directory does not exist, it will be created automatically.

If a relative path is given, the absolute path will be determined from the current directory at the start of PixScape.

3.5 Examples

In this section, some examples of complete command lines are shown.

3.5.1 Tangential metrics with point sampling

Simple example

```
java -jar pixscape-1.0.jar --project Project1/Project1.xml
    -sampling points=Points_IDF.shp id=Id
    --tanmetric A SL DL
```

The example above loads the project `Project1/Project1.xml` and calculates several tangential metrics (`--tanmetric A SL DL`) from a set of observation points defined in the shapefile `Points_IDF.shp`. The result will be saved in a shapefile named `metrics-Points_IDF.shp`, containing the observation points, and the values of the 3 metrics.

Complete example

```
java -Xmx4g -jar pixscape-1.0.jar -proc 8 --project Project1/Project1.xml -zeye 1.75
    -resdir res_tan -bounds zmin=-20 zmax=30 -sampling points=Points_IDF.shp
    id=Id --tanmetric prec=0.01 A_0,100,1000,+Infinity A[5] A[6] SL DL
```

The following example uses the same basis as the previous one but with more parameters. In this case, the maximum memory allocated to PixScape is set to 4 gigabytes (`-Xmx4g`) and the computation is parallelized on 8 cores (`-proc 8`). The height of the observer's eye is set at 1.75 meters (`-zeye 1.75`). The results will not be saved in the project folder but in a subfolder `res_tan` of the current directory (`-resdir res_tan`). The eyesight is restricted in the vertical angle between -20° and 30° (`-bounds zmin = -20 zmax = 30`). The angular accuracy is set to 0.01° (`prec = 0.01`). Finally, the metric `A` is decomposed on the one hand for 3 distance intervals (`A_0,100,1000,+Infinity`) and for 2 landuse categories (`A[5] A[6]`).

3.5.2 Planimetric metrics with grid sampling

```
java -jar pixscape-1.0.jar -cuda 2 --project Project1/Project1.xml
    -resdir res_plan -sampling land=4
    --planmetric A A[2,3] A[10] S
```

The example above loads the project `Project1/Project1.xml` and calculates several planimetric metrics (`--planmetric A A[2,3] A[10] S`) from the pixels having the landuse category 4 (`-sampling land=4`). The results will be saved in a subfolder of the current folder named `res_plan`. It will contain 4 tiff files: `A-.tif`, `A2,3-.tif`, `A10-.tif` and `S-.tif`. PixScape will use GPU acceleration on 2 graphics cards (`-cuda 2`) for the calculations.

Chapter 4

Description of visibility calculations

A visibility calculation is made from a point of observation denoted O . From this point, a set of rays are calculated to determine the visible or not visible pixels. PixScape implements two methods of visibility calculation: the planimetric view representing the viewshed on the plane (x,y) and the tangential view showing the "real" immersion of an observer.

4.1 Planimetric view

The planimetric view corresponds to the surface on the plane (x,y) containing the set of points visible from the observation point O . The result is called viewshed. To determine the complete viewshed, the rays must cover all the pixels of the image. To cover the entire image, a ray is traced from point O to each pixel of the edge of the image (figure 4.1). As a raster is a discrete surface, the path of the pixels for a ray is not necessarily a straight line but may be a broken line approaching the best of the straight line (right 4.1 figure). The classical Bresenham algorithm is used to approximate the straight line on the discrete space of the raster.

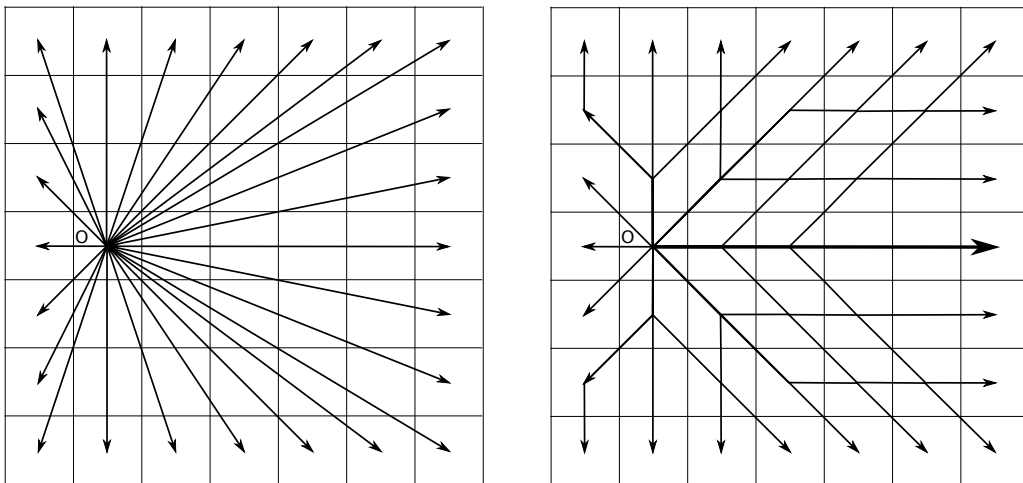


Figure 4.1: Rays traced on a 7x7 pixel image from an observation point located at $(4,2)$. On the left, the theoretical tracing of the rays, on the right the actual path of the rays on the pixels of the image.

For each ray, the pixels are traversed from the closest of the point O to the farthest. At each pixel, the vertical angle between the point O and the center of the pixel is calculated. If the angle is greater than the previous ones, the pixel is visible, otherwise it is not.

The height of the pixels corresponds to the sum of the elevation of the DTM and the height of the DSM if it is present. The height of the observation point O corresponds to the sum of the elevation of the DTM and the parameter h_1 (height of the observer eye). The height of the DSM is never taken into account for the height of the observer (the observer does not climb the trees).

The figure 4.2 represents the calculation of a ray over 5 pixels. The pixel located on the observer's point (pixel 0) is always visible. The next pixel (1) is also visible, but pixel 2 is not visible because the target angle is smaller than for pixel 1, and so on.

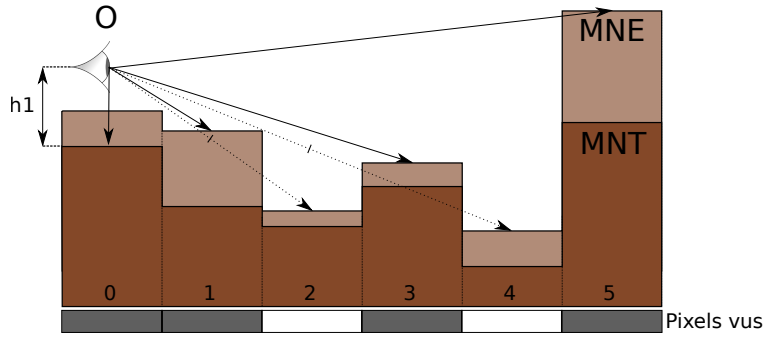


Figure 4.2: Side view of a ray tracing

The result of the calculation of the ray of the figure 4.2 is shown in the following figure (4.3). It corresponds to the pixels of the image which are traversed by the ray and which are visible from the observation point.

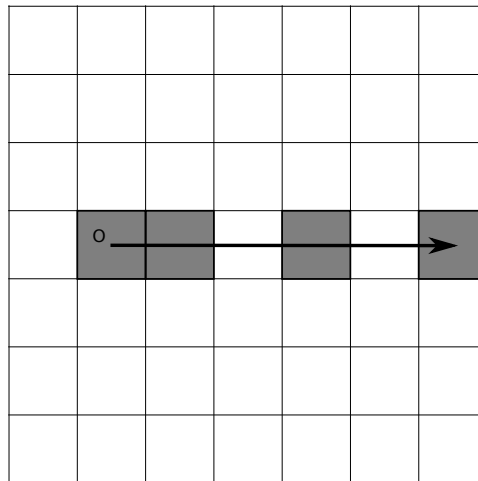


Figure 4.3: Result in planimetric view of the ray calculated on the figure 4.2. The pixels in gray are the visible pixels for this ray.

The complete result corresponds to the combination of the results for each ray. As the pixels of the image can be traversed by different rays, a pixel is marked visible when it is visible by at least one of the rays passing through it.

4.1.1 Reverse planimetric view

The reverse mode inverts the observer and the observed. The viewshed is no longer the area visible from the point O , but the area that sees the point O . The point O is the observed point and the observer is positioned on all points of the image. Figure 4.4 shows the process for a given ray.

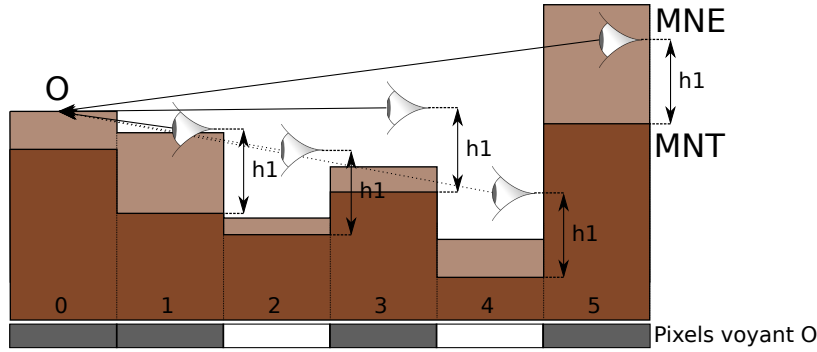


Figure 4.4: Side view of a ray tracing in reverse mode.

The result is identical in this example, but according to the configuration, it may be different in reverse mode.

4.2 Tangential view

For the tangential view, the principle of computation is close but has some differences. First, the scan of the image is not necessarily exhaustive. The number of calculated rays is no longer dependent on the size of the image but on the angular resolution parameter α . By default, this parameter is set to 0.1° . The number of calculated rays is 3600. Figure 4.5 shows an example with an angular accuracy of 30° .

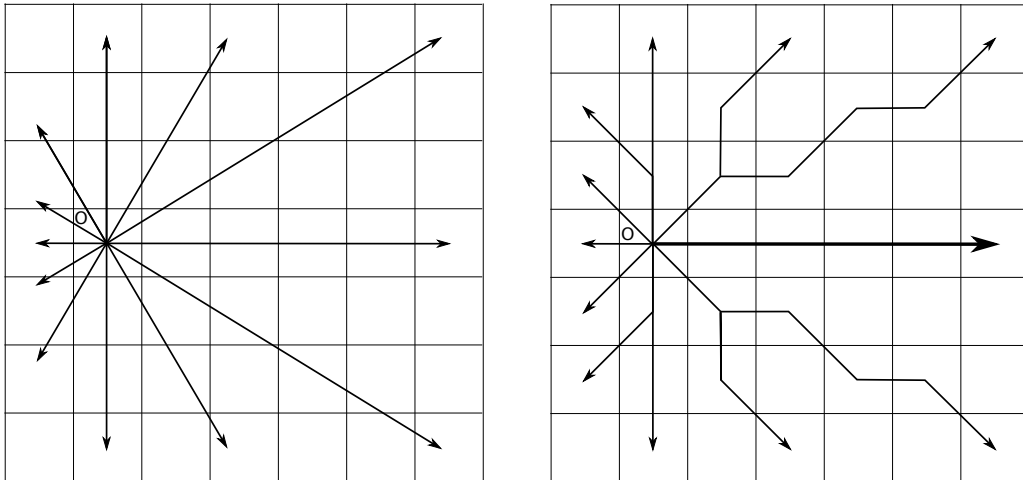


Figure 4.5: Rays traced on a 7x7 pixels image from an observation point located at (4,2). On the left, the theoretical tracing of the rays, on the right the actual path of the rays on the pixels of the image. The angular accuracy (α) here is very coarse: 30° .

In the calculation of each ray, additional processing must be performed to determine the angle height of each visible pixel (figure 4.6). The result is represented by the column to the right of the figure 4.6 for an angular precision (α) of 10° .

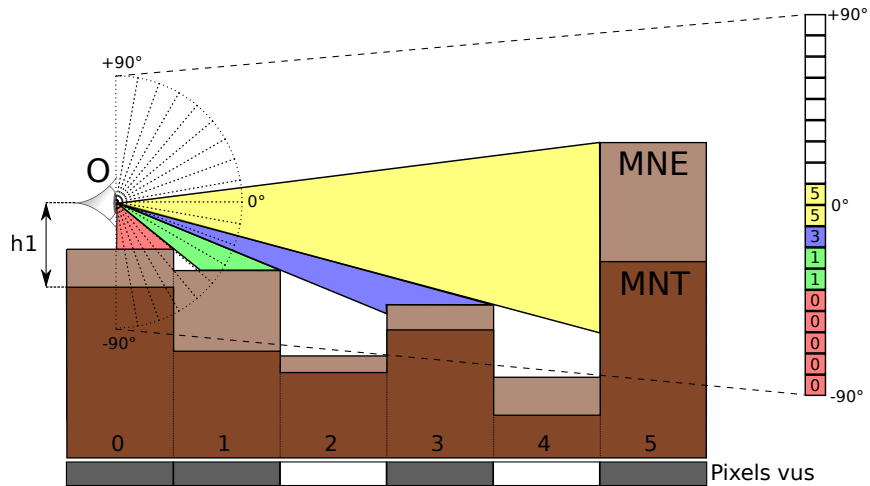


Figure 4.6: Side view of ray tracing. Calculation of the angle heights of the visible pixels for the tangential view. The angular accuracy here is very coarse: 10°.

The juxtaposition of the columns resulting from each ray forms the result of the tangential view. Figure 4.7 shows the position of each ray calculated in the tangential view for an angular accuracy of 30°.

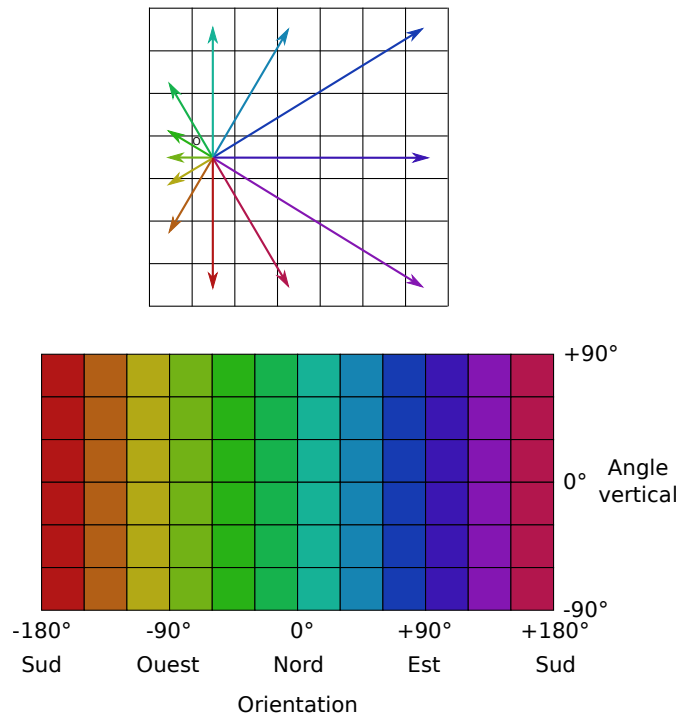


Figure 4.7: Position of each ray calculated in the tangential view. The angular accuracy here is very coarse: 30°.

4.3 Multiscale

The multiscale calculation can greatly speedup visibility processing. Indeed, the algorithmic complexity of a visibility calculation is proportional to the number of pixels of the image n . In multiscale, the complexity

decreases to $\log(n)$ as long as the resolutions evolve geometrically.

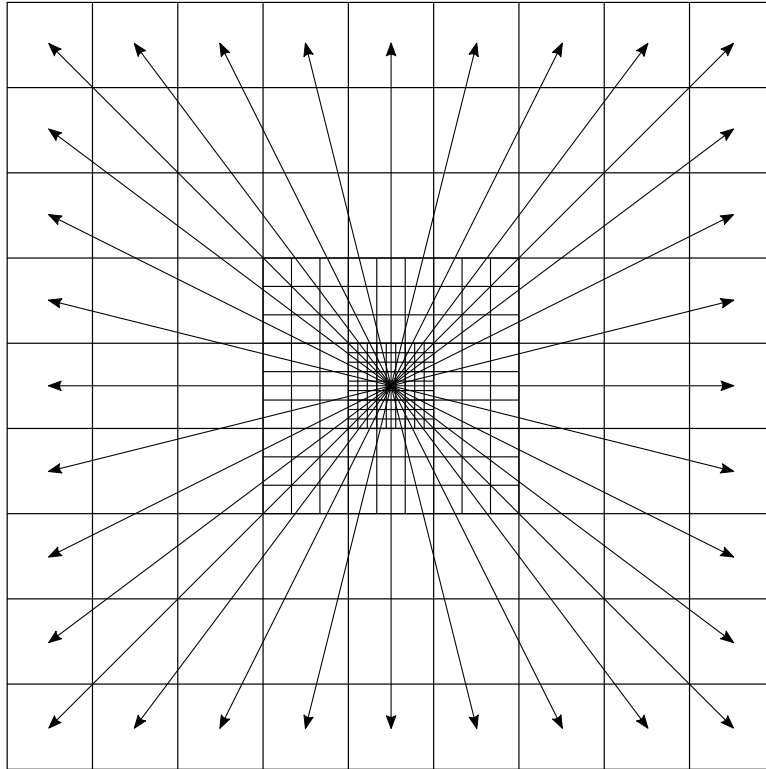


Figure 4.8: Ray tracing on a multiscale database of factor 3 with 3 resolutions

In the example of figure 4.8, the number of calculated rays is 32 instead of 306 and the length of the rays in pixels is 11 instead of 41. The gain in computation time is approximately 35 while retaining the constraint of covering the whole space.

4.4 Eyesight bounds

For all visibility calculations integrated into PixScape, the eyesight can be restricted in the 3 dimensions (figures 4.9 and 4.10) by 6 parameters: the minimum distance (d_{min}) and maximum distance (d_{max}), the minimum vertical angle (z_{min}) and maximum vertical angle (z_{max}), and the horizontal angle by the orientation ($orien$) and the amplitude (amp). The angles are expressed in degrees in the interval $[0-360]$ for the horizontal angles and $[-90; +90]$ for the vertical angles. The distances are expressed in meters and are calculated in 2D on the plane (x, y) .

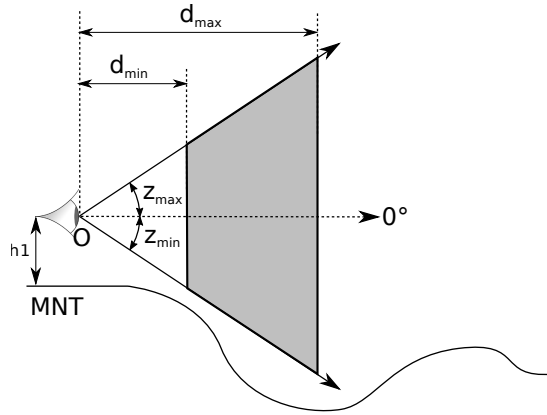


Figure 4.9: Side view of the eyesight bounds

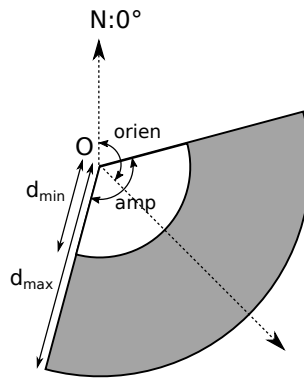


Figure 4.10: Top view of the eyesight bounds

Chapter 5

Metrics description

In this chapter, all the metrics available in PixScape are described. The 5.1 table lists all these metrics and what they support: support of planimetric and/or tangential views, support or not of land use and support of distance intervals.

Metric	Name	Planimetric	Tangential	Without LU	With LU	Distance
A	Area	X	X	X	X	X
S	Shannon diversity index	X	X		X	X
AG	Aggregation index	X	X		X	
CONTAG	Contagion	X	X		X	
ED	Edge Density	X	X		X	
PD	Patch Density	X	X		X	
PMS	Patch Mean Size	X	X		X	
DIST	Distance distribution	X	X	X		
P	Perimeter	X		X		
C	Compactness	X		X		
FD	Fractal dimension	X		X		
SL	Skyline		X	X		X
SD	Shannon depth		X	X		
DL	Depth line		X	X		

Table 5.1: List of metrics and their parameters

5.1 Common metrics

5.1.1 Area: A

This metric measures the visible area in square meters for the planimetric view and in square degrees for the tangential view. The calculation may be restricted to certain categories of land use and for different distance intervals.

5.1.2 Shannon diversity index: S

This metric calculates the Shannon index on the parts of the visible area of each land use category. It is used to measure the diversity of the visible land use.

$$S = -\frac{1}{\ln n} \sum_{i=1}^n \frac{A_i}{A} \ln \left(\frac{A_i}{A} \right)$$

With n the number of land use category, A_i the visible area of the category i and A the total visible area. Areas are expressed in square meters for the planimetric view and in square degrees for the tangential view.

The project must contain the land use layer for using this metric.

5.1.3 Aggregation index: AG

This metric corresponds to the Aggregation index (AI) defined in Fragstat. In PixScape, the calculation of this metric is applied to the visible land use areas only.

$$AG = \sum_{i=1}^n \frac{A_i}{A} AG_i, AG_i = 100 \frac{g_{ii}}{\max g_{ii}}$$

With g_{ii} the number of like adjacencies (joins) between pixels of landuse class i based on the single-count method and $\max g_{ii}$ the theoretical maximum *ie.* when the shape of the landuse class i is a square.

The project must contain the land use layer for using this metric.

5.1.4 Contagion: CONTAG

This metric corresponds to the contagion index defined in Fragstat. In PixScape, the calculation of this metric is applied to the visible land use areas only.

$$CONTAG = 100 \left[1 + \frac{1}{2 \ln m} \sum_{i=1}^m \sum_{j=1}^m p_i \frac{g_{ij}}{\sum_{k=1}^m g_{ik}} \ln \left(p_i \frac{g_{ij}}{\sum_{k=1}^m g_{ik}} \right) \right]$$

It corresponds to the probability of finding a pixel of the land use category i next to a pixel of the category j .

The project must contain the land use layer for using this metric.

5.1.5 Edge density: ED

This metric corresponds to the Edge density (ED) defined in Fragstat. In PixScape, the calculation of this metric is applied to the visible land use areas only.

$$ED = \frac{E}{A} = \frac{1}{2} \sum_{i=1}^n ED_i, ED_i = \frac{E_i}{A}$$

With E the number of edges and A the total visible area in square meters for the planimetric view and in square degrees for the tangential view.

Edges at the border of the visible area are not counted.

The project must contain the land use layer for using this metric.

5.1.6 Patch density: PD

This metric corresponds to the Patch density (PD) defined in Fragstat. In PixScape, the calculation of this metric is applied to the visible land use areas only.

$$PD = \frac{p}{A} = \sum_{i=1}^n PD_i, PD_i = \frac{p_i}{A}$$

With p_i the number of patches of the landuse category i and A the total visible area in square meters for the planimetric view and in square degrees for the tangential view.

The project must contain the land use layer for using this metric.

5.1.7 Patch Mean Size: PMS

This metric corresponds to the average size of a patch in square meters for the planimetric view and in square degrees for the tangential view. The calculation of this metric is applied to the visible land use areas only.

$$PMS = \frac{A}{p}, PMS_i = \frac{A_i}{p_i}$$

With p_i the number of patches of the landuse category i and A the total visible area in square meters for the planimetric view and in square degrees for the tangential view.

The project must contain the land use layer for using this metric.

5.1.8 Distance distribution: DIST

The *DIST* metric gives an overview of the distribution of the pixel distances visible from the observation point by four aggregation operators: sum, average, minimum and maximum.

The result of the metric is different between the tangential and planimetric view, the visible pixels being not distributed in the same way between the two views. Indeed, in planimetric, each pixel represents a surface on the ground in square meters, whereas in tangential view, each pixel represents a cone of the eyesight in square degrees.

5.2 Planimetric metrics

5.2.1 Perimeter: P

The metric P corresponds to the total perimeter of the viewshed, including the whole of contour both external and internal. The unit is in meters.

5.2.2 Compactness: C

This metric takes the compactness index applied to the whole viewshed such as:

$$C = \frac{P}{2\sqrt{\pi A}}$$

If $C = 1$ the viewshed is compact *ie.* a full disk. The more C increases, the smaller the viewshed compactness is.

5.2.3 Fractal dimension: FD

This metric calculates the fractal dimension of the viewshed. The method used is the box counting.

The result is between 0 and 2. When FD tends to 0 the viewshed is reduced to a point, when FD tends to 2 the viewshed covers the space in a homogeneous way.

5.3 Tangential metrics

5.3.1 Skyline: SL

The SL metric measures the more or less uneven aspect of the skyline. It is calculated as the ratio of the length of the skyline (l_h) to a flat horizon line ($L = 360$ deg). The lengths are expressed in degrees.

$$SL = \frac{l_h}{L}$$

SL is close to 1 when the horizon is planar, and highest (unbounded) in the case of a very jagged skyline.

5.3.2 Shannon depth: SD

The SD metric measures the greater or lesser variation of sight lengths.

It corresponds to the standardized Shannon index applied to the distribution of the view depths grouped into m classes. The distance classes are defined by a geometric sequence: less than 10m, 10 to 100m, 100m to 1km, 1km to 10km, and more than 10km.

$$SD = -\frac{1}{\ln m} \sum_{i=1}^m \frac{nd_i}{n} \ln \left(\frac{nd_i}{n} \right)$$

with nd_i the number of rays with a maximal length inside the class i and n the total number of rays calculated in the tangential view.

This metric is 0 when all lengths are inside the same class, the view depth is homogeneous. Conversely, SD is 1 when the ray lengths are equally distributed on all classes, which represents a large variation of view depth.

5.3.3 Depth line: DL

The DL metric also measures the greater or lesser variation of the depths of view but without distance classes a priori as SD .

This metric requires the construction of a polygon on the plane (x, y) grouping the visible points furthest from the observation point for each ray being traced. The DL metric is defined as the compactness index of this polygon:

$$DL = \frac{p}{2\sqrt{\pi a}}$$

where p and a represent respectively the perimeter and the area of the polygon.

This metric gives a minimum value of 1 in the case of equal view depths (the polygon is a circle), and high (unbounded) values in the case of strong variations in view depths.

Chapter 6

Performance tuning

6.1 Parallelism to speed up execution

To reduce visibility calculation execution time, PixScape implements 3 parallelization methods: thread based for a single computer, CUDA to use Graphic Processor Unit (GPU) acceleration, and MPI for computer clusters.

The table 6.1 shows the execution times to calculate 10000 views according to the type of view (planimetric or tangential), the size of the DTM in million pixels and the parallelization: 1 core (without parallelization), 4 cores, 16 cores and GPU (NVidia Tesla K40 graphics card).

Type	Size	1 core	4 cores	16 cores	GPU
Planimetric	1 M	9,8	2,5	0,7	0,6
	9 M	103,0	26,0	6,7	1,8
	100 M	1168,0	376,0	83,0	13,2
Tangential	1 M	36,0	9,8	2,7	0,8
	9 M	113,0	28,8	8,0	2,2
	100 M	357,0	91,0	24,7	6,3

Table 6.1: Average time in minutes to calculate 10000 views.

6.1.1 One computer : threads

Multi-threaded parallelization accelerates computation time on a single machine containing multiple cores or processors. If your computer has more than one core (most of them), you can take advantage of multi-threading to speed up your calculations with PixScape. The execution time is approximately inversely proportional to the number of cores used.

This parallelism method can be used in GUI and CLI.

Graphical user interface

The "Preferences" window accessible from the File menu allows you to set the number of cores used by PixScape. It is set by default to the number of cores of the computer minus 1. After changing the number of cores used by PixScape, it is better to restart PixScape to be sure that the change is taken into account.

Command line interface

On the command line, you have to define the number of cores (or processors) that PixScape can use with the `-proc` option :

```
java -jar pixscape-1.2.jar -proc 8 --project path2myproject/myproject.xml ...
```

By default, CLI uses the number of processors defined in the preferences window of the GUI.

By increasing the number of cores used by PixScape, you increase the memory size used by PixScape at the same time.

6.1.2 Graphics card : CUDA

This parallelism method makes it possible to carry out the visibility calculations on the graphics card instead of the processor. Runtime gains can be significant for Nvidia Tesla graphics cards, designed for high-performance computing (see [Average time in minutes to calculate 10000 views.](#)). However, not all metrics are optimized for the GPU. Metrics A and S are the only ones to be optimized. If you use other metrics, the time savings will be minimal or non-existent.

To use GPU acceleration, you need a Nvidia graphics card that supports CUDA and CUDA version 6.5 installed.

If PixScape fails to use the GPU, the calculations will be automatically switched to the processor.

This parallelism method can be used in GUI and CLI.

Graphical user interface

The activation of the GPU acceleration can be done from the Options window (see [Options](#)).

In graphical interface, PixScape can only use one graphics card, if your computer contains several graphic cards, PixScape can use them in parallel but only on the command line.

Command line interface

At the command line, the `-cuda` option activates the use of the GPU. It is also necessary to give the number of graphics cards to use.

```
java -jar pixscape-1.2.jar -cuda 1 --project path2myproject/myproject.xml ...
```

In the example above, PixScape use one graphics card for visibility calculations. If your computer contains several graphics cards, it is possible to increase the parameter to use them in parallel and decrease the time of execution accordingly.

6.1.3 Computer cluster : mpi

PixScape can also be used on computing clusters supporting Java with OpenMPI. This parallelism method is only available on the command line.

Exemple :

```
mpirun java -jar pixscape-1.2.jar -mpi --project path2myproject/myproject.xml ...
```

Only some commands can be used in the MPI environment: `--planmetric`, `--tanmetric`, `--landmod`

6.2 Memory management

By default, the amount of memory available for PixScape is system dependent. It can vary from 128 Mb to several Gb. In most cases, PixScape will run normally. But if you have a large project, some processes would be slow or even crash due to memory limitation.

Also, if you use thread-based parallelism ([One computer : threads](#)), PixScape will need more memory for each core used.

In all cases, if PixScape execution terminates with `OutOfMemoryError` or GC overhead, you need to increase memory allocated to PixScape.

6.2.1 Graphical user interface

The memory allocated for PixScape can be changed in the "Preferences" window on the File / Preferences menu. After changing this setting, PixScape must be restarted to take the new memory size into account.

In GUI, PixScape needs more memory than command line because of the layers display. If you are too limited in RAM, you can run the calculations on command line interface to reduce PixScape memory requirements (see [Command line interface \(CLI\)](#)).

6.2.2 Command line interface

To define manually the maximum amount of memory allocated to PixScape, use Java option `-Xmx`:

```
java -Xmx4g -jar pixscape-1.2.jar ... # 4Gb allocated
java -Xmx1500m -jar pixscape-1.2.jar ... # 1500 Mb -> 1.5Gb allocated
```

If you cannot allocate more than 1Gb or 1.5Gb and your computer has more memory available, you have probably a 32-bit version of Java, which is limited to less than 2Gb of memory. Check your Java version:

```
java -version
```

If it is a 32-bit version, install a 64-bit Java version to handle all your computer memory.