

# Mosaizer XV Help

# Table of Contents

1. About Mosaizer XV.....	3
1.1 What's new .....	4
1.2 Features of Mosaizer XV .....	14
1.3 System requirements.....	16
1.4 User interface .....	17
1.5 Online tutorials .....	21
2. The basic XV functions .....	22
2.1 Mosaic creation.....	22
2.2 Colour matching.....	30
2.3 Library creation .....	32
2.4 Masks .....	35
2.5 Smart patterns .....	40
2.6 Transparency.....	43
3. Special XV functions.....	48
3.1 Multiple masks.....	49
3.2 3D effect.....	50
3.3 Edit, search and replace.....	51
3.4 Preprocessing.....	53
3.5 Smart masks.....	53
3.6 Non-rectangular shapes.....	55
3.7 Single-cell mosaics .....	59
3.8 Projects .....	61
3.9 Super mosaic.....	61
3.10 Grayscale mosaics .....	63
3.11 Stone mosaic data.....	65
4. Mosaizer Video .....	66
4.1 Interface.....	66
4.2 Mosaic animations .....	70
4.3 Mosaic movies .....	75
5. Tools.....	81
5.1 Glossary.....	82
5.2 File locations .....	83
5.3 Licenses .....	83



# 1. About Mosaizer XV

**Mosaizer XVi** and **Mosaizer Video** are a digital toolset to create photo mosaics, for pictures and for video.

Unlike programs that create photo mosaics only, this program gives you almost unlimited freedom to create your photographic paintings yourself, based on a library of bitmaps with an extensive range of supporting creativity tools. We offer an exotic range of additional effects to play with pattern, texture, shadow, colour, shape and other effects that you find in many bitmap editing software. Actually, we have combined the basics of photo editing and painting with the possibilities of photo mosaic creation.

**Mosaizer XVi** is based on three fundamental creativity tools: pattern generation, transparent masks and the Mosaizer. The basis of each photo mosaic is selecting the pattern for putting the individual picture tiles. With masks you can leverage the transparency and colour match of bitmaps to create surprising effects. You can add your own masks. The 'Mosaizer' is a tool to recreate a bitmap from pieces of itself, like a jig-saw. In case you would like to know more... check out [this](#) promotion video on YouTube.

**Mosaizer Video 2.0** takes mosaic creation one step further: creating mosaic videos. There are two possibilities: animating from a still source picture mosaic, or using actual video footage, and recreate that footage to a mosaic movie, with zoom in and out, enlarging the tiles in the process. As far as we know this is a truly unique feature, only accessible to professional video services.

One word of caution. Although we have tried to make the interface, the function and use of the different controls and the work flow as easy as possible, the user has to keep in mind that mosaic creation is a *highly professional job*. On the web many similar products can be found who claim to have a 1-2-3 step approach, but they typically lack in options, pre- and post-processing and quality management. It's therefore highly recommended to frequently consult this help documentation.

**Mosaizer XV or XVi?** This help document is based on the interface of Mosaizer XV and Video 1.0. The latest versions, starting with Mosaizer XVi and Mosaizer Video 2.0 don't have a dedicated web-based help any more. These applications come with a Manual and in-app support for most controls and group of controls. Nevertheless, this web-based documentation is still very valid because 1. the interface is quite similar of the new versions, and 2. the content is almost identical, is explaining the key features, showing examples and following the tutorial chapters. We ask of you some flexibility when using this document while running the newest builds. Thank you!

Great worked-out examples are found on our website. Your imagination is the only limit !

## **Easy installation and quick start**

After you have installed the package you can start immediately, because the picture library is part of the installation. We have made additional picture libraries available from our website: [www.appheland.com](http://www.appheland.com)

Thank you for using **Mosaizer XVi** and **Mosaizer Video 2.0**.

[Current versions](#): 15.0 build 55 (XV) and 1.0 build 27 (Video).  
7 June 2020

[Beta \(test\) versions](#): 16.0 build 16 (XVi) and 2.0 build 22 (Video).  
27 September 2022

## 1.1 What's new

### The Mosaizer XVi & Video change history

Mosaizer XVi 16.0 build 16 - September 2022

- Optimized for Windows 10/11
- High DPI compatible (scaling along with Windows scaling)
- Simplified interface, more intuitive and logical placement
- New and more informative interface when opening a library
- New Manual for easy access to help documentation
- Now the application starts without an image
- Tips and information on the fly in the interface
- A new image library added to the installer
- Two new languages: Spanish and Frisian
- Improved Single Cell method
- Fixed a bug in the grayscale mosaic engine
- Shade effect now previewed
- Code optimizations

Mosaizer Video 2.0 build 22 - September 2022

- Optimized for Windows 10/11
- High DPI compatible (scaling along with Windows scaling)
- Simplified interface, more intuitive and logical placement
- Genuine FFMPEG video output (X.264) and other current video formats
- Redevelopment of the video movies from video footage
- Many (although rarely encountered) issues have been addressed and resolved
- More robust error management, allowing to continue to work uninterrupted
- New and more informative interface when opening a library
- New Manual for easy access to help documentation

---

15.0 build 55 - June 2020

- An issue was solved when a grace period overlapped with a valid license

15.0 build 54 - November 2019

- Now also the grayscale image can have a solid colour background
- The Super-mosaic option now also indicates where the image was stored

15.0 build 53 - January 2019

- Changed the library image resize approach to become easier to use
- Added fast evaluation for grayscale match quality of picture library and source picture
- Robust new source image read (maximum 100 MB to avoid reading issues)

## Mosaizer XV Help

- Improved stability for smart patterns

15.0 build 52 - March 2018

- Fixed an issue when all tiles are placed with colorize and blend
- Improved and more logical tile edit using colorize and blend

15.0 build 51 - August 2017

- A new feature was added to use edited tiles when redrawing the mosaic

15.0 build 49 - March 2017

- for (.png) transparent sources now a checkered pattern is seen
- now also .ico and .gif images can be used in libraries
- fixed an issue when an existing library was replaced
- added an automatic library repair possibility

1.0 build 27 - March 2017

- now also .ico and .gif images can be used in libraries
- fixed an issue when an existing library was replaced
- added an automatic library repair possibility
- fixed a bug when adding a new library

15.0 build 48 - February 2017

- improved the automatic supersize divisions
- amount of super divisions can now also be edited
- saving tile location data now also possible for a grayscale mosaic

15.0 build 46 - January 2017

- fixed a critical issue with 'colorize'

15.0 build 44-45 - January 2017

- fixed an issue with engine 'Force to use all pictures'
- restored mosaic picture location and identification outputs
- fixed an issue of the tile editor for small libraries
- added horizontal scrollbar in the library folder explorer

15.0 build 43 & 1.0 build 25 - December 2016

- fixed a freeze bug when trying to find an update without internet connection

15.0 build 42 & 1.0 build 24 - December 2016

- fully redeveloped application
- numerous code optimizations and improvements
- separate applications for picture and video
- automatic super-size mosaic creation
- transparency management simplified
- improved stability

- paintbrush feature was left out
- double mosaic feature was left out

---

## **The Mosaizer Pro change history**

### 12.2 build 126 - May 2016

- Now more than 100k pictures can be used in a library

### 12.2 build 225 - April 2016

- now also 4K movies can be created
- now also the movie frames can be saved (for post-processing)
- minor stability improvements

### 12.2 build 223 - October 2015

- Smart Patterns now also accepts the 'Optimization' engine
- maximum size error now better managed for Smart Pattern

### 12.2 build 222 - August 2015

- post-processing issue has been fixed

### 12.2 build 221 - August 2015

- new and efficient blend and colorize method for all patterns
- non-rectangular patterns have drastic speed increase
- a new non-rectangular pattern has been added: diamonds
- safe mosaic size has been increased to appr. 325 MB
- improved memory usage for large bitmaps (> 100 MB)
- undo feature added for density mask drawing
- grid drawing now fully anti-alias with grid thickness option
- minor stability improvements (non-critical)

### 12.1 build 215 - August 2015

- issue solved for non-rectangular shapes with small libraries
- a picture library integrity check added
- now also variety of 1 is possible (= no random tiles, only best fit tile is used)
- for small libraries (< 100 pictures) the quality is now always 100%

### 12.1 build 214 - July 2015

- bug in library creation solved

### 12.1 build 213 - June 2015

- size limits have been lowered to avoid memory issues

### 12.1 build 212 - May 2015

- bugfixes in engine 'Force to use all tiles'
- Turkish language added

12.0 build 211 - May 2015

- bug in 'delete library' fixed

12.0 build 209 - May 2015

- improved project read
- improved stability of black and white mosaic analysis

12.0 build 208 - February 2015

- colorize has been improved drastically
- video codec is now correctly remembered
- video conversion via ffmpeg is now working correctly

12.0 build 207 - February 2015

- masks can now also be used in movie creation
- the single cell mosaic can now also use a different picture

12.0 build 205 - January 2015

- a minor bug in the multiple page printing options has been resolved
- a bug has been solved when switching a library for the second time

12.0 build 204 - December 2014

- a minor bug in the multiple page printing options has been resolved
- a bug has been solved when switching a library for the second time
- a couple of minor issues solved with video creation

12.0 build 203 - November 2014

- a fully new interface: easy to use, focused controls and less cluttered
- a new feature: density masks to manually determine the tile sizes
- smart grids can now also be used for cropped parts
- Grace Days added to use the fully unrestricted application for 14 days
- bug in the smart pattern visualization has been solved
- larger preview and navigation window
- bug in rotating tiles has been solved
- anti-alias quality now available for all users (restriction removed for free versions)
- improved folder selection for the automation and library creation features

---

**The Mosaizer Pro 11 change history**

11.0 build 187 - July 2014

- the application now correctly requests admin rights at start (a long awaited bug fix!)

### 11.0 build 185 - January 2014

- added a video capability to create real mosaic video movies from video footage
- video features allow to fully use the rectangular pattern options
- special shapes (circular, spiral, hexagonal, smart patterns) are excluded in mosaic video movies

---

## The Mosaizer Pro 10 change history

### 10.1 build 172-175 - September 2013

- progress bar under Windows 7 and 8 now also visible in the task bar
- now automatic transparency added for .png source files for mosaic automation
- error corrected with transparent source files (now transparency is correctly showing in the thumbnail)
- new feature added to allow horizontal flipping of each tile, virtually doubling the size of the pictures library, and creates a better colour match
- new feature added to automate mosaic creation with multiple source files and libraries
- widened the width-height ratio of the cells, now ranging from 0.25x to 4x

### 10.0 build 170-171 - August 2013

- added a selector for the resampling quality
- added a new resampling engine for razor sharp results
- corrected the 'print options' skin

### 10.0 build 165 - January 2013

- transparency fully implemented: all mosaics can be saved with the alpha-channel
- double-layered mosaics: the bottom layer being a full mosaic, the top layer being a transparent layer with chroma-key colour removal options
- chroma-keying implemented: colours can be removed (using HSL levels and ranges/tolerance) with blurring capability to create a seamless blend with the underlying layer (or remain transparent)
- option to save in the 32bit RGBA formats: .PNG or .PSD (allowing for red-green-blue-alpha channels)
- intuitive and responsive chroma-key interface with proofing bitmaps to accurately tune the chroma key values
- chroma-keying and transparency also valid for super-mosaics and smart grids
- removed a bug in the genuine paintbrush (can now also paint non-square final results)
- create a picture library from a video movie

---

## The Mosaizer Pro 9 change history

### 9.3 build 158 - December 2012

- improved smart grid, now correctly sizing the bottom and right edges

- super mosaics now capable to also manage the normal options, such as (multiple) masks and random placements; the parts can now be stitched together seamlessly in e.g. Photoshop to create Gigabyte sized mosaics

### 9.3 build 155 - December 2012

- minor bugs resolved
- Pro+ options now available

### 9.2 build 152-154 - November 2012

- high quality tile rotate approach (anti aliasing enabled)
- added a few rotation masks (rectangular and square)
- added a security check for multiple masks (rotation and resize can't be used with multiple masks)
- improvement of the spiralled tiles (white spaces now won't appear)

### 9.2 build 149 - July 2012

- added a library merging tool
- added library information in the menu bar

### 9.2 build 144-147 - June 2012

- projects now also save the smart grid data
- bug in libraries with very small amount of tiles solved
- robustness of smart grid improved
- right and bottom rim of extremely large mosaics now correctly created
- edited tiles are now correctly re-used in super mosaics
- bug in blending of super-mosaic now solved

### 9.2 build 140-143 - May 2012

- improvement in black and white (ratio library/source tiles now unlimited)
- improved the controls for very tall/wide sources in 'super mosaic'
- improved mask resize for smart grids
- improved the quality of the mosaic edges for random grids

### 9.2 build 135-138 - April 2012

- improved black and white matching
- multiple alpha masks now available (re-enabled)
- minor bug in smart ('adaptive') grid solved

### 9.2 build 129-134 - April 2012

- correction in the non-rectangular pattern controls
- improvements in smart grid to avoid a partial mosaic built at low tile amounts
- improvements to better match the final mosaic size with the amount of tiles
- final release of greyscale engine

### 9.2 build 128 - March 2012

- beta release of the new greyscale engine
- error removed in library creation
- library delete now available
- black and white grey scale analysis added (part of black/white mosaic optimizer in 9.2)

#### 9.2 build 127 - March 2012

- on request: new engine added to optimize black and white mosaics

#### 9.1 build 125-126 - February 2012

- on request: 100% unique tiles for colour matching method 'Force to use every picture'
- potential error when saving the mosaic data file (if file is in use) solved
- some internal optimizations
- on request: library can now also be deleted

#### 9.1 build 124 - February 2012

- on request: super mosaic creation to create a giant mosaic in parts
- bug in 'Force to use every picture' now solved
- library creation interface fixed in the right skin
- redesign of the 'optimized tile distribution' approach

#### 9.0 build 116-118 - January 2012

- new feature added: smart patterns
- new option added to rebuild the mosaic after tile edits
- bugs removed in tile edit
- more realistic cell size range for different paper size
- non-rectangular patterns (circles, hexagon, spiral) now also can use transparent sources (.PNG files)
- 4 more skins added
- fully redesigned tile distance and -distribution optimization algorithm, now a factor 50 faster
- transparent (32bit) masks can now also be in .PNG file format
- new feature: a starting wizard for a swift application acquaintance
- bug solved in transparent source files (.PNG), now the transparent part is clearly indicated
- defaults reloaded when for some reason no source file, mask or texture is available (avoiding error messages)
- bug in transparent (.PNG) files when used for a first time (a black background) now solved
- bug solved in the printing, now printing correctly after having completed a mosaic result
- colorize and blend slider controls made simpler and more responsive to value changes
- 4 new languages added

---

### **The Mosaizer Pro 8 change history**

#### 8.2 build 99-100

- improved source file and library management
- last used library and source file now loaded at start
- small inconvenience in max amount of cells repaired

#### 8.2 build 95-98

- bugs removed in saving projects
- option added to save stone mosaic tile data
- minor corrections in skin consistency with dialogs
- added a possibility to show the current save and read locations (folders)

#### 8.2 build 93-94

- installer changed to install the application for all users on the system (admin rights needed to install)
- small visual bug in printer window removed

#### 8.2 build 92

- size errors ('Can't allocate DIB handle') now finally dealt with
- size range changed to lower the possibility of memory allocation errors

#### 8.2 build 90-91

- automatically ignoring library source file errors (file size = 0 kb or < 4 pixels in size)
- small updates in skin layout

#### 8.2 build 89

- new skinnable interface
- a number of small bugs removed
- maximum cell size extended to 2436 pixels
- grid visualization improved

#### 8.1 build 84-88

- a bug was found in the background colour which has now been corrected
- although transparent pictures can be used (.png files), these files can now also be opened

#### 8.1 build 83

- registration removed: the application is now completely functional and free ware
- skins added

#### 8.0 build 82

- French translation added

#### 8.0 build 81

- improved alignment of mouse-pointer and cells when painting (Paintbrush functionality)
- more logical and correct placement of the installation files in both Vista, XP and Windows 7

#### 8.0 build 80

- png (bitmap) transparency is used to skip miniature picture placement
- full randomization assures that a full randomized picture is created, using all tiles

#### 8.0.build 78-79

- error when source file does not exist (hang-up) now solved
- small errors in the translations have been repaired
- small but harmless bugs solved (not reported so far)

#### 8.0.build 77

- final stable release
- variable paintbrush eraser size added

#### 8.0 build 76

- the genuine paintbrush has been added
- bug in non-rectangular tiles now solved
- bug in edit tiles solved
- when scrolling over the result, a position indicator is seen
- different e-mail address for providing immediate feedback (serial request only)

#### 8.0 build 75 - February 2011

- a new release of version 8.0

---

### **Time machine for Mosaizer Pro**

- 15.0 - December 2016
- 12.1 - 16 May 2015
- 12.0 - 20 November 2014
- 11.0 - 4 January 2014
- 10.1 - 8 September 2013
- 10.0 - 4 January 2013
- 9.3 - 20 December 2012
- 9.2 - 23 March 2012
- 9.1 - 27 January 2012
- 9.0 - 1 January 2012
- 8.3 - 17 December 2011
- 8.2 - 14 April 2011
- 8.1 - 5 March 2011
- 8.0 - 26 February 2011
- 7.2 - 9 November 2009
- 7.1 - 31 May 2009
- 7.0 - 15 November 2008

- 6.1 - 1 October 2008
- 6.0 - 16 February 2008
- 5.0 - 11 February 2009
- 4.5 - 18 August 2007
- 4.4 - 30 July 2007
- 4.3 - 24 March 2007
- 4.0 - test release (January 2007)
- 3.0 - unreleased (August 2006)
- 2.0 - unreleased study (February 2004)
- 1.0 - unreleased study (January 2004)

Mosaizer Pro has quite a history of active development

With the first versions that were offered in the internet, in March 2007, we showed the strength and versatility of Mosaizer 4.x. With version 5.0 we have improved on the concept of Mosaizer, adding several features, convenience of use and more functionality. With version 6.0 we came close to a final version as we originally had in mind. Mosaizer 6 had a very different look and feel: not unlike Photoshop the user could hide and reposition tables of functions, and create his own interface. In that period many reactions have been posted to us, and we are very grateful for the tips and examples that were shared. But the application appeared to still be unstable, and the high complexity of the interface. With Mosaizer 7 we tried to solve (November 2008 - November 2009), but unsuccessful. We had to abandon the familiar interface, which we started with 4.x and 6.x. In September 2010 we started the development of Mosaizer 8, which took about 4 months in total.

We went back to the drawing board and also redesigned the whole interface, and along adding simplicity in the architecture of the software, creating a much more stable application. In February 2011 Mosaizer 8 was released. The interface was fully redesigned, where now professionally looking attractive skins were used. With Mosaizer Pro 8 the basic choice to favour 'speed' over mosaic 'size' has proven a good one. New features were introduced, like the 'genuine paintbrush' and source file transparency. Our application is still unrivalled in speed. The second fastest application that we found on the internet took over 40% more time for making the same mosaic. Other (shareware) applications are still far below the speed capabilities of Mosaizer Pro (even a factor of 10 slower was found!). This unrivalled speed is still not challenged by competitive applications.

Building on the strengths and speed of Mosaizer 8, we started implementing exciting new features and possibilities. In Mosaizer Pro 9 we included dynamic tile sizes, smart patterns, a wizard to start simple, .png mask files and again another improvement in speed by implementing two brand new tile placement optimization engines. Also the use of multiple masks in the same mosaic was implemented. These features were requested by many users and were finally realized in Mosaizer Pro 9.

In December 2012 we started to add yet another portfolio of new features, resulting in Mosaizer 10. This latest version has implemented a unique feature: multiple layers of mosaic pictures, of which the top layer can be made transparent. Finally, the concept of chroma-keying was introduced in mosaic making! Mosaizer 10 has also added the possibility to fully save the mosaic as a transparent bitmap, fully conserving the transparency of the individual tiles. Again, a truly unique feature, never found anywhere in mosaic creation software.

With versions 11 and 12 the application grew to a most popular and versatile application, used by thousands of people. We got media attention, even made it on



- library merging tool
- Mosaic dynamics: video from moving tiles
- Single-cell mosaizing: building a picture from repeatedly using its own picture
  - also a different picture can be used for the repeat
- Transparency of tiles
  - read and save as transparent file (.PNG or .PSD format)
  - advanced use of chroma-keying in colours
  - chroma keying in degrees of white
- Pre-processing features
  - fast evaluation of the colour match quality of miniatures picture library and source picture
  - smart (adaptive) tile sizes
  - horizontal flipping of each tile, virtually doubling the size of the pictures library, to create a better colour match
  - colour tuning filters to increase the overall photo mosaic quality
  - 3D Graphic effects: tile shade and texture overlay to create a 3D surface effect
  - cropping of the source picture to render only a part of it
  - transparency in source files to select areas of tiling using .PNG file types
  - analysing black and white library versus source grey-scales
  - 4 size resampling engines for setting the quality of the output mosaic
- Post-processing filters to enhance the result
  - blending with the original picture
  - colourizing each individual miniature picture (aka cell or tile)
  - editor: search and replace individual pictures of the result
  - contrast, luminance and saturation
  - super mosaic creation to generate any size mosaic
  - redraw with edited values to create editable super mosaics
- Advanced mask functionality
  - alpha masks (controlled transparency)
  - 32Bit masks where the alpha channel is used to control the pixel-by-pixel transparency of the picture
  - multiple masks in the same mosaic picture

### **Key characteristics of Mosaizer XV & Video:**

- Extremely fast and accurate colour matching
- A superior colour matching algorithm
- Different colour matching methods
- Dedicated grey-scale matching method for greyscale mosaics
- Option to apply smart ('adaptive') cell sizes
- Multiple file formats: bmp, jpg (jpeg), png, ico and gif
- Transparent source files can be used to only create a mosaic in opaque parts
- Select different background colours or use the original source picture
- Non-rectangular shapes (circles, spirals, hexagons and diamonds)
- Source picture cropping to only render what is required
- Hardly any limit to the result output (external post-processing/stitching is required for > 300 MB mosaic)
- Cell variation and cell distance control
- Load and save projects ("presets")
- Work with transparent bitmaps
- Multiple masks in the same mosaic
- Multi-language support
- Printing the result over multiple pages

- Option to save the mosaic data for creating (stone) mosaics
- Well documented help, examples and web-page
- Picture libraries and tutorials are available for download from the website
- Shareware, unrestricted and fully functional during the grace period

**Limitations of the free version:**

- Limits the output size to A3 paper (40 MB output size)
- Only 6 masks in the multimask selection
- Only 3 levels for smart grids
- No automatic super-size mosaics
- Demo nag text in mosaic video

## 1.3 System requirements

- 1.8 GHz Intel or AMD processor
- 4 GB RAM memory
- 50 MB of hard disk space for picture libraries
- 100 MB of hard disk space to store photo mosaic pictures
- Windows XP 32bit
- 19" Monitor with 1280 x 1024 pixels (4:3)

**The following system is recommended:**

- 2.9 GHz Intel or AMD processor
- 16 GB RAM memory
- 100 MB of hard disk space for picture libraries
- 750 MB of hard disk space to store big photo mosaic pictures and movies
- Windows 7 64 bit
- 21.5" Monitor with 1920 x 1080 pixels (16:9)

The Mosaizer XV rendering time is about proportional to the CPU speed. The choice of video card is not of great importance for Mosaizer XV, but the choice of CPU is of great influence. In general the application runs faster on (Intel) i7 then on i5, which in turn runs faster than on i3 machines. The amount of RAM is important for running the application, although more RAM will not result in higher speed. More RAM simply manages the internals faster than less RAM.

**Microsoft Windows versions and Service packs**

Mosaizer XV has been designed to run on all (current) Windows systems. The application has been optimized to run on Windows 7/8 (32 and 64 bit) systems. Mosaizer XV has been tested on machines with Win XP-SP1, Win XP-SP2, Win XP-SP3, Vista Home Premium (and with SP1), Vista Pro Ultimate (XP and Vista: both 32bit and 64bit versions) and on Windows 7, 8 and 10 (also 32bit and 64bit versions, and 32bit with SP1). All systems work with Mosaizer XV with no known incompatibility issues.

Vista and Windows 7/8/10 users may experience the 'Floating point division by zero' after installation. This is because Mosaizer XV needs to write on the installation drive (usually C:) and Vista/Win7/Win8/Win10 needs administrator rights to allow this. When no special precaution is taken the required data to start the application is not read and written properly and Mosaizer XV will generate this error. The workaround is very simple: start Mosaizer XV as admin (right-click +

'run as administrator').  
See [here](#) for more details.

### Mac users with OS-X

Mosaizer XV only works on Windows systems. In case you wish to use a mac, you first need to install a Windows environment. Most users use Wine (freeware), Parallels or Bootcamp, to install a second operating system. This only works on macs with an Intel processor. From reports we receive of mac users this appears to work very well.

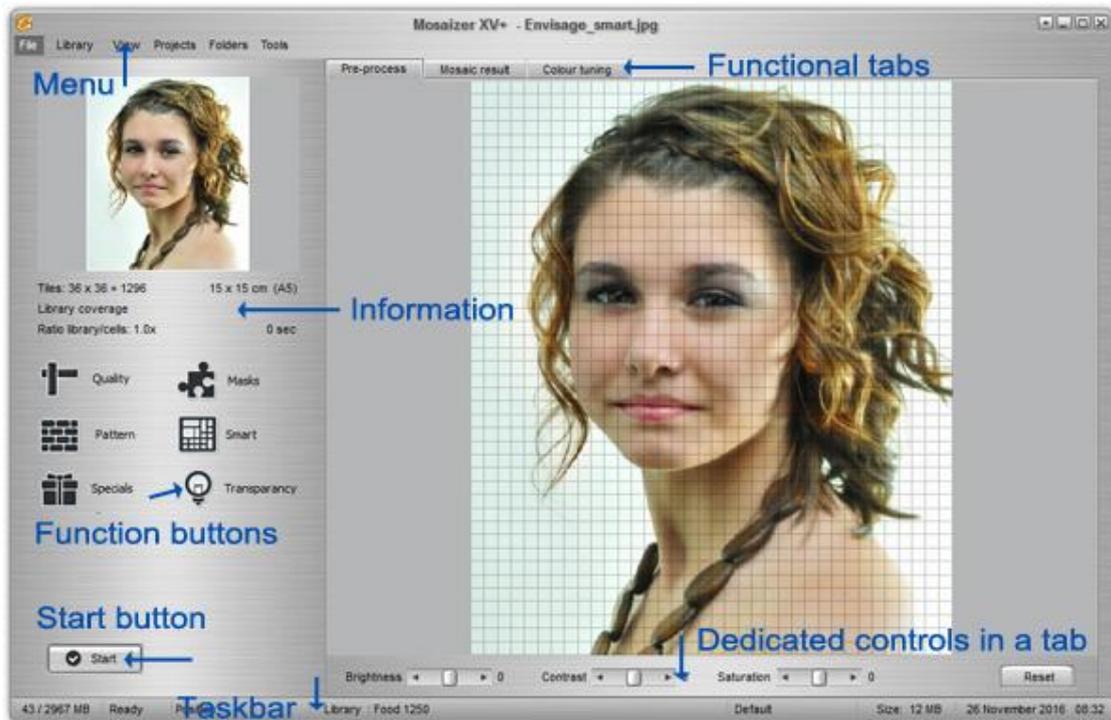
### Mosaizer XV may run slow on older OS's

Assuming that older systems have slower CPU's, Mosaizer XV is not recommended on Win95 or Win98 systems to avoid excessive long rendering times. We have not tested on Win98 systems. The skin feature of Mosaizer XV is expected to run without problems on these systems.

### Known issues

The most notorious issue is '[Can't allocate the DIB handle](#)'. It is a complicated error and is difficult to predict when it will happen. It is a message to indicate that Windows is not able to allocate sufficient memory to create the mosaic picture. It seems that on Windows 7/8/10 systems the memory allocation is better than on XP systems, but still the error pops up when a too large claim on the system memory is done, e.g. when you want to create a very large mosaic (such as > 300 MB in size). For example you can create a 400 MB mosaics on a Windows 7 laptop with 3GB of RAM, but only a 300 MB mosaic on a XP machine with 4GB of RAM, so the available RAM doesn't seem to be of influence (but perhaps it does).

## 1.4 User interface



### Menu

Here you find the controls for managing input and output ('File', 'Library', 'Projects'), show or hide parts of the interface ('View'), and convenient shortcuts to open the Windows Explorer of a certain folder (e.g. 'My Pictures' folder), and finally the tools ('Settings', 'Registering', 'Help', etc).

### Information

This is the picture that is used to recreate from a library of (smaller) pictures. Some basic information is also shown here: tile size; the amount of library coverage when the current size is applied; and how many tiles are required compared to the available amount in the library. One word on the coverage: this is a figure that indicates how many tiles have been applied in the last mosaic rendering relative to the full library size. For example: 67% means that the last mosaic rendering has used 67% of all the pictures in the library at least once.

### Function buttons

Each function button will show its corresponding controls in a pop-up window. In this pop-up window you can set the parameters that apply to the selected function. Only one pop-up window can be open. So, when clicking another button the content of the open pop-up windows is replaced with the selected function.

### Functional tabs

These are the workspace tabs that are required to show the result, pre-process, post-process etc, and some additional functions that require an user interaction, like creating and drawing a so-called 'density mask' . At the bottom of each of these tabs the applicable controls are visible. Each tab has **dedicated controls** that are relevant for the tab.

- 'Pre-process' pre-filter sliders are found and a checkbox if the pattern/grid would be shown.
- 'Mosaic result' the zoom buttons are shown, incl. a seamless zoom slider.
- 'Colour tuning' the colour tuning controls and the texture parameters are found here.
- 'Library' all the parameters and controls are found in this tab to create new libraries, and merge existing ones.
- 'Smart mask' the controls for creating a black and white mask (for use in smart pattern) to manage the size of each tile manually.

The user can move the picture in each viewer window by dragging the picture by the mouse (press left button while moving the mouse in the desired direction). This only works if a magnification is used.

### Dedicated tab controls

Each functional tab can have a couple of controls. Some control are common (zoom), other are specific and are shown or hidden, depending which tab is open. The control values of a tab will not change with selecting a different tab.

### Start button

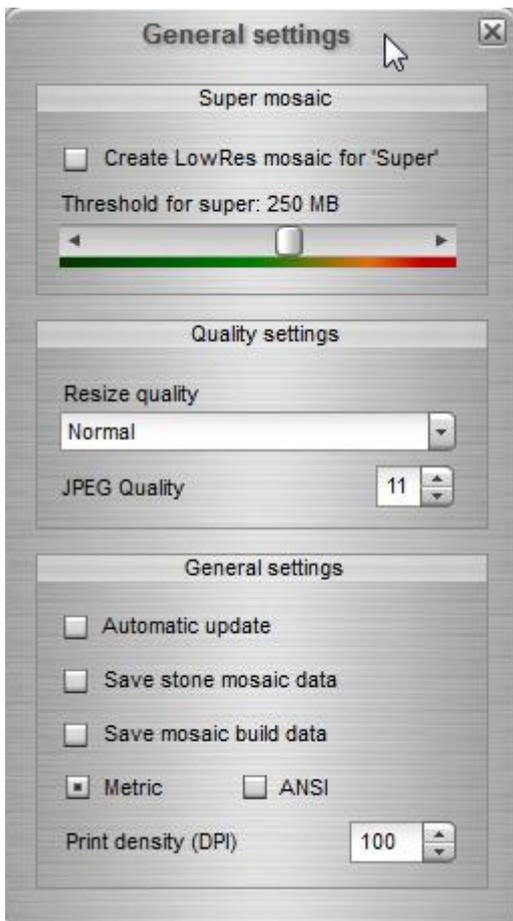
The start button is of course the button you need to press to create a new mosaic. When the application found no obvious errors, the button will show a small 'OK' icon. In case a super-sized mosaic will be made, the icon changes to an exclamation mark. When a serious (potentially error) issue is expected the 'NOK'

icon will appear, and in most cases the button is then disabled until the issue is solved.

### Taskbar

Here useful information is found for the application. From left to right: RAM usage, working status, cursor location and selected portion size, library name, project name, mosaic result size (bitmap size in RAM) and the time/date.

### General settings

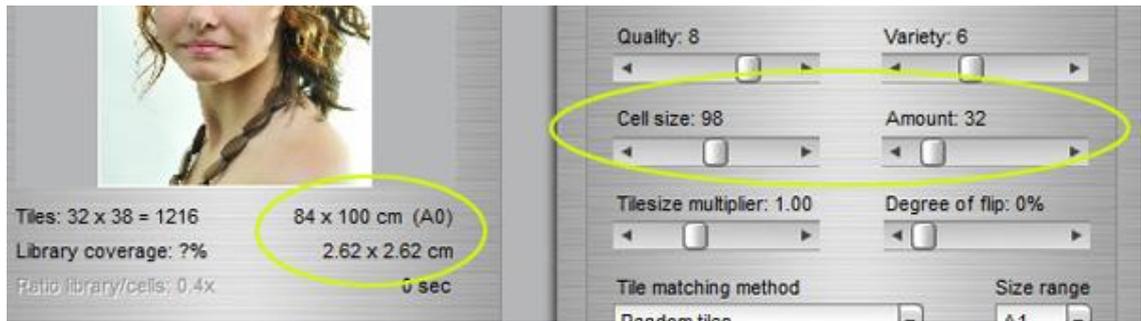


The General settings apply for all the functions. Similar to Mosaic Video (see dedicated chapter on this application), the following settings are available:

- **Create LowRes mosaic for 'Super'**: the super mosaic is automatically initiated when the maximum output size is met. The default is set to 250 MB, a quite safe limit. By default, the super-size mosaic will skip the creation of a lower resolution version (much lower: cell size appr. 16 pixels). To activate this option you need to check this box.
- **Resize quality** : here you can set your personal choice of how bitmaps needs to be resized. The quality can vary largely, depending on the selection. The choices are: Draft, Normal, Smooth, Sharp. The draft settings is slightly faster than the normal. The smooth is the slowest. The picture below shows the differences for three options, enlarged to 400% of the original size to see the effect in detail.

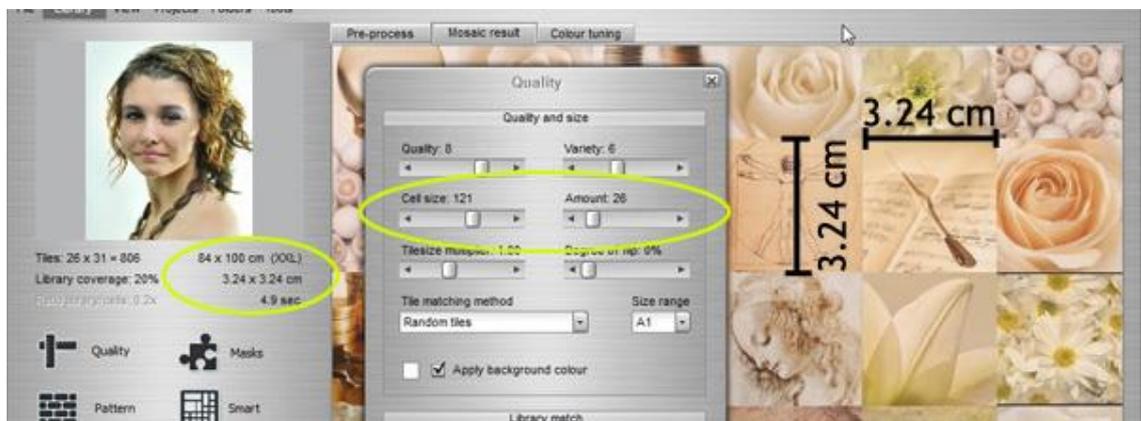


- **Automatic update:** each time the application starts it will check if a new version is available (and will notify if this is the case).
- **Save stone mosaic data:** three files will be generated and saved in the default save location (initial default: 'My pictures'). When the user changes the save location (e.g. by saving a mosaic picture), this default save stone data location is changed as well. The files are: 'Mosaic\_Layout.jpg', 'TileDetails.csv' and 'TileDetails.jpg'.
- **Save mosaic build data:** one additional file is created: 'Tile\_Locations.csv'. This file contains all the locations of each picture. It's basically an excel spreadsheet indicating where a picture can be found in its horizontal and vertical position in the mosaic. It refers the picture name in the library (so: not the full path, just the file name in the picture library).
- **Metric or ANSI:** indicate whether you'd like the GUI data in metric system (cm) or ANSI (inches).
- **Print density (DPI, dots per inch):** this value is used to calculate the actual size of a tile and full mosaic, when printed in this density. It's a bit of an artificial value, since most printers can easily print at 1200 dpi. The value is defaulted to 300, meaning: when one pixels is printed as one drop of ink on a 300 dpi printer, the actual mosaic size will be the value indicated in the GUI. In reality you can safely enlarge the mosaic to a factor 4 or beyond, where pixels do not correspond any more with actual ink droplets. To help you calculate the tile size and mosaic size, the dpi value is largely useful to set, and make sure your intended print size (real world print size) is achieved, where you can now also see how large and actual tile will be when printed in that size. An example: suppose you want to create a mosaic of 100 x 83 cm (A0 size), then you can set the dpi to a value of 96 (the line density of many household monitor screens), and start composing your mosaic. We have selected two possible sizes: 32 tiles of 98 pixels or 26 tiles of 121 pixels. In the latter case the real world tile size will be 3.24 x 3.24 cm. When you now create this mosaic and show the result 100% on your screen you will get a pretty realistic impression of the real world print size. **But foremost: you cannot accurately predict the tile size when you print at an unknown density or in automatic print settings**, e.g. from Windows. You really need to perform proof printing to accurately set the final size for mosaic, and therefore real printed tile size.



Picture above: 32 x 18 tiles, 98 pixels square = 2.62 x 2.62 xm per tile on 95 dpi monitor screen.

Picture below: 26 x 31 tiles, 121 pixels square = 3.24 x 3.24 cm per tile on monitor screen.



## 1.5 Online tutorials

We have a made couple of **tutorials** available on YouTube. The Mosaizer XV tutorials have a sound (i.e. music) channel, and they're subtitled. The Mosaizer Pro tutorials have no sound channel, but are subbed instead. That way you can keep listening to your own music.

Here is a list of the video tutorials for **Mosaizer XV**:

- Install, start, 'Run as Administrator' and how to register the unlock (license) file ([YouTube](#)).
- Basics (10 min): in an uncut 10 minute video we show you how to create a good photo mosaic, with colour corrections. We show how smart patterns are applied, and we use masks to make some nice 3D effect. By using an overlay (mask) we will demonstrate how to make a seamless jigsaw puzzle. It doesn't get much more real than this ([YouTube](#)). The tutorial is subtitled, and has great music (courtesy of Creative Commons), but no voice over. Take your time to see, learn and copy the lessons in that tutorial.
- In the Tools Tutorial (14 min) the most common tools in Mosaizer XV are explained. The tools are: rectangular and non-rectangular picture shapes, edit and replace pictures ('tiles'), the Mosaizer tool, create a new picture library, select and use the different colour matching engines and how to make a lucky shot (random settings). As usual the Tutorial is subtitled, with great free music from Creative Commons ([YouTube](#)).

Here is a list of the video tutorials for **Mosaizer Pro**:

- Lesson 1-3: Use of the graphical user interface, tweak the colour quality and how to use and create picture libraries ([YouTube](#)).
- Lesson 4: Apply regular and non-rectangular patterns, and how to use the smart pattern feature ([YouTube](#)).
- Lesson 5-6: Pre- and post-processing, how to use the Mosaizer tool, how to create single cell mosaics, and how to use the genuine paintbrush ([YouTube](#)).
- Lesson 7A: Apply single and puzzle masks ([YouTube](#)).
- Lesson 7B: Apply multiple and density masks ([YouTube](#)).
- Lesson 8-9: Make a true black and white picture mosaic, and how to effectively apply the right colour matching engine ([YouTube](#)).
- Lesson 10: Single and double layer transparent mosaics ([YouTube](#)).
- Lesson 11-12: Create a super mosaic of 1.8 x 1.4 meter in size, and how to use the chroma key feature to use green screen pictures in a picture mosaic ([YouTube](#)).

### **The Mosaizer promo video**

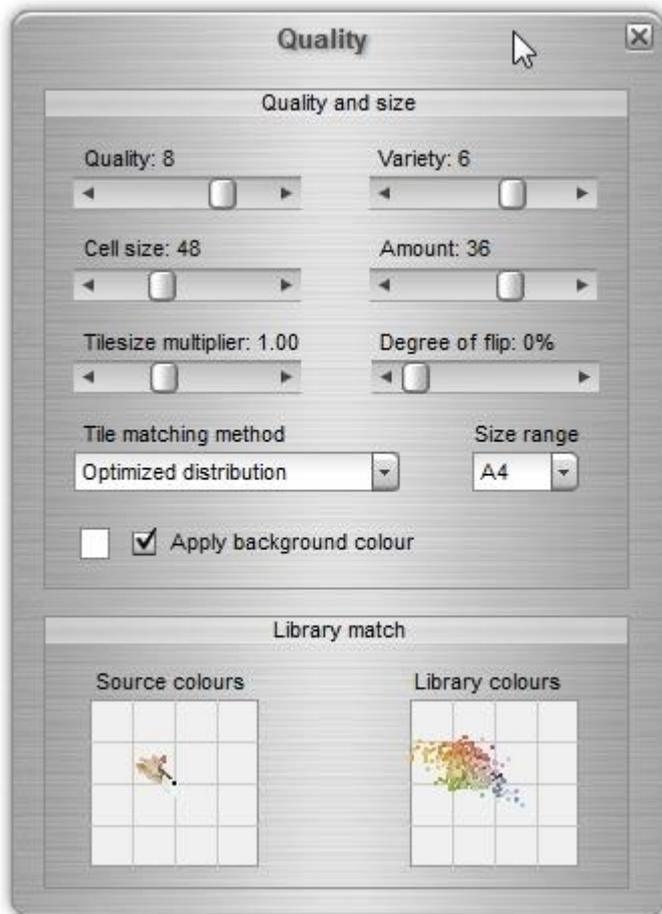
In case you would like to get a 2:44 minute summary of what Mosaizer Pro is capable of... check out [YouTube](#) promotion video.

## **2. The basic XV functions**

There are 6 basic functions in Mosaizer XV. Of all these functions first an explanation is given of the key controls, followed by a description of how to use it, from start to finish (when needed).

### **2.1 Mosaic creation**

With only 4 parameters a perfect but basic photo mosaic can be made. These are:



- **Quality:** the quality of the *colour matching* selection (not of the bitmap output quality). When set to a low value the speed will increase, but the colour match will be lower. When set to a max position the quality is max and the speed is the slowest. A setting of 8 is a pretty good balance between speed and colour match. This is particularly useful for large photo mosaics and for slow systems.
- **Variety:** the distance between two same cells in the final photo mosaic. The (minimum) variety of 2 will result in a very low variation of the selected miniature picture for a certain matching colour, and the result will not look very good. A high variety may result in lower colour match since also the colour match of the optimum miniature tile will become worse. The maximum value of the variety settings increases with increasing paper '[Size range](#)' (and thus with the amount of cells).
- **Cell size:** the size of each miniature picture ('cell' or 'tile') of which the mosaic is built. You should avoid a too small or a too large cell size: too small and the miniature picture becomes unrecognizable. Too large and you may need many cells to recognise the photo mosaic source picture. When the selected cell size is larger than the minimum size of a picture in the miniature picture library, the result may deteriorate due to enlargement of these miniature pictures.
- **Amount:** the amount of cells on the horizontal axis of the source picture. When a portrait source picture (or cropped part) is selected, the amount of cells on the vertical axis will be higher than this number indicates!

#### Four steps to create a picture mosaic

### Step 1: open a source picture

Via **File > Open > <source file name>** select a picture from the list of previously opened files. In case a new picture need to be loaded follow **File > new**. Many different picture formats can be opened by Mosaizer XV. A word of caution: the best way to use Mosaizer XV is to use moderate size source files. When a picture is used from e.g. a digital camera, the pre-processing will be considerably slow due to the large size of such a picture. It is better to prepare a picture upfront and resize this to e.g. 800x600 pixels in size.

### Step 2: read a library of pictures

Via **Library > Open > <library name>** select a miniature picture library from the list. In case a new library is required, follow the steps to create a new library, as explained [here](#). Mosaizer XV reads all data from a picture library when being opened. This data is stored on the hard disk when a new picture library is made. This reading allows therefore a fast switch of picture libraries.

### Step 3: set the required quality and size

Use the settings as described above. In the **'Information'** area a live-update of the photo mosaic size is seen, where the amount of tiles is given and the output size of the photo mosaic picture in megapixels.

### Step 4: press the 'Start' button

After this button is pressed, the full mosaic is generated. It depends on the size of the result and the speed of the computer how long this will take. Small bitmaps (4-10 Mb) usually take a few seconds, also on slower systems. Large bitmaps (100+ Mb) can take up to several minutes. Wait until all actions are finished and look at the results. The result window is automatically activated.

## Cropping

The user can crop a part of the picture. Simply left-click with the mouse in the **'Preprocessing'** tab, and then select (while holding the mouse button down) the area of interest. The selected area will be seen as a rectangular dotted box. All pre-processing filters and grid size are only shown for this selected area.

By selecting a part of the source file, the effect of a selected setting can quickly be checked, before rendering the full picture. This may speed up tuning of the photo mosaic quality and size. By (double) clicking anywhere outside this box, the cropping will be undone. An echo of the cropped area is seen as a red box in the source picture, top-left of the application.

## Saving the result

The resulting mosaic picture can be saved via **'File > Save mosaic'**.

## Printing the result

Mosaizer XV has a simple built-in printing functionality. This can be found via **'File > Print mosaic'**. Mosaizer XV has the possibility to print the result over several pages. The options are:

- Page orientation: Landscape or Portrait.
- Size and fill: **'Page fill'** will fill the best possible fit on the chosen page; if not selected the slider can be used to set the size manually.
- Centre the image: when checked the picture is automatically centred on the page.
- The **nr. of page divisions** will create separate pages for portions of the picture; the max number of pages is 16x16.

If the picture is already saved on the hard disk, Mosaizer XV can't be used to print the photo mosaic. Instead, use a bitmap editor such as Photoshop or Paintshop Pro to print the result. Alternatively, use the file-print functionality of the Windows Explorer itself.

### Advanced controls for tile size and quality

With only 4 parameters a pretty decent photo mosaic can be made. There are three additional controls available:

- **'Tile size multiplier'**: this is the control to put larger tiles on the photo mosaic canvas than indicated by **'size'**, but still on the same tile-grid. As a consequence, each tile will overlap its neighbours. This may not seem very useful for picture photo mosaics, but will become useful when masks are used. Especially brush masks, which have to be relatively small to allow for rotations in the grid, some enlargement may lead to fully filled out tiles on the grid.



- **'Tile matching method'**: there are three ways to select a tile from the picture library, matching a certain colour in the source picture:
  - **Random selection**: a random selection is made from the top matching miniature pictures. The amount of top pictures is indicated with the value of **'Variety'**. For example: when the variety is set to 7, for each tile a random choice is made of the top 7 best matching miniature pictures. This usually leads to a pretty well distributed distribution of tiles when areas of the same colour appear. The **'Library coverage'** shows how much of the full picture library was used in the photo mosaic result. Left: random selection, right: optimized variety.

Tiles: 46 x 34 = 1564	19 x 14 cm (A5)	Tiles: 46 x 34 = 1564	19 x 14 cm (A5)
Library coverage: 21.1%		Library coverage: 27.8%	
Ratio cells/library: 1.1x	1.7 sec	Ratio cells/library: 1.6x	1.5 sec

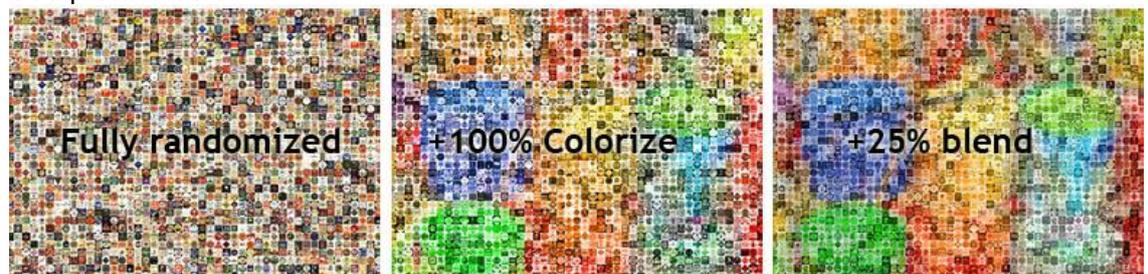
- **Optimized distribution**: an optimization is done over all positions to make sure that the variety is met for all tiles that were initially

randomly selected from the top 'X' matching miniature pictures. Since it's almost impossible to guarantee no same pictures next to each other, this will pretty much repair the areas that have many same pictures next to each other. In general tile optimization **has a downside**: the overall impression may slightly deteriorate because non-optimized cell choices may be forced in favour of a better tile distribution. There is an indicator that shows the amount of tiles in the photo mosaic compared to the number of available pictures in the library: '[ratio cells/library](#)'. This value is dimmed when below 1.

- [Force to use all pictures](#): in case it is necessary to use all pictures in the library, this tile matching method should be selected. Like with the 'Optimized distribution', the indicator '[ratio cells/library](#)' is very useful to see if the colour match will be successful. In this case the amount of cells in the mosaic is lower than the amount of pictures (ratio < 1); there is a 100% certainty that not all pictures will be used. And the result is not really good. However to also get a good colour match *and* use every library picture it is better to work with a ratio of 4-6, meaning: 4 to 6 times more tiles in the photo mosaic than is available in the library. This way the colour match will be much better. In the picture below the effect of this method is shown. From left to right: ratio cells/library = 2.0, 3.0 and 4.0, and the library coverage = 84%, 100% and 100%.



- [Full randomization](#): all tiles are forced to be used in a fully random order. This is not usable to create photo mosaics, however it can be used as an alternative for 'Force to use all pictures' when the colour match quality is not of importance. This setting will typically be used to generate a sort of photo mosaic where absolutely all tiles need to be used, and the 'Force to use all pictures' is not sufficient. It is recommended to use this setting with 100% Colourize combined. See picture below.



- [Black and white optimized](#): this method is only used for greyscale source files and greyscale picture libraries. When used for colours...it may result in unexpected mosaics. There is a dedicated [chapter](#) on this engine.
- [Degree of flip](#): you can indicate to what extent each tile (or cell) picture should be flipped horizontally. In many cases this will expand the possibility to find a better tile picture without adding new pictures to your library. The default is zero (no flipping is done).

## The other controls

The '[Size range](#)' parameter is a practical way to indicate what the output (paper) size of the photo mosaic would be. This setting determines both the range of the cell size and the amount of cells. The extremes are: for A4 cell size is 128 pixels, and amount is 128; for A0 the cell size can become 512 pixels and the amount can be 256 cells on the horizontal. To create **really big** mosaics with **really big** cell sizes, the 'XXL' setting can be used, stretching the cell size to 2432 and the amount to 288. Between XXL and A0 is no intermediate setting available. In case the size gets too big, the super-size feature is activated (automatically).

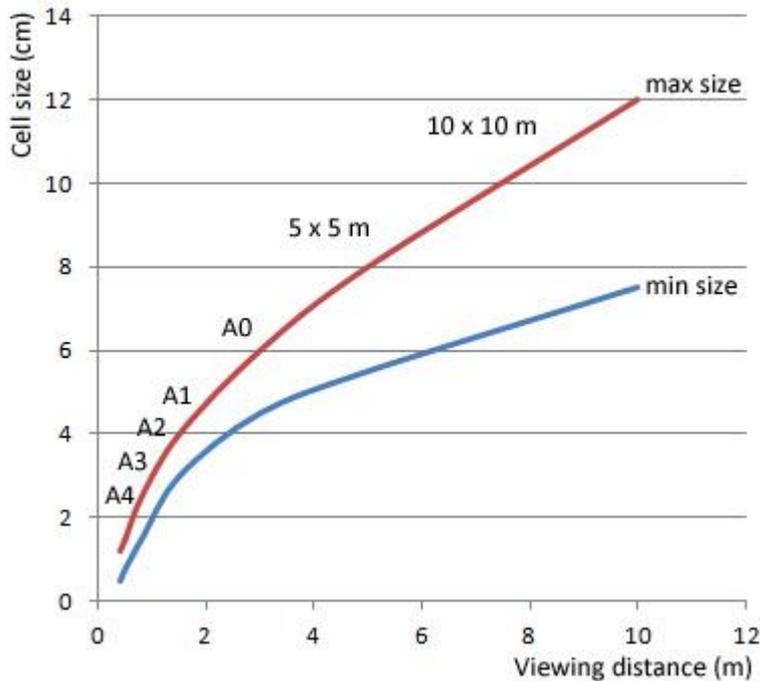
The '[Apply background colour](#)' will set the colour of the mosaic background. In case you have square tiles this hardly has any effect (the full picture will be covered with tiles), but for transparent mosaics, and when masks are used, the background colour may be important to define.

The two pictures at the bottom of this pop-up window show the colour requirements of the source picture (left) and the available colours of the current picture library (right). This is a useful visualization to see what can be expected from the library when using it for that picture. In case of black and white ('Grayscale mosaic') two lines are shown: one representing the amount of tiles of a certain grayscale impression value (any value between 0 and 255) that is required to create the mosaic, while the other represents the available amount of tiles with a certain grayscale impression in the selected library. The more these lines differ in value (vertical axis), the less likely the match will be perfect. And vice versa: the more these two lines get closer to each other, the better the grayscale match of the library and the source image have, resulting in a better grayscale mosaic.

### **Size matters!**

The bitmap size of the mosaic is quite important to manage well! Most users want their mosaic result to be printed, and if possible at the highest print density. The following factors are to be considered:

- *Print density*: an industrial printer for posters (outdoor/indoor) is usually between 100 and 150 dpi or lpi. For photo-quality prints 300 dpi will suffice, and for extremely sharp printing 600 dpi and higher might be required, e.g. for gallery prints and books.
- *Number of tiles (cells) per mosaic*: this value should be at least 24, preferably close to 36 of the shortest side. But the final amount is extremely personal and how much blend and colorize is needed. Our proposed values are just provided for a good start.
- *Size of the tile (cell)*: similar to the number of tiles the tile size is very dependent on the expected viewing distance of the mosaic. A good rule of thumb is 1 cm per meter distance. The closer you get, the smaller the tiles can be to still be recognized. At 60 cm (screen distance) the tiles can be as small as 1x1 cm and still quite OK. The size/viewing distance is typically not constant. Some guidelines are provided in the picture below.



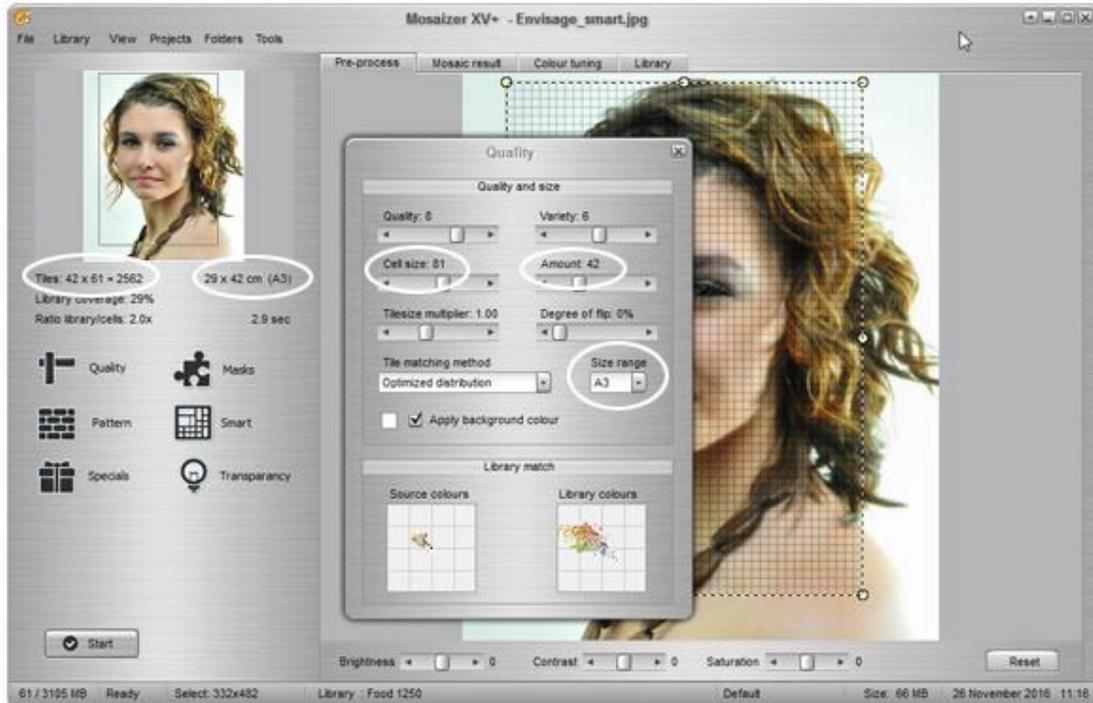
*Graph: the recommended cell sizes, as a function of the viewing distance. The typical paper sizes are also indicated in this graph (as on horizontal axis), from which you can still see the details of each cell picture. All values are based on simple visual evaluations, and have no further value than being a guideline. Personal preferences are always prevailing.*

To create a mosaic that meets the sharpness and viewing distance and tile size requirements you need to do some maths first. And apply a bit of trial and error. The best way is to start with the total result bitmap size numbers using the printer density and print size as initial values.

**Example: make an A3 poster.**

Suppose you want to print an A3 poster (29.7 x 42 cm) at 300 dpi (or 11.69 x 16.53 at 300 dpi), so 3507 x 4959 pixels to completely cover the A3 with a 300 dpi picture. The viewing distance is about 1 meter, so you can probably try 1.0 cm tiles first. That equals 42 x 61 tiles, or  $3507/42$  to  $4959/61 =$  appr. 82 to 81 pixels per cell. Now run the mosaic with these (approximate) numbers and see if the 100% view on the screen is indeed matching the viewing sharpness (printing will get sharper views, since monitors typically only can output at 96 dpi).

The result of this example is shown below. The white circles are the data points that we explained. If the mosaic doesn't look very matching, e.g. too few tiles, then *increase* the number of tiles, while proportionally *decrease* the tile size.



This is just a simple example, and you need to tweak these values a bit, to finally get you where you want to be. As mentioned earlier, the final settings are for your personal judgement and is very hard to *prescribe* in this example. We can only provide some bit of advice on the sequence and basic calculations.

**Quite important need-to-knows:**

1. *The internal dpi value of Mosaizer is 300 dpi. So, in case you want different density values for printing, you need to correct your calculations accordingly, because the output values in the interface are based on 300 dpi. Example: you want to print at 600 dpi, then the indicated size must be divided by a factor 2 (= 300/600) to find the *real* indicated print size.*
2. *The output size can always be different than the ideal target print size. The pixels size can be printed at any format, using output scaling. But to make sure the printer does not interpolate your bitmap you need to calculate the real sizes, and print on a 100% scaling basis.*
3. *The paper size indication in the main interface does not always reflect the actual paper size. In the example above: if you decrease the cell size to 82 it will incorrectly show A2 (here: A3). Although the difference is marginal, yet may cause some raising eyebrows, questioning the internal accuracy ;-)*

**Optional tweaking parameters for mosaic creation**

There are many parameters that can be used to tweak or to create a range of effects to the mosaic, such as:

- random cell position settings
- crop the picture to select only what you want to render
- use of alpha masks, overlays and source masks
- mask rotation and resizing
- pre- and post-processing
- 3D effects
- use the result-file as a new source to create an interesting second effect

- use the transparency options of a source file, e.g. using a .PNG source file

## 2.2 Colour matching

The colour matching controls manage the colour match of the source picture with the library of pictures. In the tab 'Colour tuning' are several controls available, the most important ones are 'Colorizing' and 'Blending', but also 'Textures' are part of the colour tweaking interface.



In the above screenshot of the colour tuning tab three pictures are seen: the top-left is the source picture; the top-right is the resulting mosaic; the bottom right is the result when a certain amount of blend and colorization is applied. This latter picture is instantly updated when the tuning/slider values are changed.

To use the colour tuning effectively you should always render the first mosaic result *without any blend or colorization*. In that case the top-right picture will be a 'raw' mosaic result, which can effectively be tweaked for colour and blending corrections. When you are satisfied with the colour tweaking values in this tab, create a new mosaic, and inspect if the result is what you expected in the first place. Try again (but first run a new uncorrected mosaic first). It's a bit trial and error, but the results are truly great, and the way of working quite structured.

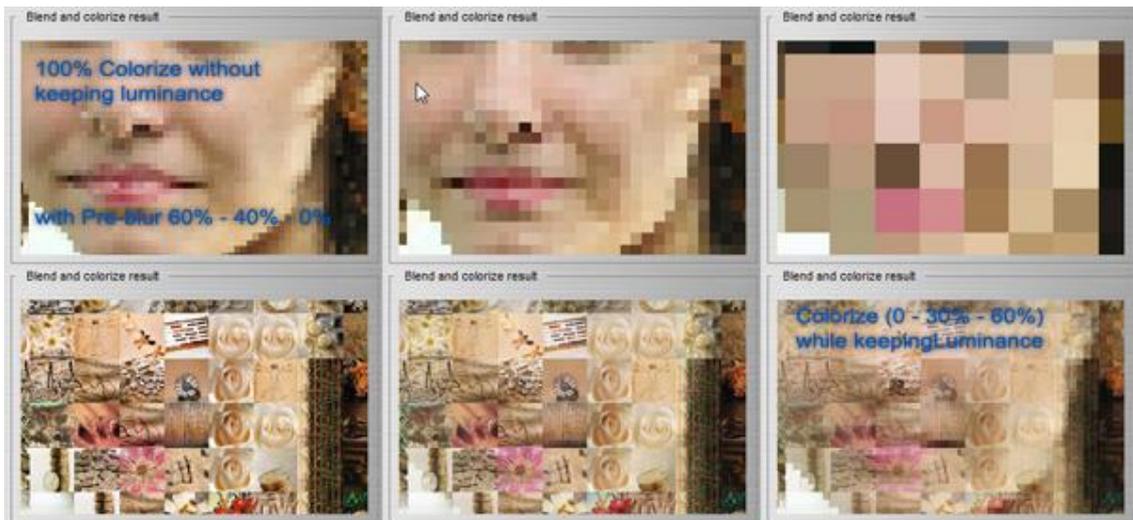
### Colorizing

Colorizing is comparable to the 'photo filter' in Photoshop. It forces the tile picture to take the colour of the source picture (at that position). When the 'Luminance' checkbox is checked only the hues and the saturation values of the picture are forced to the new hue and saturation values, while the lightness values are maintained. If the Luminance is not checked, a 100% colorize value will fully recolour the tile, as one giant pixel.

The Colorize parameter has two additional supporting parameters:

- The **Luminance** (the left checkbox, without a name): when this box is checked the lightness values of the HSL colour of each pixel in the tile are kept untouched (for more information: see HSL colour system, e.g. on Wiki).
- The **Pre-blur** parameter: this parameter will blur the source picture before extracting the hue values for the mosaic result. The lower the pre-blur the more pixelated the colorize effect will become.

In the picture below we show the effect of the colorize parameters in more detail.



The picture above shows two extremes: the top row shows what will be created when 100% colorize is set, and where the pre-blur parameter is varied. The lower row is the result of changing colorize values, but now the luminance is kept. In the lower row the pre-blur was set to 50% (a quite OK-ish value in most cases).

## Blending

Blending is the process to merge the original source picture with the resulting photo mosaic picture. It's not considered 'fair play' to blend the photo mosaic result, but when done in a clever way the effect looks much better, while the source should be not recognised. Especially, when a picture library does not have the full spectrum of colours available, and thus the next best picture is chosen, the result may look 'bad' and blending can be a good colour correction approach.

Similar to colorizing, the blending has a percentage slider to control the amount of blend. Example picture is shown below. In addition to setting the amount of blending, the source picture may also be **Pre-blurred** before blending. This is particularly useful to 'hide' traces of the individual source picture pixels when blowing up in a large photo mosaic and thus create a more natural effect, as if that blend was part of the miniature picture itself.

## Textures

The textures can both add a new 3D 'cloth' effect to your mosaic, but also add the photo-filter effect, not unlike the Photoshop photo-filter. The parameters are quite straightforward:

- **Transparency**: the amount of texture that must be made visible

- **Magnification**: for 'cloth'-like textures (e.g. burlap, jeans) the magnification of the texture can be set, enlarging the characteristic structure, or 'fibre', of the texture. For photo-filters the magnification has no effect.

The picture below shows some examples of applied textures.



## 2.3 Library creation

A picture library is a set of miniature pictures. These pictures are used by Mosaizer XV to build the photo mosaic picture. Mosaizer XV uses a built-in picture library generator to make sure that each library is valid and can be used without error messages. The library file extension is **'\*.lst'** and all library files are stored in the subdirectory **'/Libraries'**. This location cannot be changed by the user.

With the installation of the application comes one picture library; so the user can start immediately. To create a new library, select the Library tab in the interface. If this tab is not visible, make it visible by checking the **'Library management'** in the **'View'** menu. You should now see the following interface:



the same for a library (avoid mixing png transparency and non-transparent .jpg or .bmp files). The current limit of amount of pictures is 100k (hundred thousand).

### Step 2: Check the content of the library

Before continuing to actually create a library, make sure the pictures are large enough, but not too large. The reason to emphasize on the right miniature size is to avoid excessive long rendering times because each miniature pictures needs to be resized when used in the photo mosaic. This may increase the rendering time with a factor 10 or more.

To check the size of the pictures in the folder, press the '[Analyse folder](#)' button. When analysing is done, the summary shows the smallest and largest size of a picture in the selected folder. If too small, you better remove these pictures first before continuing with library creation. When too large, use the '[Resize pictures](#)' option (explained below).

### Step 3: Create the new library

When the button '[Make library](#)' is pressed, the library is created. You are now prompted to choose a library name. We recommend to also adding the amount of pictures in the name, for quick reference of its size. After the new library is created, it will be analyzed on its colours (this may take seconds to minutes, depending on the size and amount of pictures). The new picture library is added to the library selection in the library list. You can now select the library via '[Library](#)' -> '[Open](#)'.

## Resize pictures

In case the miniature pictures are too large it may be useful to copy and resize the entire folder, and save these pictures in another (new) folder on the hard disk. The resizing has two basic controls:

- new picture size, ranging from 48 to 1800 pixels
- new aspect ratio, ranging from 2:3 to 3:2

Use the two sliders to set the desired resize dimensions. The miniature picture may be of help to see what size is currently selected. In case a landscape or portrait size is selected ALL pictures will get this new size. When a photo mosaic picture is made from landscape pictures, but require portrait miniatures, a large part of the miniature will be cut out. It is therefore recommended to only select non-square resize dimensions when this library is intended for similar non-square photo mosaics.

When a resize is selected, also the new folder must be selected, since Mosaizer XV won't overwrite the original pictures of a picture library.

## Library quality check

The application has a built-in security check if the library pictures are of sufficient size. When a library picture on hard disk (ready to be taken into the library) is empty (size = 0 kb) or is smaller than 16 pixels in width or height, this picture will be discarded from the library *and* the user is informed/advised to remove this file from the folder. It is highly encouraged to really remove such a small (or corrupt) picture and avoid error messages during mosaic creation.

The '[Add library](#)' button can show three different icons:

- v: everything's fine. Go ahead with library creation.

- x: something's wrong. You cannot create a library. Possible issues: empty files were found, no files were found at all.
- !: watch out. You have too few pictures selected for a proper library (< 100 is considered a few).

### Some rules of thumb to create a good and versatile library

A prime condition for a good library of pictures is a good colour spread. Usually a large number of files will finally lead to a sufficiently palette of colours. Mosaizer XV is as good as the library that is used. We provide some picture libraries on the website, for a good start. Other useful suggestions:

- Use the Windows Explorer to gather copies of the pictures in a single directory to assure that original pictures are not changed during library creation.
- Preferably use one directory for all files in each library; rename that directory to the intended library name.
- Do not move or alter the library directory or its content, it will lead to error messages when a file is not available during rendering and the library becomes unusable.
- Make sure that the library files are not too large, but merely matches the intended maximum tile size (e.g. 150 x 150 pixels); the larger the library picture difference with the mosaic cell size, the more time is required for cell resizing and build the mosaic picture.
- Only use high quality jpg files because uncompressed bitmap files can take up to 10 times more space.
- Make sure that pictures are not too multi-coloured because this will lead to false colour impressions when no dominant colour is present (as a consequence: holiday pictures are not always a good source for a library).

### Merging two libraries into a new library



You need to select two existing libraries which must be merged, and what new name this merged library must have. Then press the '[Merge library](#)' button and this new library is created. This new library is fully comparable with normal library creation, where after completion this new library is added to the library-list and also made the active library.

### Information about the current library

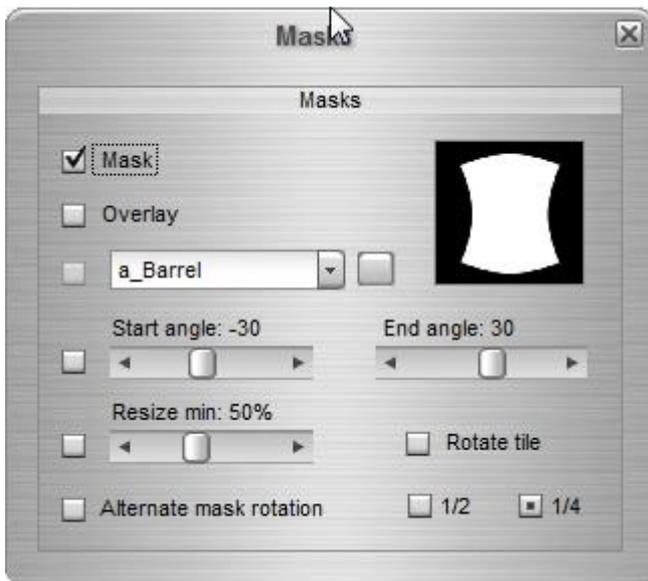
A simple summary of the library is shown when selecting '[Library](#)' -> '[Library info](#)'. From this pop-up summary you can also open the library folder itself.

## 2.4 Masks

A mask is a (small) bitmap that can show or hide portions of the picture cells that will be placed in the photo mosaic. A mask is basically a greyscale bitmap with 256 shades of grey from black to white. All normal masks **must** have the .jpg format to allow for use in the application. The masks that are shown here are the ones that are found in the '/Masks' folder and pre-loaded after starting the application. To easily recognize each mask, all masks usually have a prefix 'a\_' for a normal mask,

and 'b\_' for brush masks. Brush masks are usually fully rotationable, and as a consequence the mask appears to be somewhat small.

The 32Bit overlay and transparent mask must be of .png or .bmp file format to be read by the application at start. Other file formats are not recognized. When new masks are added to the '/Mask' folder, the file prefix 'o32\_' MUST be used to be recognized as transparent masks. If .bmp files are added as transparent masks these must be of 32bit format. If .png format masks are added these must have transparent parts to effectively work as a (32bit) mask. At the end of this chapter it is shown how to make such a mask in Photoshop.



### The 'Masks' button

In this control window all the mask settings are shown. In the drop-down list all the different masks are found, where the selected mask is shown in the thumbnail box.

### The types of mask

#### Alpha mask

An alpha mask is a 24bit bitmap with 256 shades of grey. The alpha masks that are installed in Mosaizer XV have a dimension of 300 x 300 pixels. Black is opaque, white is transparent, all shades in between are degrees of transparency. A good example is the '[a\\_Circular.jpg](#)' mask: the centre is fully opaque (showing the cell picture), fading to transparent (black) towards the edges. As a result a circular picture will be seen on the canvas, fading to the background towards the edges (picture below).



#### 32Bit overlay mask

A 32bit overlay mask is a 24bit bitmap with 256 shades of grey with an additional 8bit alpha channel. This bitmap conveniently combines the alpha mask with an overlay mask. A good example is the '[o32\\_BeerCapsule.bmp](#)' where the 3D effect

is visible in the 24bit portion of the bitmap, and the outline is stored in the alpha channel. This alpha channel only shows what portion of the 24bit part will be transparent and what should be opaque (picture below).



### 32Bit overlay .PNG mask

This type of mask behaves identical like the 32bit overlay mask but it is made from a .PNG file format. PNG is a common standard file type ('Portable Network Graphics') where the bitmap has a transparent channel. It is not entirely the same as a 32bit bitmap (.BMP format). The .PNG file can easily be made with most bitmap editors with layer capability. For example: create a new file in e.g. Photoshop, and transform it into a layer. The layer is not transparent by default, so first delete all pixels in that layer (now the blocked transparency pattern is visible). Then copy or paint anything on that layer, colour or greyscale, while leaving the transparent pixels out. That way the layer remain transparent where the pixels are absent. As a precaution you better select all transparent pixels, and delete these (again if necessary to assure the layer is truly transparent). Finally save that layer in the .PNG file format. And that's basically it.

### Brush mask

This is a mask that acts like an alpha mask, yet is used for a different purpose. Brush masks can be rotated and resized without losing shape or detail in the rotation. Therefore brush masks are rotational symmetric. A good example is the mask '[b\\_Chalk.jpg](#)' where the grey-scaled picture is 'confined' within a virtual circle.



### Mask rotation and resize

Each mask can be rotated and resized. The amount is set by three controls. The angle is set by two sliders, where each slider sets the limit of the rotation (start- and end angle).

- **Start angle:** the starting angle from the vertical axis (negative is rotating to anticlockwise), maximum value 180 degrees.
- **End angle:** the end angle, maximum value 180 degrees. It is recommended to have a lower end angle than the start angle.
- **Resize minimum:** the percentage to which each miniature/mask is resized, where 100% is the original cell size.

When also the box '[Rotate tile](#)' is checked, each miniature picture rotates along with the mask. Below some typical effects are shown, from left to right: rotated mask (-70 to +70 degrees), + rotate tile, + resize mask (50%). Since this example

uses a circular mask, the effect of the rotation can't be seen unless the tile is rotated as well. In the last picture on the right a square mask is used, but this does not rotate very well, since the angles are cut off by the rotation.



The checkbox '[Alternate mask rotation](#)' is required to rotate the mask only on odd numbered tiles. This can be very useful for e.g. puzzle shaped masks. The example below shows the effect, with the notion that the background is set to black, and the '[Tile size multiplier](#)' was set to a value that each mask left a small gap between each tile, showing the black background. Furthermore, to get this effect in place, the number of tiles on the **vertical** axis must be odd numbered. To make sure this is the case you need to change the '[Amount](#)' slider until you have an odd amount of tiles on the vertical axis. This number of tiles is echoed below the thumbnail picture of the source image (top-left).

The two options '[1/2](#)' and '[1/4](#)' indicate how the mask must be rotated: half circle (180 degrees) or a quarter circle (90 degrees). The example below has a '[1/4](#)' rotation and apply the next of the four sides of the puzzle mask.



### Name conventions

Each mask in Mosaizer XV follows a logical name convention. The user is free to give mask any name, but for clarity of intended use, the prefix of each mask is fixed according to the following rules:

- alpha masks must start with [a\\_](#) like 'a\_Circle.jpg'
- 32bit masks must start with [o32\\_](#) like 'o32.BeerCapsule.bmp' or 'o32\_Birds.png'
- brush masks must start with [b\\_](#) like 'b\_Oil.jpg'

### How to make a 32bit .BMP mask

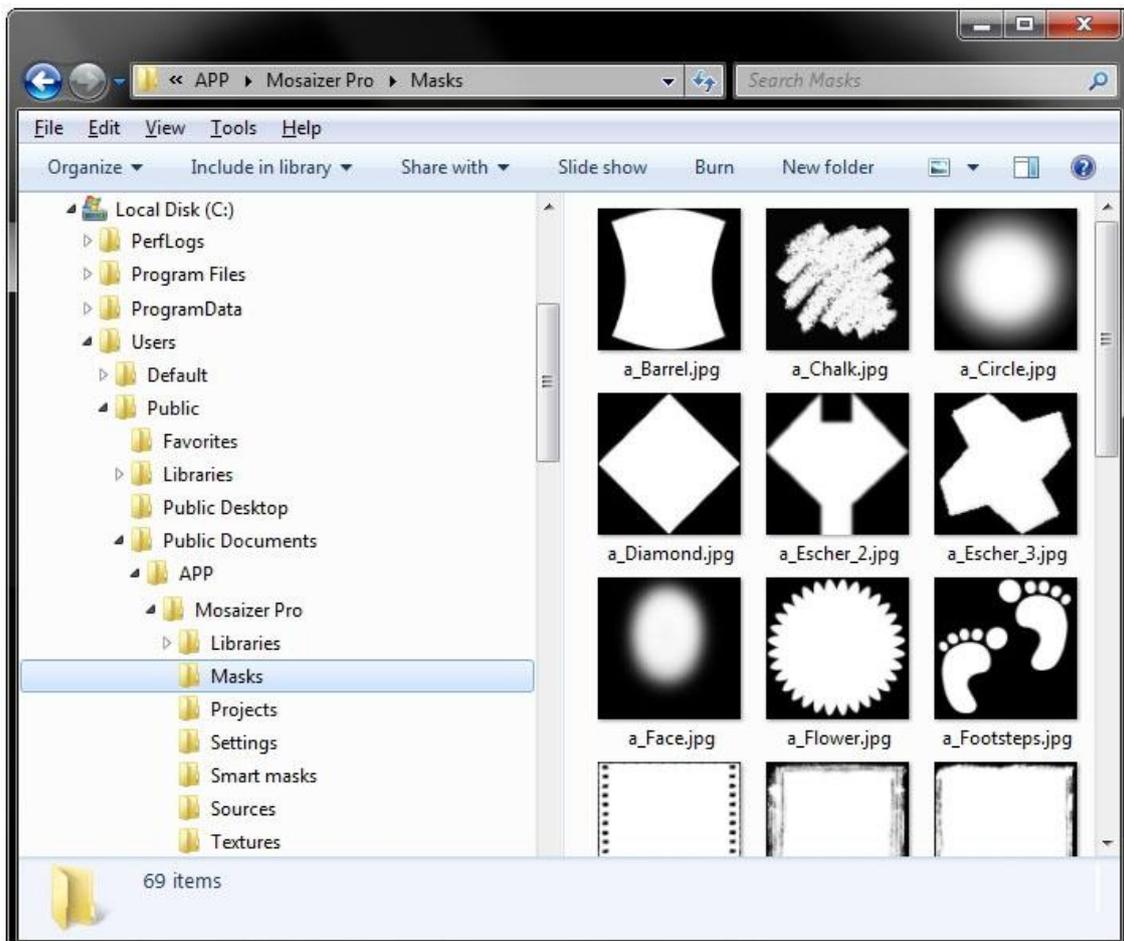
The best way to make an alpha or overlay mask is by using a bitmap editor like Photoshop (Elements) or Paintshop Pro. The alpha, source and overlay masks must be saved as a 24bit greyscale (jpg) image. To retain the quality of the mask when resized during use, the format **must be a high quality jpg format** (less quality will blow up artefacts from the jpg compression), **except for 32bit masks, which must be in the 32bit bitmap format** (\*.bmp extension).

The 32bit overlays require more skills. In Photoshop Elements the following actions would be required to successfully make a 32bit overlay:

- Step 1: start a new picture of e.g. 150x150 pixels.
- Step 2: make your overlay picture (remember: white pixels are fully transparent).
- Step 3: select the pixels that must be left out of the overlay mask, e.g. using the magic wand (contiguous); these are the black pixels in the alpha channel (see picture above).
- Step 4: save this selection via '[Select -> Save Selection](#)', choose a name for the new channel and press OK.
- Step 5: save the bitmap as a 32bit bitmap.
- Step 6: add the 32bit overlay via the windows explorer to the application.

To add masks to the application cannot be done via Mosaizer XV. In previous versions this was possible, but with version 8 the user has to manually copy the mask to the correct folder.

The mask folder is located in [C:\Program Data\APP\Mosaizer XV\Masks](#) (Windows 7/8/10) and [C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Masks](#) (windows XP). Normal (24bit) masks MUST be of type [.jpg](#), and the file name MUST start with '[a\\_](#)' (alpha mask) or '[b\\_](#)' (brush mask). 32bit Masks MUST be of type [.bmp](#) or [.png](#), and the file name MUST start with '[o32\\_](#)'.



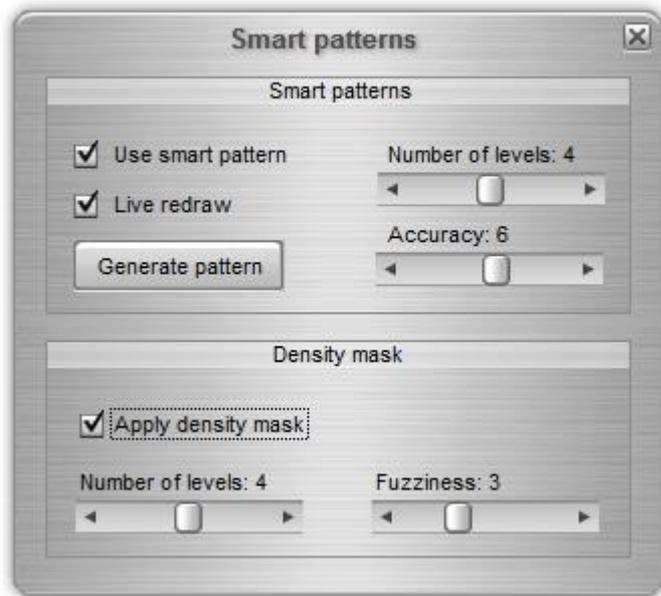
## 2.5 Smart patterns

Smart patterns are rectangular tile patterns but with different tile sizes. These patterns are sometimes referred to as adaptive patterns or even as a 'photo collage'. The size depends on the need to have a smaller tile for that part of the pictures. For instance for areas with large colour changes, or more accurately expressed: changes in hue, saturation and/or luminance. There is a big advantage: to create a good photo mosaic fewer tiles are required and only when details are needed smaller tiles will be used. The overall impression of the mosaic result becomes more brilliant. The (slight) disadvantage is that it takes longer to make a photo mosaic picture. Here is an example of the difference between an 'ordinary' photo mosaic (left) and a mosaic with a smart pattern (right). The details are better and sharper because smaller tile sizes can be used. No blend or colorize was used in both pictures. The full picture is available on our [website](#), where you can zoom in and out to see the details at 1:1.



### Smart pattern controls

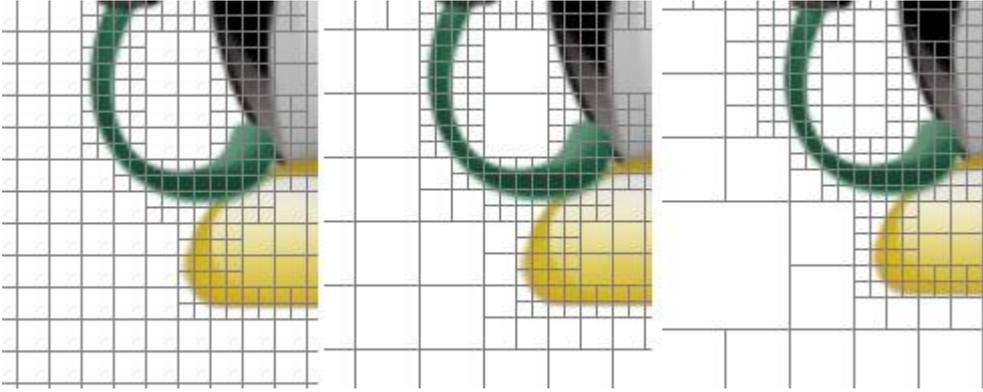
There are two groups of controls: '[Smart patterns](#)' and '[Smart mask](#)'. The first is explained here, the second in the next chapter.



- **Use smart pattern:** activate the smart pattern feature. Mind: some features will be made invisible or disabled while this option is active.
- **Live redraw:** for fast (Intel i5, i7 processor) systems this option should be active. For slower systems the redrawing action may take too much to show a new smart pattern (grid). In that case leave it off, and use the Generate pattern button to manually redraw the new pattern.
- **Number of levels:** the amount of levels in the pattern. In some cases, when the amount of tile is too small, the difference with a higher level won't be visible. Only when the amount of tiles is increased a higher amount of levels will result in a new pattern.
- **Accuracy:** the accuracy of determining when a pattern needs to be made smaller. Low accuracy will result in an almost normal pattern, while a high accuracy will result in a wide range of tile sizes.
- **Density mask:** this is explained a different chapter.

### Number of levels

A smart pattern can exist from 2 to 6 levels in depth. Each level adds another tile of twice the previous level size. Suppose the standard tile size is 32 pixels, then a level 3 pattern will add tiles of 64 and 128 pixels. A larger tile size will be used when the picture allows for areas with less colour dynamics and can be replaced with a larger tile. A level 4 pattern would then also allow for tiles of 256 pixels. We offer 6 levels but 6 levels are hardly required. An excellent effect can already be reached with 3 or maybe 4 levels of tile sizes. In some cases the high levels are not even reached because the picture has no large area to apply an e.g. level 4 tile. The default amount of levels is set to 3. In the picture below from left to right: 2 levels, 3 levels and 4 levels.

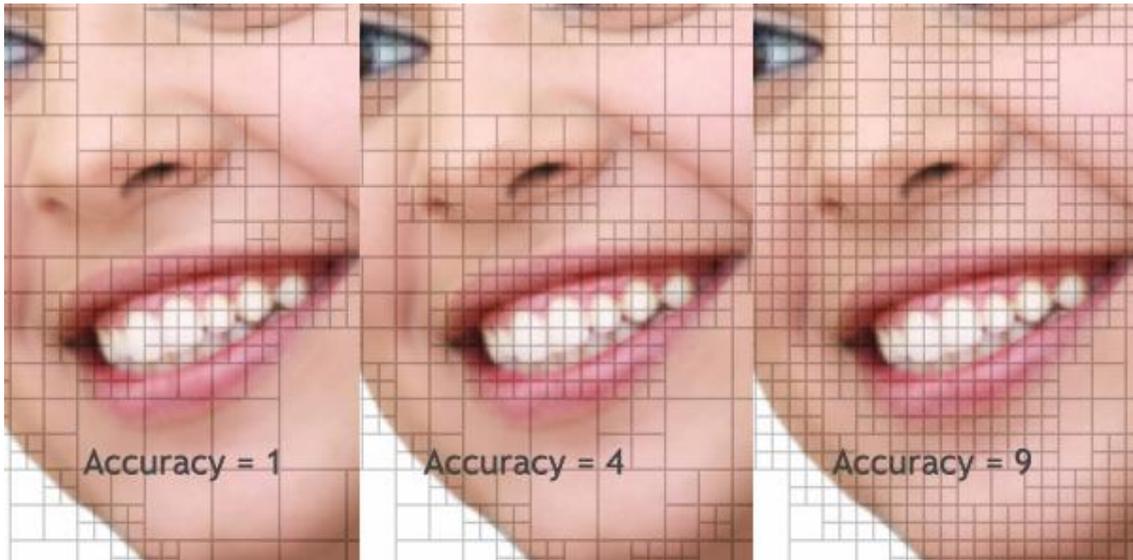


#### How to find a good combination of levels and cell amounts

The source file should have ideal dimensions to work well with smart pattern. This is because if the amount of tiles does not fit in the available width or height, it will discard these tiles. To explain we will apply a pattern of e.g. 24 pixels in cell size, and 4 levels. So the cell sizes in the smart pattern will be 24, 48, 96 and 192 pixels in size. The source file must therefore have dimensions that are multiplications of the largest tile size: 192. A source file of dimensions 576 x 768 is then correct (3\*192 x 4\*192 pixels). The amount of cells on the horizontal is a bit trial and error but for sure it should be an even number. To be precise: the amount of tiles on the horizontal and the vertical should both be dividable by 8 (in the example of 24-48-96-192 pixels, 4 levels) to round numbers, like 32 x 24 tiles (as indicated under the source picture, top left), or 64 x 48 tiles. The amount of tiles gets quickly too high, so a bit or starting low, and see how it looks like is generally a good approach. As a rule of thumb: find the smallest mathematically correct amount (like 32 x 24) and then use multiplications of these amounts. That will always work correctly.

#### Accuracy

The smart pattern is determined allowing for a certain spread of colour dynamics. A setting of low accuracy means that the pattern easily changes to a larger tile: the colour change may be relatively large (= inaccurate) to allow for a cell size increase. An accurate setting means that even for minor colour changes smaller tiles are required. The effect of the accuracy is easily demonstrated when the user slides the accuracy bar from left to right to see the pattern effect. It's a personal choice what accuracy is needed for the end result. The default accuracy is 6 (out of 10). Below an example of three different accuracy values.



#### Additional functionality

The smart pattern is much different from the regular pattern. For that reason not all creativity tools/functions in Mosaizer XV are available for these smart patterns. Some tools are included such as the Mosaizer, textures, cell edit/replacement, colorize and blend, quality, distance and the choices of tile matching method. Tools that are excluded are masks, shade, tile size multiplier, random grid and non-rectangular tiles. In the current version the smart patterns option is also *not* working with the engine '[Black and white optimized](#)'.

#### Cell edit and replacement

Basically, the cell edit and replacement is the same as for the normal patterns. There is one adaptation to allow for cell replacement of larger cell sizes: the cell that is replaced must be selected via the *left-top corner* of that cell. When the user moves in the cell that is larger than the smallest cell size, the selection can only be made using the left-top corner. **NOTE: THIS FEATURE IS TEMPORARY UNAVAILABLE.**

#### From good to great

Several software packages are offered on the internet that does a similar photo mosaic creation. We believe we have a strong solution for the user because of three important reasons:

- Speed: Mosaizer XV has an unrivalled speed in mosaic creation (and where the pattern is generated almost instantaneously).
- Quality: the resulting pattern is creating very sharp and lively photo mosaic results.
- Cost: Mosaizer XV is moderately priced shareware, and even fully functional at no cost during its grace period

## 2.6 Transparency

Preamble: please be aware that this function is a highly professional and complex feature. To work with chroma-keyed pictures requires a deep understanding of how this works, and what effects can be achieved with transparency. It is therefore highly recommended to study and try-out this function prior to real mosaic creation. The manual in this chapter provides a good start to work with this function, but the user might still encounter issues in transparency and dig in the

matter deeper than most functions of Mosaizer XV. Typical issues are: insufficient chroma-key uniformity, green pixels at the edges, less colour matching pictures in the final combination with tiles. Please keep in mind that APP Helmond cannot provide support for unlicensed users.

### What are transparent mosaics?

Creating transparent photo mosaics is a unique and unrivalled possibility of Mosaizer XV. It applies chroma-keying techniques to make a certain colour (range) transparent. This approach is well known in the film industry and for television shows. Although these industries use hardware chroma-keying, we use real-time software chroma-keying algorithms. A number of properties can be set to define the chroma-key for transparency. The proof picture on the right shows which colours will be made transparent. The application preselects a random picture for transparency proof. By clicking on the 'New example' button you can select a new random picture from the current library.

### Transparency settings



The Chroma-key approach in Mosaizer XV follows two different paths:

- Use a single colour, then change the tolerance of that colour in the RGB colour space. Small tolerances will create good results for almost monochrome colour keying. For large colour variations it's not specific enough, and may lead to too many colours washing out. It will work however quite good on black and white.
- Use the HSL colour space. This colour space is quite good in selecting colours with similar hue, luminance and saturation, but is quite sensitive for high or low saturation and luminance of the target colour, such as black and white. The use of the HSL parameters are not always as expected, and requires a bit of patience and trial and error.

## Selected colour approach

To apply the 'Selected colour' approach you need to check the corresponding checkbox. Also activate the transparency by checking the box 'Apply transparency'. To select a transparent colour, simply click on the proof picture on the right. The selected colour will then appear. In this case only the 'Hue tolerance' is of importance, the other parameters do not influence the transparency. When moving the Hue tolerance slider, you will notice that similar colours as the one you selected will now also disappear (become transparent). In many cases you should also check with other proof pictures if the selected colour and tolerance will sufficiently remove that colour. The more variation in the picture library, the more difficult it gets to create full transparency all over the library. It's now the quality of the picture library that determines the result.

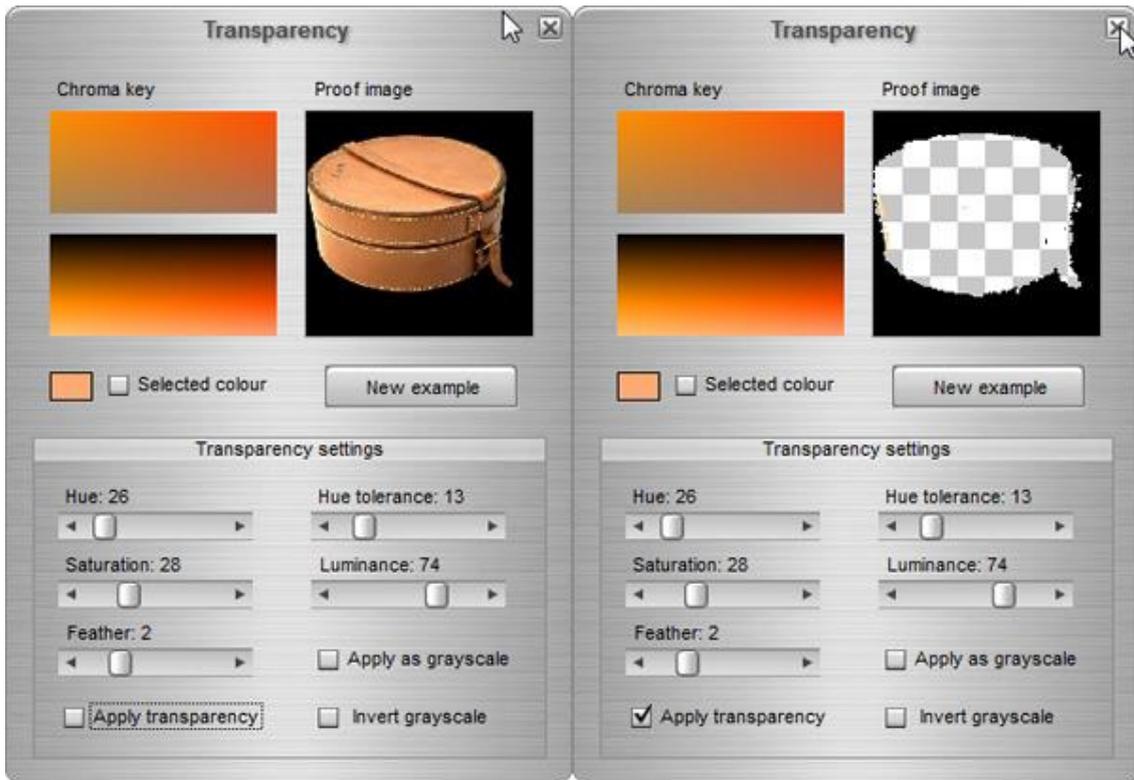
## HSL approach

The HSL colour space consists of three parameters: hue, saturation and luminance. The advantage is that the colour tolerance can now very well be managed, yet the saturation and luminance parameters will also count if a pixel will become transparent. This approach works very well when one specific colour is slightly varied, e.g. in luminance within a picture. In that case the HSL will probably show a quite good transparency. A number of properties can be set to define the chroma-key for HSL transparency:

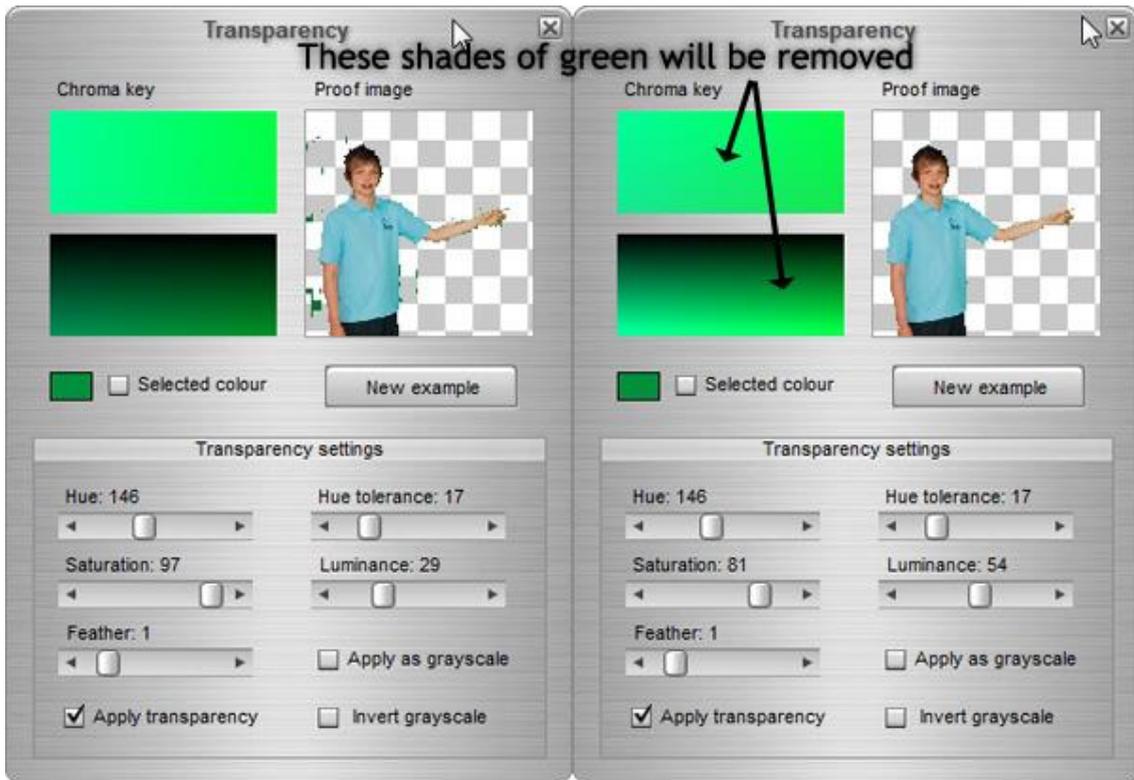
- **Hue:** the initial colour tone that needs to be removed. The hue value varies between 0 and 360° and is cyclic, meaning: the colour tone at 360 is equal to 0.
- **Hue tolerance:** the tolerance of the selected hue value. An increased tolerance will widen the transparent colour (hue) range.
- **Saturation:** the saturation sensitivity of the to-be-made-transparent colours. The less saturation, the more transparent colours will be used.
- **Luminance:** the amount of white and black in the colour tone. A value of almost 0 means: the selected colour looks very dark (almost black), while a luminance of 100 means the colour looks almost white.
- **Feather:** the edges of the transparent parts will smooth out, avoiding hard-edged transparent parts. The result will however look slightly blurred at the transparent edges.
- **Apply as grayscale:** when activated the transparency options only involves grayscales. For transparency hues that is close to black or white this options is quite useful, and in case you use grayscale pictures.
- **Invert grayscale:** for grayscale picture you can either choose to discard the higher luminance parts, or the lower parts. Inverting the range will either make the highlight transparent, or the shadows.

## Some examples of the use of HSL

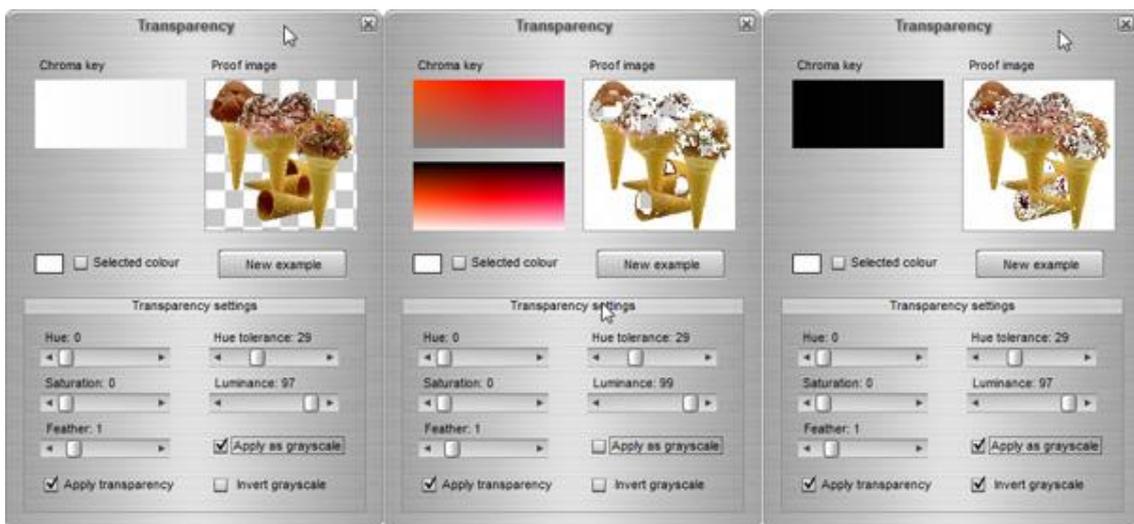
**Example 1:** the skin-coloured box is selected by clicking somewhere on a representative colour. The H, S and L values are copied to the position of the three corresponding sliders (HSL = 26-28-74), and after checking the box 'Apply transparency', the box will almost completely disappear. The feather value of 2 is perhaps too low, and should be a bit higher, to remove the rough edges.



**Example 2:** the initial hue (green) is removed almost instantly, but still a few traces are left behind. Using the 'Selected colour' will not improve, but decreasing the saturation level, while increasing the lightness level will now effectively remove the green around the boy's head. A slight feather is sufficient in this case. As the lower green chroma key picture shows, all colours that are light green to dark green (almost black, but still have a green hue value) will now effectively be removed. And by lowering the saturation levels, also less saturated greens will now become transparent.



**Example 3:** To remove the white background will result in a grayscale approach (suggestion). When clicking on white the application will already assume that the picture is in grayscale, and it will check the box 'Apply as grayscale'. In case you uncheck this box, the red colour is shown as the colour. Which is of course not true, but there is no such thing as a white hue... only a very high luminance; and saturation is not relevant in this case (so: 0). By slightly lowering the luminance it will widen the white areas until the entire white background is made transparent. The effect of 'Invert grayscale' is also demonstrated: in that case it assumes that the white is actually black, and all *non black* parts are now made transparent. Hence the increased transparency of the ice-creams, while the white remains visible. In case the background is black, it's the same approach.



**Example 4:** In the last example we show a complete rendering of a library of transparent tiles. In this case all the white parts have been removed by first selecting the white colour in the proof picture, then adapt some values, especially

the Luminance (=98) and the Feather (=5) values. In addition, we used a random placement, and random grid (Density = 30), and kept the source picture as the background. The result is quite interesting, where the full-size inset shows how accurate the transparency actually works for this library of pictures.

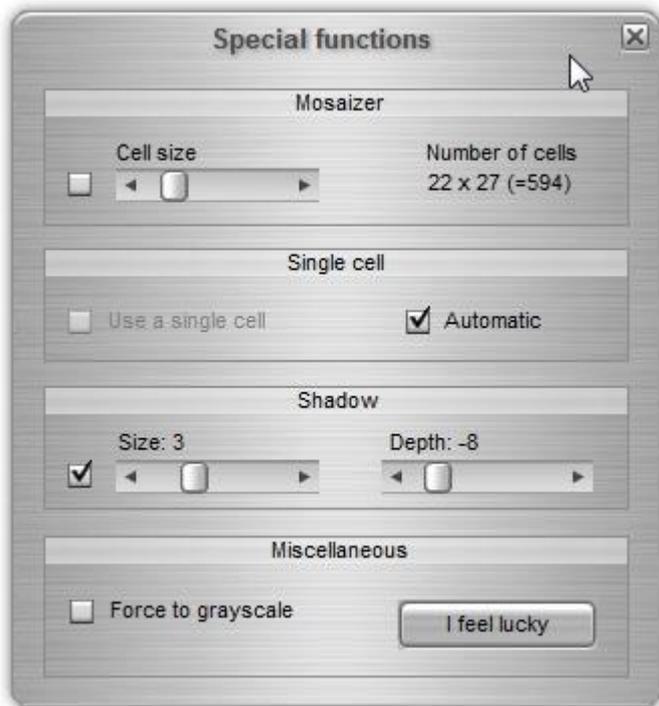


### 3. Special XV functions

#### Contents

These are the additional special functions to get the most out of the application. Some of these tools require advanced hands-on experience, others are easy to use. There are tutorials available on the website that helps to create certain effects, using these special functions.

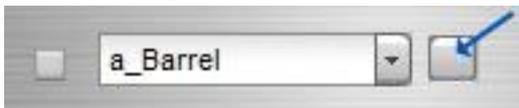
Some functions can be managed by pressing the '[Specials](#)' button:



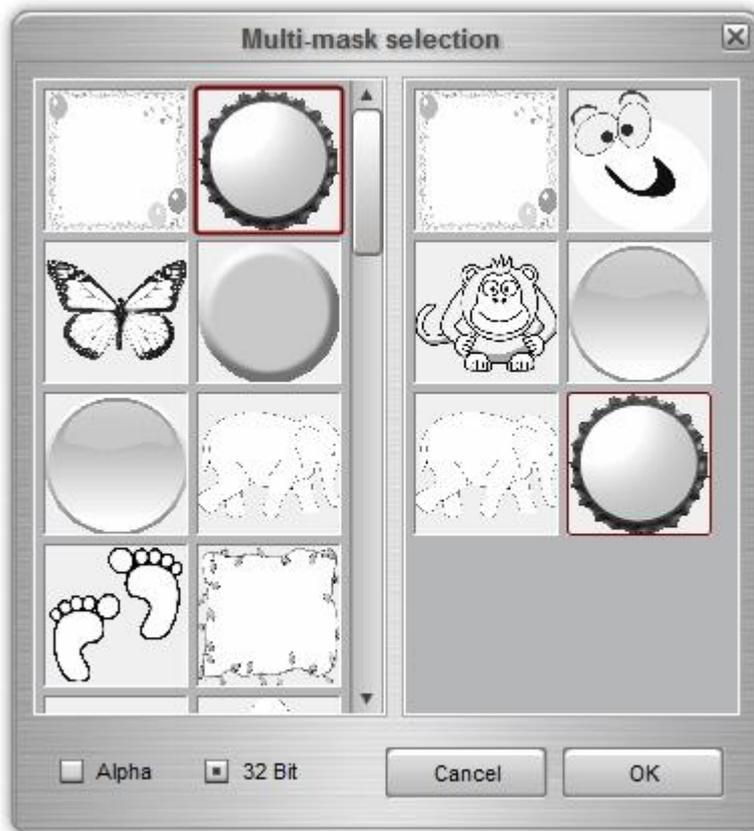
- [The Mosaizer](#). A tool to cut your source picture in parts, and rebuild it again with variations of the cut parts.
- [Single cell](#). Just use one source picture from which you also take the library pictures as a tile, repeated smartly.
- [Shadow](#). Add a naturally looking 3D effect of a shadow/shade to each tile. It also works for masks and overlays.
- [Miscellaneous settings](#). You can force the mosaic to become a grayscale mosaic, and the '[I feel lucky button](#)'. By pressing this button a fully random setting is generated for you. Then, just press the '[Start](#)' button and be amazed.

## 3.1 Multiple masks

The multiple mask feature basically allows for a selection of masks that is applied to the tiles in the photo mosaic. Now each cell can have a different mask. The user can select the multi-mask option by pressing the small button on the right of the mask selection box.



The (blue) button opens a small window in which the user can drag and drop the selection that must be used in the photo mosaic. The (orange) checkbox on the left must be ticked to activate the multiple mask option. If un-checked only the current mask in the mask selection box will be used. Don't forget to also indicate if the mask and/or overlay must be applied (the two checkboxes above the mask selection box).



In the picture above the multi-mask selection window is shown. On the left the whole range of available masks is shown. On the right the selection that must be used. To add masks to the selection you need to drag-and-drop the thumbnail pictures from left to right, and drag-and-drops from right to left to remove the mask from the selection. There must always be one mask available (that mask can't be removed). The sequence of masks is not important. The user can add several same masks to the selection. The more same masks, the higher the probability it will be used.

#### Alpha or 32 bit

Make your choice of mask selections: alpha masks are greyscale masks (file type is .jpg and are starting with 'a\_' and 'b\_'), 32 bit are colour or greyscale masks with an additional transparency channel (file types are .bmp and .png and is starting with 'o32\_'). If the selection is changed the mask selection on the right is lost.

#### Cancel and OK

When the 'Cancel' button is pressed the mask selection is not updated. When pressing 'OK' the mask selection is saved.

## 3.2 3D effect

Mosaizer XV offers a simple 3D effect, applying a shadow under each tile. This effect is best achieved using a mask, preferably an overlay mask (the 'o32\_' mask name prefix). Don't forget to also set the 'Overlay' checkbox when you are using an 'o32' mask.

#### Shadow effect

To activate the shadow effect, press the '[Specials](#)' button. You will now see two slider controls and a checkbox to set the shadow effect. The shadow size determines how wide/broad the shadow must be. A size of two generates a simple black shade around the tile, and for larger values this shade effect will grow, but also gets paler (like in real life).

The second slider determined the position (depth) of the shadow. When its value is positive the shadow moves to the right-bottom corner, when the value is negative the shadow moves towards the top-left corner. The picture below shows two examples of a 3D effect with a mask ('o32\_Beercapsule').



### 3.3 Edit, search and replace

Almost every photo mosaic picture can be edited, where each of the cells can be selected and replaced with other cells. The user fully controls the choice and selection. The cell edit function is located in the control window for '[Colours](#)'.

#### The cell editor



The cell editor is activated as follows:

- Create a photo mosaic picture.
- Select the tab 'Mosaic result' (should be seen after finishing the mosaic picture).
- Press the 'Edit' button (bottom right), the left part of the mosaic result now shows the edit panel.
- Zoom in to 100% exactly (use the '1x' button).
- Press the SHIFT button while moving the mouse over the result mosaic picture. A yellow box appears at the mouse pointer.

In the 'Tile editor' panel on the left the selected tile is reflected, along with twice the amount of top 'X' best matches. In case the 'Variety' was set to 6, the 12 best matches are now seen (from top-left to bottom-right). As explained in the Colour matching chapter, the selected tile is not necessarily the best match because of randomness of tiles. In the picture below, the selected tile is the third best match, out of 6 (so: 12 substitutes are seen). The information in this box is live-updated while the mouse moves with the SHIFT button pressed.

### Replacing a tile

The 'old' tile is the originally placed miniature picture, the 'new' tile is - for the time being - the same. When you have made a choice which tile needs to be replaced, release the SHIFT button and move the mouse to the box 'Best substitutes'. Similar to the photo mosaic tile selection, use the SHIFT button while moving the mouse pointer over the best substitutes and stop at the one that will replace the original tile. The 'new' tile should now be the substitute tile, while the 'old' tile is the tile that needs replacement. The green rectangle is still seen on the resulting photo mosaic picture. Then release the SHIFT button again. All choices are now made and ready for replacement.

**Important to notice:** the edit feature depends on a couple of settings. First, it relies on the current tile placement. All the placement data is stored, and re-used when you want to reset your changes. The new tile is a temporary tile, and is used until you reset or recalculate the entire mosaic. Second, it relies on the current settings for blend and colorize. If you used *active* settings for colorize and blend during initial creation, and then open the edit panel, and start replacing tiles, the *current* settings are used. So, you can change these values while still editing tiles. This way the current tiles with some *original* blend and colorize can be replaced by tiles with a *different* amount of colorize and blend. By keeping the original values of course the *original* amount of blend and colorize are applied. Are you still there?

The actual replacement is managed using one of the three buttons:

- **Replace all:** All cells that match the selected tile are replaced with the substitute picture.
- **Replace one:** Only the selected tile will be replaced with the substitute picture.
- **Reset:** reset the changes. The original mosaic is redrawn, and all changes are lost.

Select your choice and wait until all changes are. This works for all rectangular patterns, but not for the Circulars.

### Replacing a replaced tile

The 'old' tile is only visually replaced. The original data is fully kept intact. When you reselect a replaced tile, you will now see the original tile instead of the already replaced one. This is not an error and it will assure that you can still see how the original version looked like, and it helps to make a new choice for replacement.

### Recalculating with edited values

The replaced tiles are discarded when a new mosaic is rendered with the 'Start' button. The default is therefore 'Recalculate' of the checkbox at the bottom of the edit window. To avoid recalculation, and therefore keep the edited (replaced) tiles, this checkbox must be unchecked. In that case you can now create the same mosaic again, where e.g. the tile size is now larger. In this way you can create super mosaic sizes, while the mosaic is first edited in a smaller version. Only the tile size can be varied this way. All other pattern controls (e.g. amount, randomness) will lead to a full recalculation. Also post processing values (e.g. blend, colorize, texture) can be adapted without recalculation.

## 3.4 Preprocessing

Each source picture can be modified to a certain degree before making a photo mosaic picture. The best way to do this is by using a bitmap editing program like Photoshop or Paintshop Pro, but also Mosaizer XV offers a few simple controls to do some colour tweaking. These controls are found at the bottom of the tab 'Pre-processing'.



Mosaizer provides three pre-processing filters:

- **Brightness:** the overall lightness of the picture is changed
- **Contrast:** the difference between dark and light areas is changed
- **Saturation:** the intensity of the colours is changed

By moving the slider controls the degree of the pre-processing step can be changed, and the effect is seen instantaneously. The 'Reset' button restores the source picture to the default (unfiltered) values. Some examples of a pre-processed picture are shown below.



After the 'Start' button is pressed first the source picture is filtered according to the pre-process filters. The original (source) picture will not be changed during this process. The colour correction is entirely done internally by Mosaizer XV. As can be seen from the example above, filtering only applies to the selected area (see also 'Cropping').

## 3.5 Smart masks

A 'smart mask' is an addition to the smart patterns. With a so-called 'smart mask' you can indicate in which parts of the mosaic smaller tiles have to be used. Where

'Smart patterns' will detect the 'smart' (= size) of the tile depending on the colour changes, with 'smart masks' you can freely set the tile size yourself.

Below you see an example with three pictures: on the left the smart mask; the middle picture shows the corresponding smart pattern and on the right the resulting mosaic.



### The smart mask controls

The smart mask controls are located in the 'Smart patterns' functional button. In addition to a few controls for the smart mask via the functional button also the drawing tab must be made visible in which to actually draw a mask pattern. This tab can be made visible via the main menu 'View' 'Smart mask'. The following controls are available:

- 'Apply a smart mask': this option only becomes enabled when at least one line is drawn in the mask area. When checked the smart mask will be applied in generating the pattern. The smart pattern engine is then used. It's important to *first* activate this option and then use the smart pattern controls to make the pattern, using the drawn mask.
- 'Fuzziness': when a mask is drawn the edges between black and white are sharp. The fuzziness parameter will increasingly smoothen these edges by blurring the mask internally, allowing a smoother transition from large to small tiles.
- 'Number of levels': similar to the number of levels for smart masks. The more levels the smaller the tiles will become in the masked area, and the more difference will exist between the largest and smallest tile. The level will split the tile size in half, so 6 levels means that the largest tile will be  $2^5 = 32$  times bigger than the smallest tile. When insufficient tiles are available the user will notice a warning message in the pattern area to increase the amount of tiles on the horizontal.

The 'smart mask' tab to draw the mask also has a few controls:



- 'Pen', 'Eraser', 'Bucket', 'Undo' and 'Restart' buttons: with these buttons you can draw, erase, fill and undo the mask changes. The draw is a black pen, as is the fill bucket. The erase is a white pen, basically redrawing the black parts. The redo button undo's the last edit change, until the first undo is reached. In that case the mask is complete erased. There are 10 undo actions, after which the first edits are removed. The restart button fully erases the mask.
- 'Pen width': in most cases the pen width is not of big importance since you will use the bucket tool to make a full mask. The pen can however also be used to draw thick lines that won't need filling in with the bucket.

In the picture below you find a simple mask creation: left picture is the outline of the dodo's beak, the right picture shows how the bucket has filled the area. One word of caution: when the black outline is completely closed the bucket will only paint the interior; but in case even 1 pixel is 'leaking', the outside area may get filled too.



When the option '[Apply smart mask](#)' is now activated you should first set the desired depth of level. The best is to also select the '[Pre-process](#)' tab and see how the smart pattern evolves when levels of smart and fuzziness is changed. After a satisfactory pattern is generated, simply press the '[Start](#)' button and watch the levels being created. And that's all there is to do. Then you can tweak the levels, mask and fuzziness as much as needed.

### Save and (re)load smart masks

Smart masks can be saved and reloaded. This way you can restart from a previous session without having to redraw the mask again. Both functions are found via the main menu: '[File](#)' -> '[Save smart mask](#)' and '[Open smart mask](#)'. Make sure the mask has the same width:height ratio as the current source picture because the mask that is loaded will be resized to fit with the current picture.

## 3.6 Non-rectangular shapes

Non-rectangular shapes (or 'Circulars') are special pattern shapes. There are four different patterns: circles, spirals, hexagons and diamonds. These shapes follow a different creation route than the rectangular shapes. Although a hexagonal and diamond pattern is not a real circular pattern, we still refer to it as a 'Circular'.

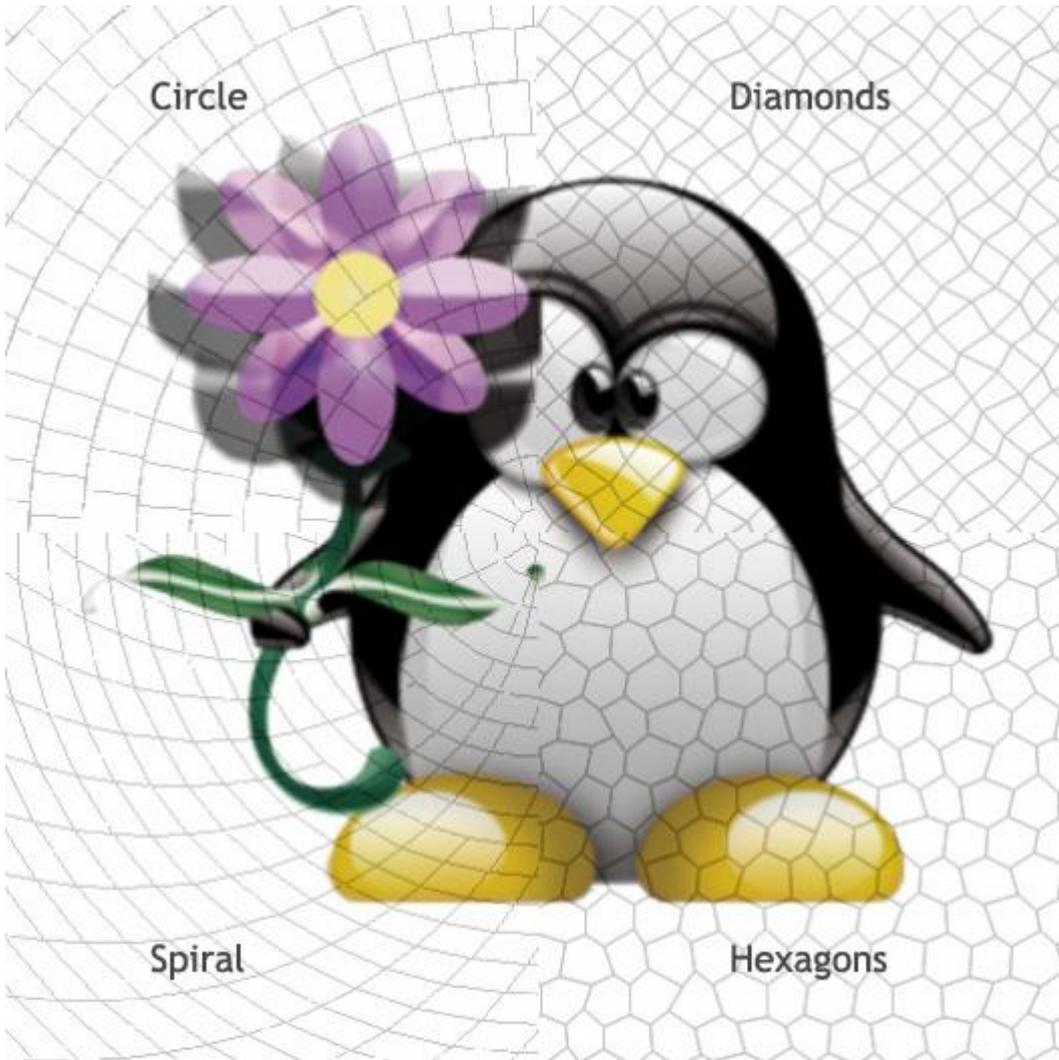
Secondly, there is an option to also make each tile rectangular, and create a brick pattern. Both in horizontal as in vertical direction. Lastly, there is an option to create a slightly random placement grid, also with a certain density. For dense patterns several layers of pictures will be created.

### The Circulars

To start with the Circular patterns, these shapes also have some restrictions:

- Masks do not apply (although 'Colorize' works correctly)
- Not editable (search - replace)
- 3D effects do not work on Circulars

Circulars can be selected via control window '[Patterns](#)'. Only when a circular pattern is selected, its controls become visible. The controls vary per circular pattern.



### Controls for circle patterns

- **Amount:** the amount of circles that is used. There is an upper limit: 100 circles.
- **Density:** the density of each of the circles, populated with tiles, 100% is maximum filling of tiles.
- **Balance:** the balance between the tile size at the circle centre compared with those at the outer rims. In practice this slider will compensate for a high density of tiles at the centre.
- **Shift:** this control acts different than the other three. It only shifts the tiles in the odd circles relative to the even circles. This way a different circular pattern can be achieved. It works best with small amounts. The effect is overruled when another slider is changed, and vice versa.

### Controls for spiral patterns

- **Amount:** the amount of spiral circles that is used. There is an upper limit: 100 circles.
- **Density:** the density of each of the spiralling circles, populated with tiles, 100% is maximum filling of tiles.
- **Balance:** the balance between the tile size at the spiral centre compared with those at the outer rims. In practice this slider will compensate for a high density of tiles at the centre.

- **Angle:** this curvature angle of the spiral. The more curved, the more stretched the outer tiles will become. This can (partly) be compensated with a higher circle density.

### Controls for hexagonal patterns

- **Amount:** the amount of hexagons. The upper limit follows the setting of 'Range' for rectangular patterns (in the tab 'Quality')
- **Wackiness:** sets the degree of deviation from a perfect hexagon. The more whacky, the less the hexagon will resemble a hexagon.
- **Cell size:** the (average) cell size of a hexagon. Like 'Amount', the upper limit follows the setting of 'Range'.
- **Orientation:** the aspect ratio of the hexagon width over its height. A low value results in portrait-like cells, a large value result in landscape-like cells.

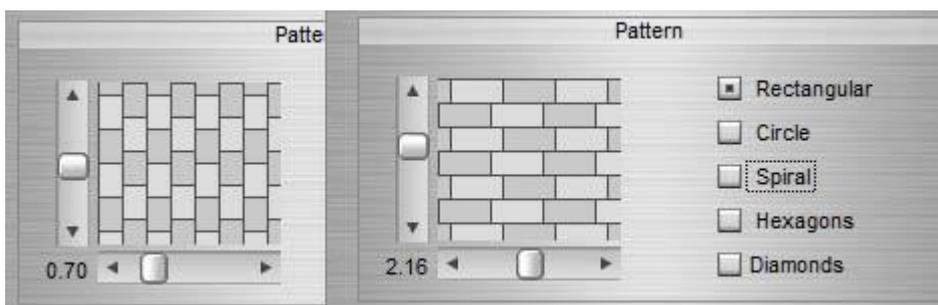
### Controls for diamond patterns

- **Amount:** the amount of diamonds. The upper limit follows the setting of 'Range' for rectangular patterns (in the tab 'Quality')
- **Wackiness:** sets the degree of deviation from a perfect diamond. The more whacky, the less the diamond will resemble a diamond.
- **Cell size:** *this control is located in the 'Quality' tab!* and is the (average) cell size of a diamond. Like 'Amount', the upper limit follows the setting of 'Range'.

### How to set change the position of the centre of the circular pattern?

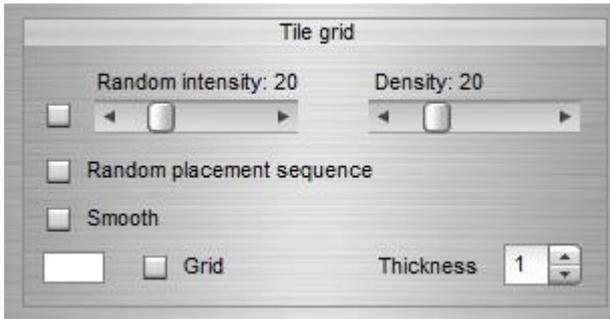
This is done by clicking on the desired position in the Pre-process window. In case also a cropped part needs to be cut-out, this centre will move along with one of the corners of the cropped part. To counter this (unwanted) repositioning of the circle centre, simply click on the picture (but still within the selected rectangle). The position of the mouse pointer will then become the new centre, while the cropped area remains in position.

### The rectangular shape



To change the orientation and shape of each tile, you need to press 'Pattern' button, after which a pop-up window is shown. The two sliders left and bottom of the grid will make your shape non-rectangular. Double-click on the pattern itself will reset the grid again to square tiles. The picture above shows two example shapes: vertical and horizontal bricks.

### The (random) tile grid

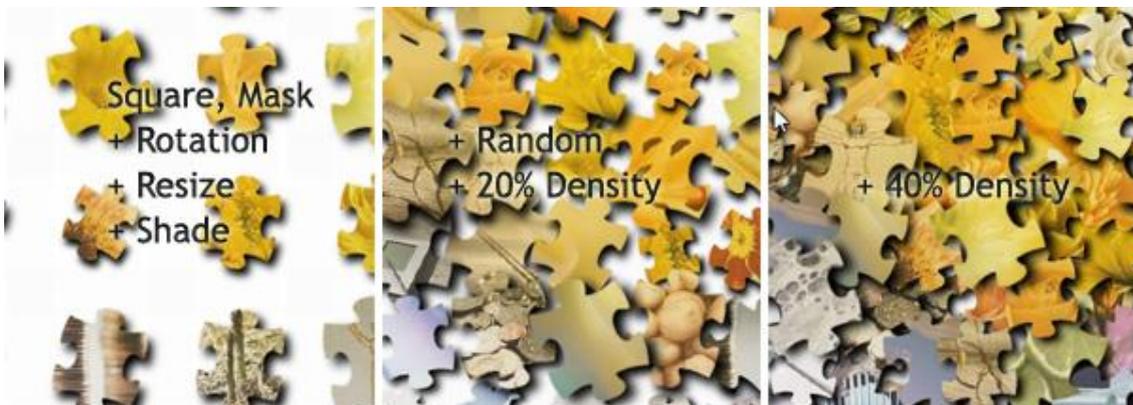


The grid for rectangular pattern is regular. The amount of tiles simply are put in a logical and neat order on the canvas. In some cases it may be interesting to have more loosely positioned tiles on the canvas, creating a natural placement effect. Especially when masks are applied that have fading edges e.g. to create a seamless tapestry of adjacent pictures.

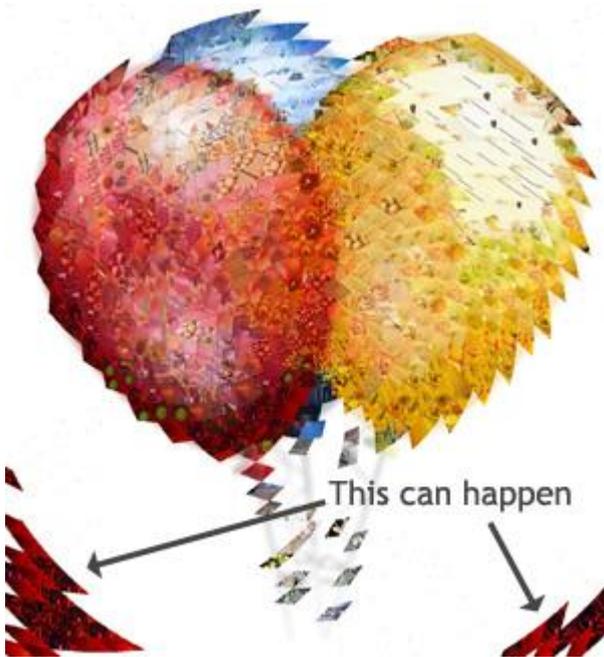
The random grid has a few controls. The random order will only be applied when the checkbox on the left of the 'Random intensity' slider is ticked.

- 'Random intensity': the amount of randomness of the gridpoints. The higher the intensity the more random the grid will become (but never chaotic!)
- 'Density': the amount of increase of additional tiles. When a random grid is used, the distance between adjacent tiles may lead to gaps. When increasing the density, these gaps are bridged by simply putting more tiles on the same area. Consequently, the total amount of tiles will increase and the rendering time will increase accordingly.
- 'Random placement sequence' : when checked the tiles are placed in random order on the canvas. This will assure that the effect will even look more natural than when positioned in a predefined order (i.e. when a library picture is used more than once it will place this tile at all its specific positions, before going to the next library picture). When checked the rendering is slowing down quite considerable (a factor 5-10 slower !).
- 'Smooth (finishing)': the placement steps can be anti-aliased, especially when rotating tiles are used. Activate this option to remove anti-alias artefacts, but it will slow down the mosaic creation a bit. It's typically the last step when a good result is created, where only the anti-alias should finally be added.

Below you find an example of adding random placement, and more density to a mask tile.



**Possible issues with non-rectangular shapes**



In some cases, especially when transparent source files are used (.PNG format), the edges can sometimes be 'forgotten', or jagged. This is a known issue and at the moment we are working to correct it. This is easily fixed by erasing these parts in e.g. Photoshop.

## 3.7 Single-cell mosaics

Single cell mosaics are a special type of photo mosaics: the picture is rebuilt from a smaller version of itself. This feature only works with special source pictures, preferably with little or no colours available. Greyscale pictures are the best source for this feature.



### Settings for a successful single-cell mosaic

The best single-cell mosaic can be made from greyscale images. To activate the single cell, check the tickbox. The other tickbox 'Automatic' is a very useful tool to start with a fully automated and optimized setting to create a good first single-cell mosaic. When unchecked you have to find your way around and tweak and tune the single-cell mosaic. It is highly recommended to tweak and tune the single-cell mosaic with the pre- and post-processing controls. For high values of Colorize the 'Keep luminance' should better be unchecked.

The 'Automatic' sets the following values:

- Blend: 50% (blur before blend)
- Colourize: 80% (keep luminance)
- Brightness (Pre-processing): 10
- Contrast (Pre-processing): 5



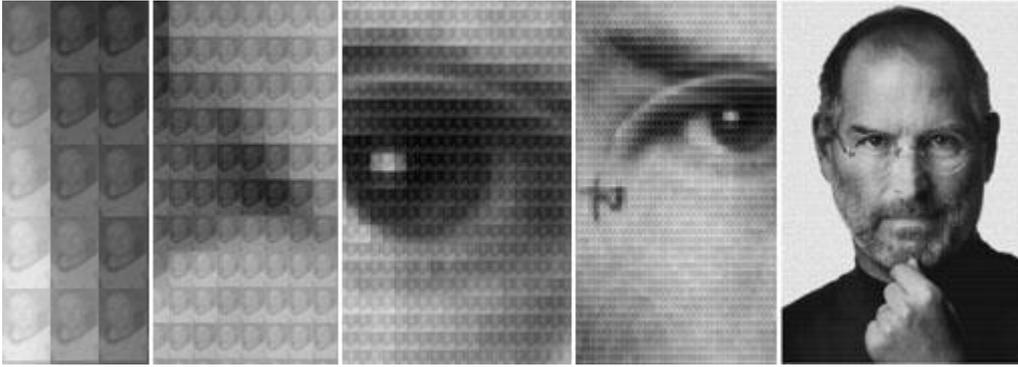
The above picture is the result of an automatic setting, where only the photo mosaic creation has been done twice. So: use the above settings, press **'Start'**, set the Brightness value in Post-processing to 5, and then press **'Start'** again. The pre-processing values are applied twice.

### **A different picture for the single cell**

In some cases it's nice to use a different picture for the single cell. To select such a different picture select **'Single cell picture'** from the menu bar **'Library'**. This will prompt for the location of the picture. When a picture is selected the option is checked (and therefore active). To deactivate the selected (different) single cell picture just click it, and the option will be unchecked. You have to follow the same procedure every time the tickbox is unchecked. Some guidelines:

- Make sure the picture has the same aspect ratio as the original.
- The best effect is made with black and white sources, thus also the alternative single source picture should be the same colour type as the original (either both in colour or both in grayscale, but you are encouraged to try out variants to see what happens).
- The grayscale of the alternative picture may differ from the original (e.g. more whites), so it's a bit of trial and error to find a good setting, better start by using the **'Automatic'** values and tweak from there.

Example: Steve Jobs' famous picture made with Neil Armstrong's famous picture. the picture on the right is actually the resized picture of the original mosaic of 6032 x 8924 pixels, with 250 cells on the horizontal and vertical sides. That is equal to 1.39 x 2.06 m when printed in an industrial 110 dpi printer.



There is one additional settings which may be be useful in this feature: [Force to grayscale](#). When checked the output mosaic will be converted in a grayscale picture. When this checkbox is checked this will also apply to normal mosaics, not only for single cell!

## 3.8 Projects

Projects are small (text) files that are used by the application to quickly regain a favourite setting for each relevant variable: size, pattern, mask, blend, etc. These sets can be created, reloaded and defaulted by the user. Only the sets that are available in the subdirectory `'/Projects'` can be opened in Mosaizer XV.

Each set can be opened or saved via the main menu [Projects -> Open](#), [Save](#) or [Reset to default](#).

### [Open \(existing\) project](#)

At program start, the project files that are available in the subdirectory `'/Projects'` are stored in the menu list. When selected, the application will open the chosen project. The source picture and the mask will however **not** be changed when a project is opened and has to be selected manually.

### [Save project](#)

The current settings are saved in a new project file. The user is prompted to provide a filename. Each set is stored in a predefined subdirectory (`'/Projects'`), which cannot be changed by the user. After save, the project added to the list in the menu.

### [Reset to default](#)

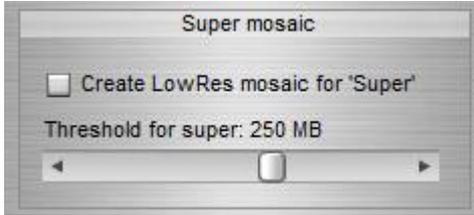
With the installation, a project called `'default.mpr'` is saved in the `'/Projects'` subfolder of the installation folder. This project contains the default values, and can be used to quickly reset all parameters in the application. When this file is not available, nothing happens.

When this file is available when Mosaizer XV is started, it will be opened. The user can create its own default project file, so this way the user can set one's own default values when Mosaizer XV starts.

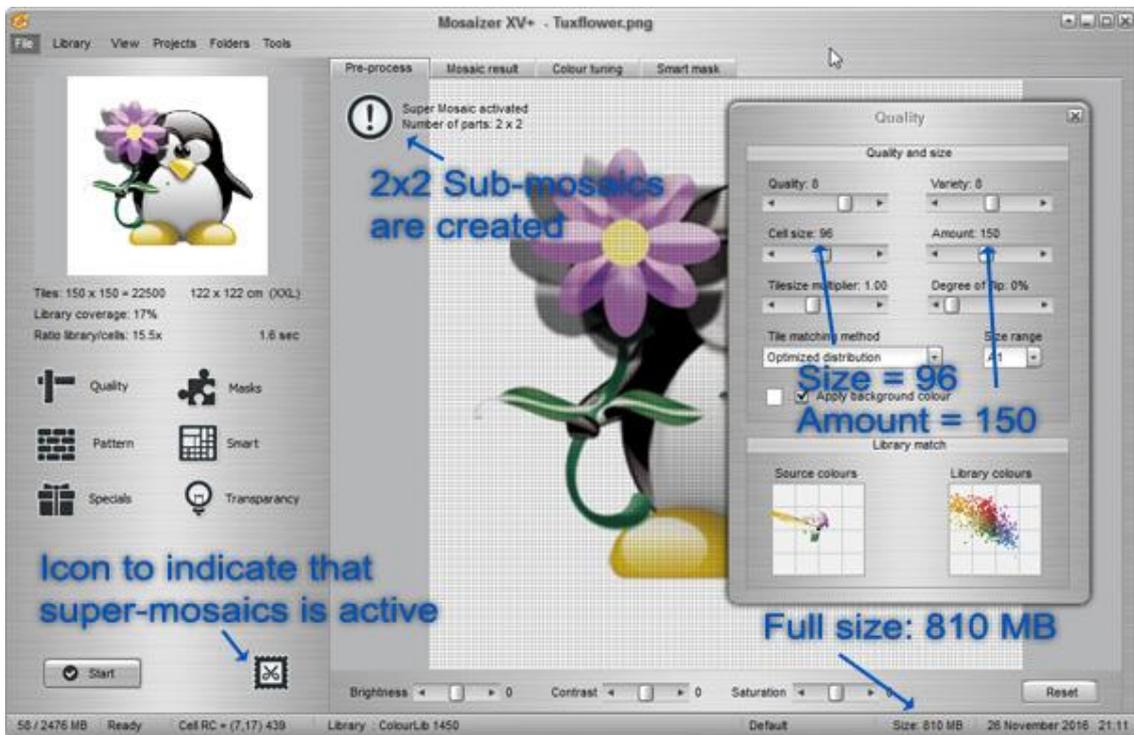
## 3.9 Super mosaic

Super mosaic management are controls to generate a super mosaic: a giant mosaic built from several smaller pieces. This way there is virtually no limit to the size of the photo mosaic. Mosaizer XV is targeted for high quality and speed of

mosaic creation, at the cost of size (in MB). For that reason Mosaizer XV can't create mosaics larger than appr. 350 MB in size (uncompressed bitmap). To overcome that limitation the super mosaic management can be used. The super mosaic basically cuts the photo mosaic in several parts and creates then individual parts of the giant mosaic.



The super mosaic is automatically initiated when the maximum output size is met. The limit is set from the 'Tools'-> 'General settings' pop-up window (see picture above). The default is set to 250 MB, a quite safe limit. By default, the super-size mosaic will skip the creation of a lower resolution version (much lower: cell size appr. 16 pixels). To activate this option you need to check the 'Create LowRes mosaic for 'Super' box.



For instance (see screenshot above): the photo mosaic size settings of 150 tiles on the horizontal, each 96 pixels in size, will result in a bitmap of 810 MB. That is above the 250 MB size limit, and the application proposes to cut the mosaic in 2x2 smaller parts, each about 200 MB.

### How to make a super mosaic?

Basically... there is very little you have to do. When the limit is reached the application will generate smaller parts. Each part is generously overlapping its adjacent parts, so stitching them back together is easy. The individual parts are automatically saved (in the 'Save' folder - to open this folder: menu -> 'Folder' > 'Open save folder'), and take the names of the source picture and an indication of the position, like '1\_1', '1\_2', '2\_1' and '2\_2' suffixes, for this 2x2 example.

You need of course a post-processing application that can handle the giant bitmaps sizes in the first place, like Photoshop. Mosaizer XV has clearly size limits (350 MB is the limit), but Photoshop hardly hasn't.

The automatic super-mosaic options works for all the rectangular mosaics: normal mosaic (patterns or not), and for the smart mosaic. It doesn't work for the 'circulars' (spiral, hexagons, etc). Also the blend and colorize corrections are correctly applied, and also the black and white optimization engine will work in super-size mode.

One word of caution: when pre-blurring before blend and colorize the edges can have a slightly different colour because the blurring edges are treated a bit different. It's best to make sure that the correct stitch part is layered on top of the others. In some cases this is not very convenient, but looking from a distance to the mosaic the effect may be considered negligible.

#### **How to change the number of divisions?**

When you click on the '!' button on top (in case a super mosaic is required), you can manually override the proposed amount of divisions. The application proposes a best fit division over horizontal and vertical directions, but in some cases you might want to change this. Simply modify the values in the popped-up window, and press the ['Apply'](#) button. You need to make sure the total amount of divisions is equal or larger than the required amount.

## 3.10 Grayscale mosaics

### **Greyscale mosaics**

With version 9.2 a dedicated 'colour' matching engine is made available to create greyscale mosaics. This engine evenly distributes all the greyscale library pictures over the greyscale source picture. This greyscale matching approach forces to use all the library pictures with almost the same frequency. This is in contrast to the engine ['Force to use all pictures'](#). With that engine the mosaic integrity still prevails over the library picture frequency and distribution in the mosaic.

The 100% use of all pictures makes this black and white (or grayscale) engine very special. The obvious disadvantage is that the quality of the initial mosaic looks quite disappointing. Some post-processing is needed. Thanks to the fact that we don't have to manage colours, the post-processing blending step is very well applicable, without getting the 'cheating' effect as sometimes seen with colour pictures.

### **Grey quality**

The example below shows the difference between the two engines ['Optimized distribution'](#) and ['Black and white optimized'](#).



From left to right: *Optimized distribution*, *Black and white optimized* and *Black and white optimized + 60% blend*.

There is a striking (and disappointing?) difference between the left and middle picture. Because the greyscale engine will evenly use every library picture, so the source matching integrity will be sacrificed. The result is a not very good looking mosaic, and therefore colorizing and blending is the only way to compensate for this. Since we are dealing with black and white (greyscale) pictures we can blend to pretty high values without the impression of 'cheating'. With colours a blend of 50% will absolutely show the 'cheat' to create a good match. It's no use to apply the option '[Colorize](#)' because that principle of changing the Hue and Saturation of the colour doesn't work in black and white pictures!

Another reason to generously use the '[Blend](#)' option is because the result looks usually more realistic ('softer') and resembles the source picture much better. The left picture is pretty 'hard' where the blended result on the right appears more natural, more 'soft'.

An alternative way to repair the low resemblance of a black and white mosaic is to add an extra layer to the mosaic in Photoshop, using a slightly blurred version of the source file, and apply the 'Soft Light' layer combination mode. When set to '90% Fill', the effect is quite good.

### **Number of library picture repeats**

For coloured mosaics, the details of repeats is sometimes quite visible. Even when '[Optimized distribution](#)' is selected, a frequent use of the same library picture is usually still found. Especially areas with little colour variation are notorious in reusing the same picture frequently. The use of the black and white optimized engine should be restricted to grayscale mosaics, where both a grayscale source picture is used, as of the library of pictures.

The example below shows a clear difference between the two approaches. Left: *Optimized distribution* , Right: *Black and white optimized*. Take a close look at the darker areas for comparison: here you find several repeats of the same tile (left) where on the right *no* repeats are found.



Left: *Optimized distribution* , Right: *Black and white optimized*.

## 3.11 Stone mosaic data

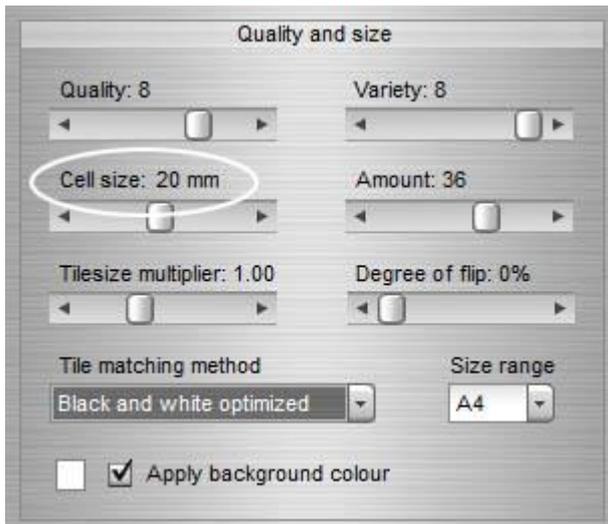
Stone mosaics are already thousands year old.... but still have a vivid community of people who like to use Mosaizer XV to recreate mosaics and build the mosaic with real coloured pieces of stone (and ceramics). The user can activate this option to automatically save the mosaic data to create stone mosaics. This is done by activating the option '[Save stone data](#)' in the '[Tools](#)' -> '[General settings](#)' menu.

Three files will be generated and saved in the default save location (initial default: '[My pictures](#)'). When the user changes the save location (e.g. by saving a mosaic picture), this default save stone data location is changed as well. These are the files:

- [TileDetails.csv](#): this file contains the amount of stone tiles that is used in the mosaic, incl. some basic statistics. The file can be read as a text file in e.g. NotePad, but also as a spreadsheet in Microsoft Excel or Open Office Calc.
- [Mosaic\\_layout.jpg](#): the picture is the visualization of each of the tiles, referenced by a number. The number corresponds with the number and (stone) tile name as summarized in the [TileDetails.csv](#) file and [TileDetails.jpg](#) file. This file also has a printed scale in cm on both horizontal and vertical axes. This picture also shows a 'light' version of each of the coloured stones, as a practical colour reference of the entire scene.
- [TileDetails.jpg](#): the overview of (stone) tiles and corresponding reference numbers.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

When the option is activated the [Cell size](#) format will change from pixels to mm. The range of the cell size is then fixed from 10 to 40 mm. The grout width is fixed at 1 mm. In case you want to increase the grout size, simply add that value to the stone/cell size. You can only select a grout size in millimetres.



For example:

- Your stone size is 20x20 mm and the fixed grout size is 1 mm. The scale on the mosaic layout picture will then calculate  $20 + 1 = 21$  mm for each stone tile.
- Your stone size is 25x25 mm and you want a grout width of 2 mm: select the **Cell size** = 26 mm and since the grout size is fixed at 1 mm the scale on the mosaic layout picture will then be calculated as  $26 + 1 = 27$  mm. Just as you intended.

## 4. Mosaizer Video

The latest version of Mosaizer has two executable applications. One is dedicated for picture mosaics ('MosaizerXV.exe'), the other for video mosaics ('MosaizerVideo.exe'). Both applications are part of the same installer and both use the same unlock file and license files.

In this chapter you will learn about how to work with Mosaizer Video: creating animated mosaic tiles, and creating a mosaic movie from actual video footage. These are the possibilities to create:

1. A mosaic animation. One complete mosaic picture is animated by moving the individual tiles along a certain path. The input is a source picture only (and a library of pictures of course). After completion of the full mosaic, the animation is created.
2. A mosaic movie. Footage of an existing movie is used to recreate that movie where each individual frame is replaced by a mosaic picture.

Most of what has been explained in the Mosaizer XV documentation can also be applied to Mosaizer Video. In these dedicated chapters we only explain the additional features of Mosaizer Video, as compared with the pictures variant, Mosaizer XV.

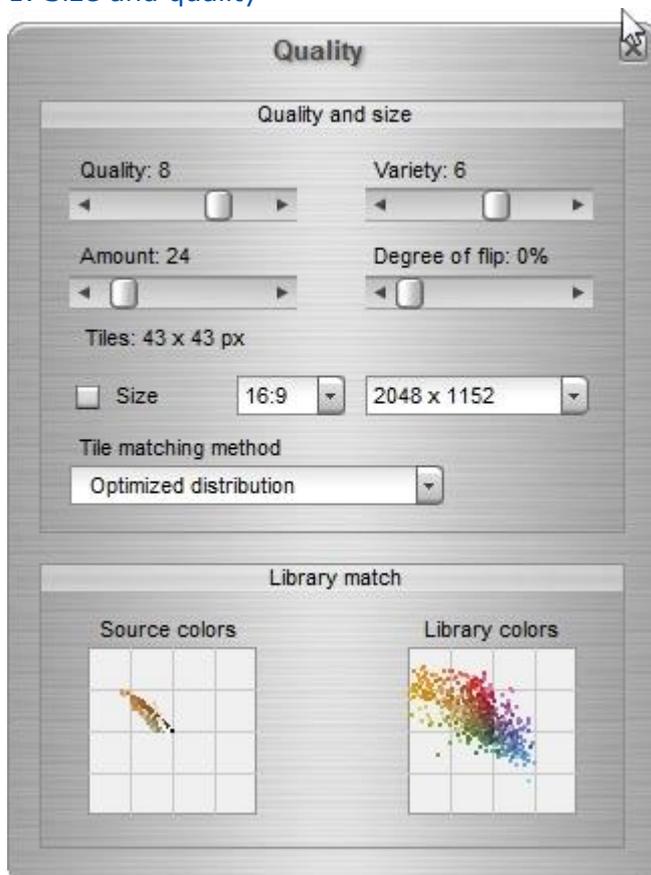
### 4.1 Interface

The graphical user interface (GUI) looks very similar to the Mosaizer XV application. There are many similarities, such as identical look and feel, same buttons and the same way of responding to user interaction (pop-up windows for controls, tabs for details, menu with input/output and tools). There are however a couple of differences:

- There is not a cell size control, only an amount. This is because the movie output size is fixed (drop-down size) and when the amount of cells is set, the size is automatically defined as well.
- There are no fancy features like masks, textures, shade, etc. However, the smart pattern feature is still available for Animation creation.
- Only rectangular cells can be used. No hexagons, spirals, etc. are possible.

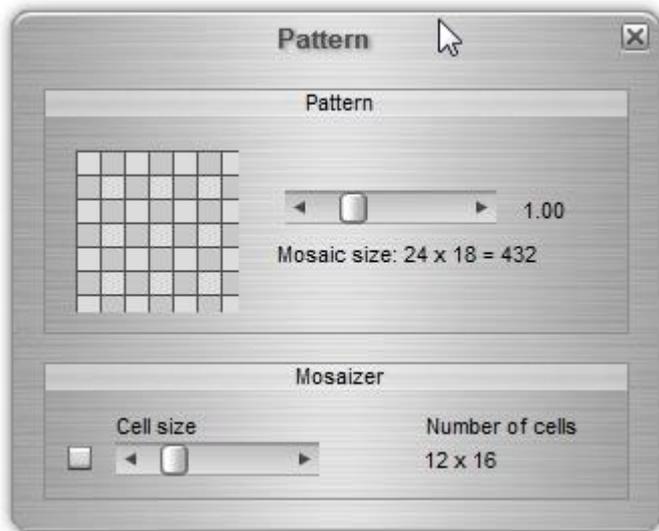
## The six pop-up windows

### 1. Size and quality



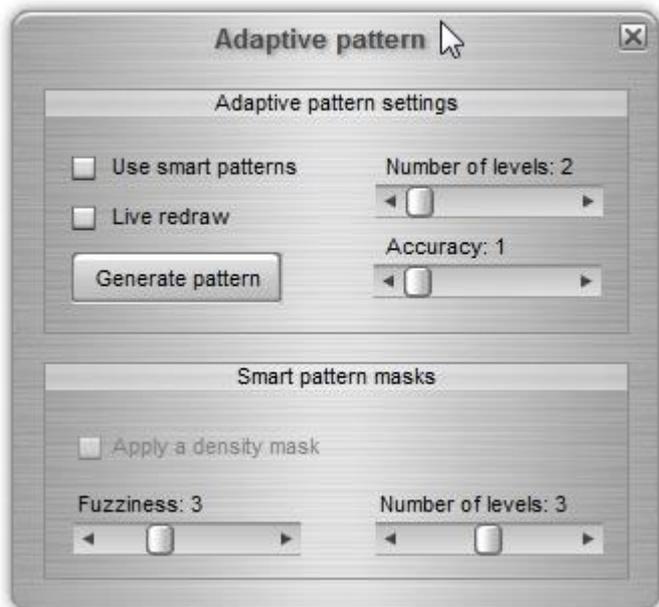
For movies one extra setting is available: the size of the output video. When the 'Size' box is not checked the size of the source video (movie) or picture (animation) is used. In case a typical video size is required, first check the 'Size' box, and select which typical size (aspect ratio) is needed (4:3, 16:9, 16:10 and HD), and then select from the size options your choice of movie size.

### 2. Patterns



For movies the pattern parameter is a simplified version of the Mosaizer XV version. You can only set the shape (rectangular) and if the movie should be a mosaized picture. There is no option for brick-shaped patterns for movies.

### 3. Adaptive (smart) patterns

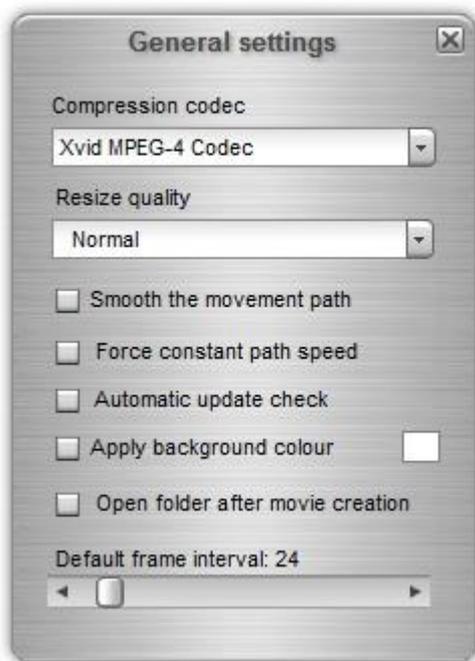


These settings are fully comparable to the XV version. The only difference may be found when a high level is chosen, because the output size will need to match the selected output size. The smart pattern requires tiles that are 2x, 4x, 8x etc. times the basic size, and for high levels, this may not entirely fit the predefined output size. For that reason only 5 size levels are available, but it's still a whopping 64x size difference between the smallest and the largest tile. The smart pattern *only works for animations* and won't work for movies. If that option is active while creating a movie, the smart settings are ignored.

### 4. Animation settings and 5. Movie settings

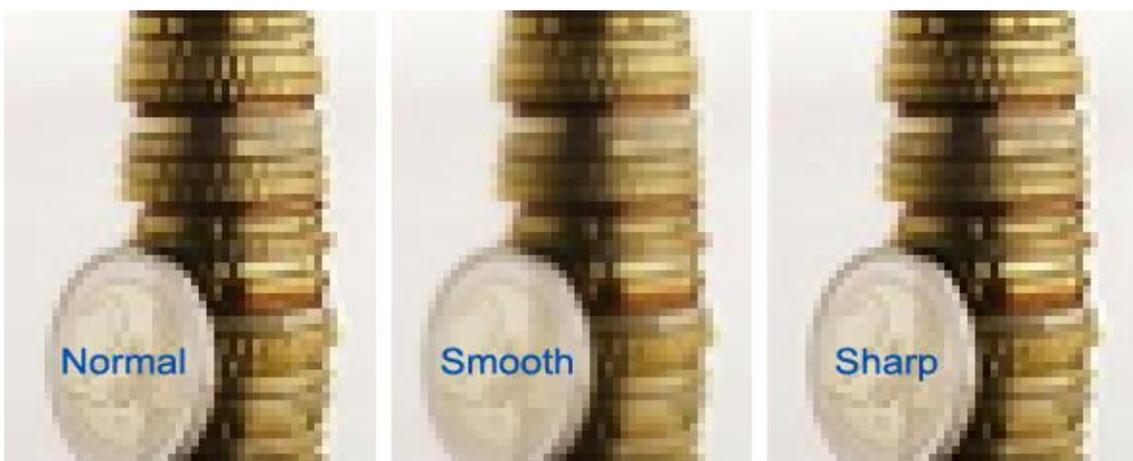
These will be explained in the dedicated chapter on Animations and Movies.

## 6. General settings



The general settings apply for the application itself, and for both animations and movies. These are:

- **Compression codec:** here you will find a list of the available codecs in your system. The selected codec will be used for both Animation and for Movie creation.
- **Resize quality:** here you can set your personal choice of how bitmaps needs to be resized. The quality can vary largely, depending on the selection. The choices are: Draft, Normal, Smooth, Sharp. The draft setting is slightly faster than the normal. The smooth is the slowest. The picture below shows the differences for these four options, enlarged to 400% of the original size to see the effect in detail.



- **Smooth the movement path** (for Movies only): the path is drawn as straight lines between points, but these lines can also be made smooth where the movement is less 'jerky' when passing a new movement path point. More detail is explained in the chapter of [Mosaic movies](#) .

- **Force constant path speed** (for Movies only): the path points per frame are calculated over the length of the path, as will keep the path increment constant over the movie. In this case the frame number of the path point is ignored.
- **Automatic update check**: each time the application starts it will check if a new version is available (and will notify if this is the case).
- **Apply background colour**: instead of using the original source (for Movies: the current frame) the frame is recoloured with a background colour.
- **Open folder after movie completion**: a new Explorer window is opened of the saved folder directory.
- **Default frame interval** (for Movies only): the default frame interval when a new path point is added.

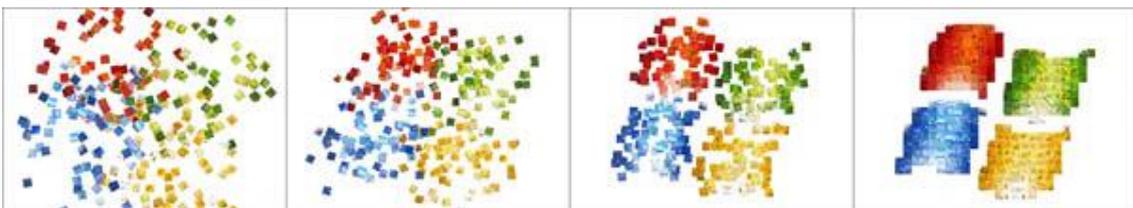
### The compression codec

One word about the '**Compression codec**'. In most cases a simple codec will suffice, like Microsoft Video 1. Experienced users will very probably like to use a more sophisticated codec like XviD, DivX or H.264. Only the codecs that are available in the system will be shown in this drop-down box. Please be aware that *many codecs cannot be used for encoding* but only for *decoding*. Ask for expert information on how codecs work and what codec is a good choice. Codecs that are working in VLC player do *not* work for Mosaizer Video because these are not shared by VLC with the system.

Just recently the X.264 codec was released, but initial tests show that Mosaizer Video didn't always succeed to finish the movie to the last frame. Post-processing with FFmpeg did however do a proper job using the XviD codec in Mosaizer Video. We are still investigating this issue.

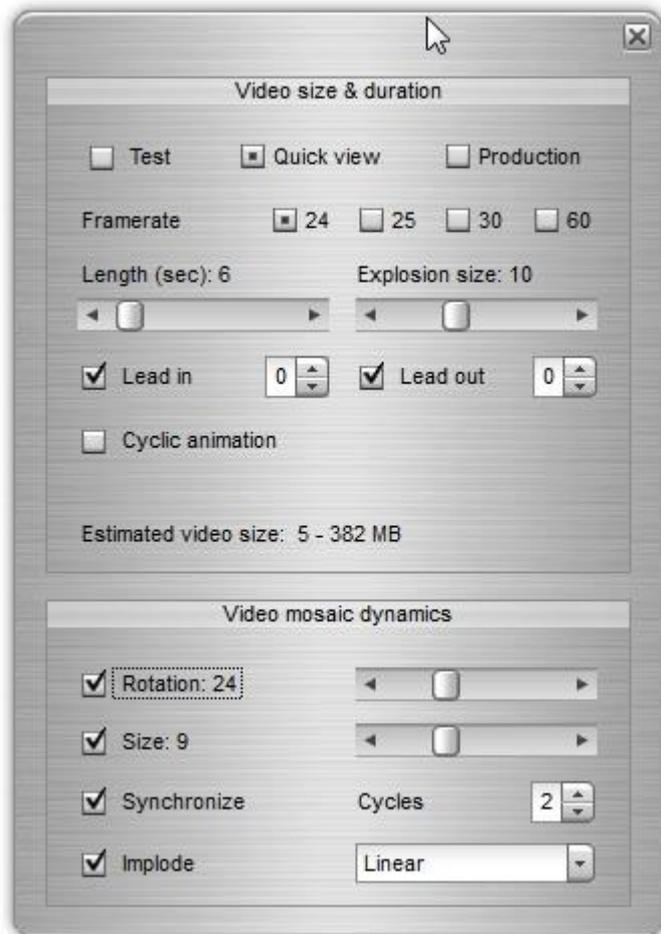
## 4.2 Mosaic animations

This feature will allow you to create a movie from the static (finished) mosaic. This is how it works: at first the full mosaic is calculated and created. Then you need to indicate what path each individual tile will follow to recreate that same mosaic, but now starting from random positions, and dynamically moving towards the final destination, as calculated by the mosaic.



In the picture above you see four screenshots from such a movie. From left to right the chaotic tiles will gradually move towards their correct positions. The creation of such a movie does not require anything else but a finished mosaic. Some features are not implemented in mosaic video: when using masks the individual tiles *cannot rotate nor resize* in the video animations.

### The interface of the movie creator



### Video size and duration

- **Test**: in many cases you only want to see the effect and don't want to wait for all the frames to complete. This option will set a frame rate of 4 frames per second. The result is not shown as a video but as a sequence of bitmaps. The other options are **Quick view** and **Production**. The **Quickview** will provide a very good quality movie, where all the frames are rendered. The **Production** option will however also anti-alias the tiles, resulting in not only a very sharp movie, but also less pixel artefacts because of picture rotations. **Production** will take considerably more time to complete. The framerate for **Test** is 4 fps. The **Quick view** and **Production** follow the indicated framerate (24, 25, 30 or 60 fps).
- **Frame rate**: the number of frames per second in the resulting movie.
- **Length**: the length in seconds of the animation. This is the length without the lead in and lead out.
- **Explosion size**: the starting positions of the tiles in the first frame. A high value will lead to a relative long path from the start to the target end position (= the correct position in the mosaic). The expression 'explosion' is used to indicate that if the mosaic was to explode how far the pieces would fly away.
- **Lead in/out**: the lead in will start the explosion/implosion at a slow pace, and then accelerate to the normal 'speed'. A number of 0.3 x the framerate frames will be added. The lead out will add a number of frames to let the result last a little longer without word animation. A number of 0.8 x framerate frames will be added. The values behind each control indicate the extra number of seconds you would like to add to the lead in (freeze the

first frame) and lead-out (freeze the last frame). Default is 1 second, maximum is 5 seconds.

The actual size of the output video file is set via the [Tools -> General settings](#) pop-up window. This can follow either the size as indicated by the original mosaic (using the '[Amount](#)' and '[Tile size](#)' settings), or can be pre-set using this control. The best reason to use a pre-set is that the file-size should be following certain rules to be played, and the pre-sets follow the usual international formats for video. When a preset is active (checking the box on the left), a rectangular box in the viewer will show which portion of the mosaic can be rendered with the current setting. There are basically two settings to be chosen: the width : height ratio (4:3, 16:9, 16:10, HD) and a size preset with the selected aspect ratio. The HD settings also include 4K sizes.

### Video mosaic dynamics

- [Rotation](#): the amount of tile rotation during the video animation. Range: 0-72 or 0-2 full rotations (= 720 degrees).
- [Size](#): the size of each tile can be enlarged. Range: from 0-30 or from 0-10x the indicated tile size.
- [Cycles](#): the number of rotations and or size 'pulses' per movie. For short movies and high number of cycles the tiles will rotate quite fast, and some motion blur may be experienced. Maximum number of cycles: 6.
- [Synchronize](#): when enlarged or rotated each tile follow a random rhythm of change. When synchronized these pulses are the same for each tile during the animation.
- [Path](#) (in the above interface picture the '[Linear](#)' path is shown): the animation moves randomly starting tiles into the final mosaic. When unchecked the tile paths follow the opposite direction: starting from the intact mosaic each tile explodes towards the outside. Possible paths are: Linear, Spiral, Wave, Curve, Big Bang, Rain, None and Vibrate.
- [Implode](#): the animation can either start from the final mosaic, or end with that mosaic. The implode options will end with the final mosaic.

### Steps to create a mosaic movie

#### Step 1: create a mosaic

Use the normal controls to make a nice mosaic. Make sure you don't use too many tiles or too large tiles, because the initial video movie will be as big as this mosaic in terms of pixels on the horizontal and vertical. In case you want to target for HD quality you can go as high as 2560 x 1440 pixels in size. Higher is considered a super-professional format and requires special systems to show (there aren't yet so many 4K monitors...). Use the '[Make mosaic](#)' button to create the mosaic.

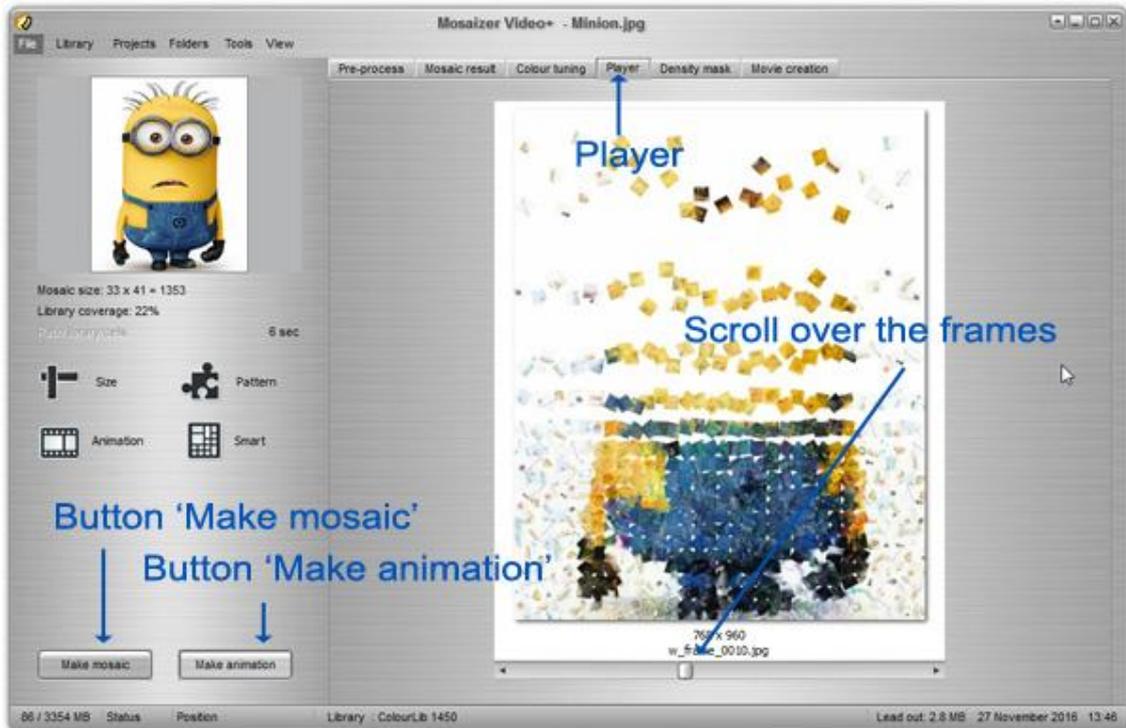
You can easily calculate the tile size = amount / amount size, e.g. a 2560 x 1440 animation with 40 tiles on the horizontal will require  $2560/40 = 64$  pixels tile size. If you would choose 50 tiles, the tile size would have been 51.2 pixels, rounded off to 51 pixels. In that case the movie would be  $51 \times 50 = 2550$  pixels wide. The final (corrective) frame resize action will then resize the frame to meet the target value of 2560 pixels. The advantage of such a last resize correction is a slightly better anti-alias of the edges of each tile.

#### Step 2: set the required mosaic dynamics

Set the animation parameters as you like. It's recommended to first start with small lengths, test a few times and then go to the length settings you target for.

### Step 3: test run your first mosaic video

The checkbox 'Test' should be checked. If all is set, use the 'Make animation' button to create the mosaic, and wait until all the frames are completed. After completion a frame inspector will be made active (picture below). Only a limited number of frames are created, at a frame rate of 4 fps. This seems very low, but it's not intended to make a movie, but to inspect the animation.

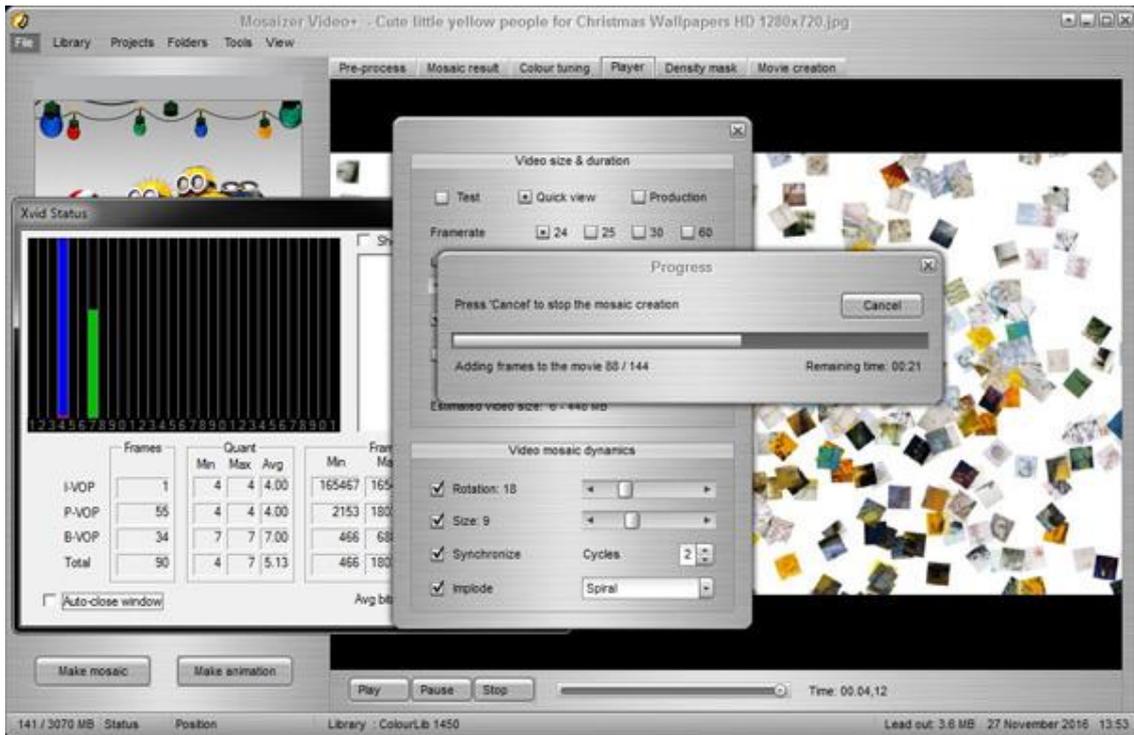


You can either use the scroll-bar at its bottom or first click on the mosaic 'frame' picture, and then use your left and right arrow buttons to scroll through the frames. In case you are not satisfied, set the correct movie values, and start again in test mode. The previous result will be replaced in the frame inspector.

### Step 4: make the final movie

When you truly satisfied with the path, size, rotation etc settings, you can check the 'Quick view' or 'Production' box. Make absolutely sure that the selected codec can *encode* on your system. In doubt always select the 'Microsoft Video 1' codec, it will always work (but has an average quality). The screenshot below shows what you will see when an Xvid code is working properly.

## Mosaizer XV Help



When the codec cannot encode an error message may appear, or the resulting movie file will not be written correctly (and then generate an error when read back into the application). In most cases you will then have to start again (close the application and restart). To counter for problems, you are encouraged to first save the current settings in a project file. This way you can always start where you left last time. This kind of error cannot be managed, nor is generated by the Mosaizer Video application. This kind of error is created by the codec itself.

When the movie is created successfully you will then see the movie player interface (see below). Now you can play, pause and scroll through the movie.



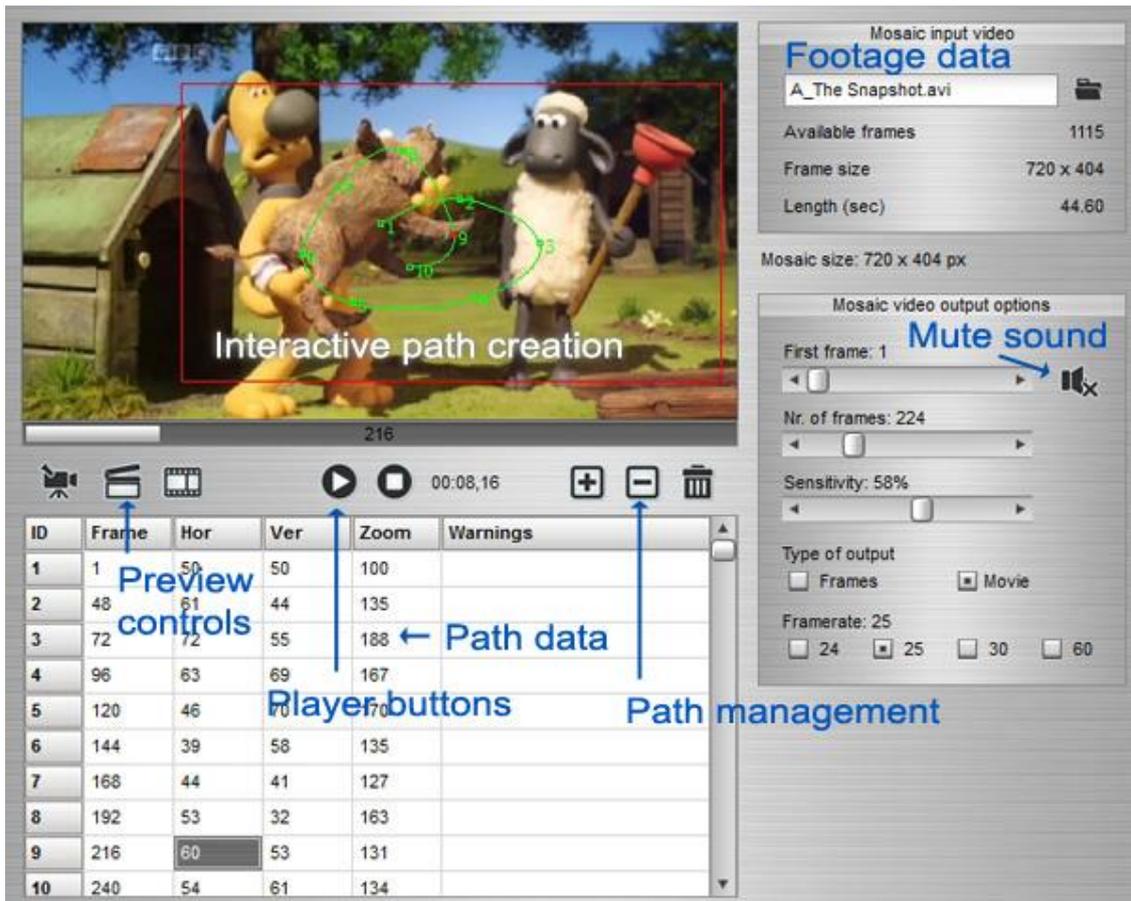
#### Step 5: save or resample the video movie file

The video file is always saved in the default save folder. At start this is the folder 'C:\current user\User Documents\My Pictures'. The most convenient way to locate this file is to use the menu command 'Folders' 'Open save folder'. You will find a file named 'Mosaic\_Result\_01.avi'. When you render a second time the number '\_01' will increase. It's highly recommended to manage the amount of files. With each rendering it will locate the next available number and save the movie under this new name.

## 4.3 Mosaic movies

A much appreciated feature is to create mosaic movies with Mosaizer Video. [Here](#) is an example that we made with the application. This is considered to be a highly professional feature (check out some [Vimeo](#), YouTube movies, or even MTV sources!), and this feature is therefore only accessible for licensed users. For non-licensed users the resulting movie will show a (nag) 'Sample' text. There is no limit to the length and size of the movie. For practical reasons the resulting movie should better be kept small (when rendered to a video file), or otherwise save the individual frames to later create a movie out of these. The user needs to have basic knowledge of digital video management and how to work with video sources, codec, audio and video channels. The resulting video has NO audio channel and it needs to be re-muxed by the user in the post-processing stage (not part of Mosaizer Video).

### The user interface



The interface for creating mosaic movies is roughly split in three parts (see screenshot above):

**Mosaic video input:** the video footage (source) file from which each frame is extracted to create a mosaic.

- The filename of the input file. With the 'explorer' button on the right the source video file is loaded. After successful opening the first frame is shown in the interface on the left, and also replaced as the new source picture for a mosaic.
- Information: the frame size, number of frames and duration are shown after a successful loading of the video file. When strange numbers appear here you need to check if the source file is indeed a valid video source, with a known compression format (the corresponding codec *must* be available in the system to read the source file). The input file does not need to be cinema quality, since each frame will be use for mosaic creation, not for projecting the movie. A frame size of 480x360 would even suffice!

**Mosaic video output options:** the options and format for the output video.

- **First frame:** the first frame that must be used for the mosaic video. The maximum value of the starting frame must not exceed 90% of the total source video length.
- **No. of frames:** the amount of frames that must be rendered by the mosaic engine, and saved to disk as frame of in a full video file. The count starts from the **First frame** that is selected. The number of frames can never exceed the available number of frames. For that reason the max limit of this slider is dynamically changed with changing first frame.

- **Sensitivity:** this indicates at what threshold a new tile must be selected. It depends on the change of colour from frame to frame if a new tile is required. In most cases a value of 60% gives pretty good results. A lower value will generate modest frame changes, a highly sensitivity will probably change all the tiles with every frame.
- **Type of output:** frames or as a video movie. In case **Movie** is selected two additional settings must be selected: frame rate (the found frame rate is already selected), and codec for video compression. The codec is selected from the menu bar **Tools -> General settings**. Only the list of available codecs is shown. Please make sure you select a coded that is also capable to *encode* as well as *decode*. A good codec to try is DivX or Xvid.

**Interactive path creation:** the dynamic path and zoom during the course of the movie.

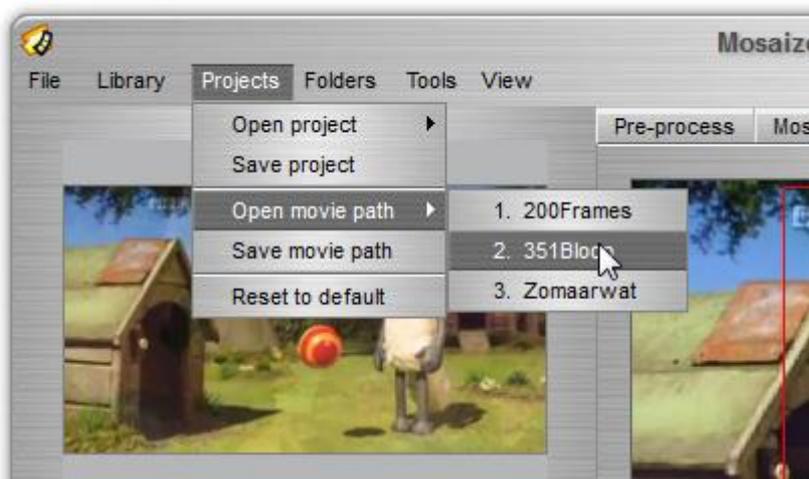
- The top picture has two functions: to create the path and to show the path when the video is showing. The path is created by clicking the mouse at the desired location. As a result a suggested frame and zoom factor is proposed, and the horizontal and vertical coordinates (100% full scale) is shown in the table below the picture. The lime-coloured path is redrawn each time a new point is added.
- The movie controls (see picture below). From left to right:



- - **Preview controls:** Initiate and run the path simulation. You always need to press this button to start a path simulation.
  - Simulate the path using the actual frames, zooming and panning. When clicked the icon changes to a grid icon (on the right). In that case the tile pattern will also be shown in black lines.
  - Show the actual frames, zooming and panning. When clicked the icon changes to an empty rectangle (on the right). In that case zooming is not done, and an animated lime rectangle is zooming and panning to see what part of the footage is used in the actual mosaic movie.
  - **Player buttons:** Play and Stop buttons. The play button will show a pause icon when pressed. Clicking again will resume the path simulation. By pressing the Stop button the animation will stop, and returns the interface back to edit mode. You always need to press this button to keep on editing the path.
  - **Path management:** The + and - buttons are used to add or delete a new path point in an existing path. The selected cell (= entire path data row) will be deleted, shifting the latter frames up. The + button adds a new frame in the current row, shifting the current row and higher frames one row down. You cannot add a new row in the first row, since the first data point must always be at frame 1.
  - The waste basket will delete the entire path, leaving only the first frame path point defined. The default path point is always 50% x and y axis, and 100% zoom. This is identical to 'no path, no panning and no zoom'.
- The values in the table can be edited, by clicking in a cell and change the data. The path in the picture is immediately refreshed. The values are

- percentages of the position relative to its width and height. The 50-50-100 point (frame 1) is always the centre, at no zoom (=100%).
- Each time a new point is created by clicking on the picture a frame and zoom factor is proposed. The zoom factor is calculated by taking the minimum zoom that is needed to avoid a white edge.
  - It is highly recommended to increase the suggested zoom factor with a few points. Also the zoom speed (between two frame values in the table) should at least be twice the number of frames to avoid 'jerky' movements in the mosaic movie. The zoom factor represents the tile size, so a zoom value of 200 means that the tile size will be doubled at that moment.
  - When simulating the video path the sound can be muted. Default is muted, but clicking on the loudspeaker icon will turn the sound on again.
  - The [Warnings](#) field is for convenience, to show warnings when an invalid value is found. The warning is also shown on the right bottom of the interface, with a little more information on the error or invalid value. When this field is showing an error, the simulation cannot be started.
  - Adding points is conveniently done by clicking on the frame. When a different row is selected also the corresponding frame in the footage movie is shown. This way accurate paths are possible. Adding anchor path points is automatically done after clicking on the footage frame. There is no need to press the + button when adding a new point at the end of the sequence. However, it's not possible to add a new path point by typing new values in a new row in the data table. The application only recognizes new path points from clicking on the frame, and from the + and - buttons.

The current path grid can be saved via [Projects >- Save movie path](#). Similarly, the existing paths can also be reloaded in a new session.



### Steps for mosaic movie

#### Step 1: select the video source file

Use the button on the right top to select a video source file. The codec must be recognized by the application. Good codecs are usually Xvid, DivX. The video source file does *not* need to be in hi-res to create a hi-res movie (e.g. 1920x1440 output size). A source footage format of 480 pixels wide is pretty much good enough. When a blend is required some larger sizes may be useful, like 800 pixels width. Large input size will take much more time to process.

#### Step 2: set the desired size

With the tile size and amount sliders in the [Quality](#) pop-up window you can set the initial mosaic tile and amount sizes. The frame output size is also selected in this

window. You can either use the footage dimensions, or select a preset from the two pull-down boxes, e.g. 16:9 and **Frame size: 1280 x 720**. The indicated size will be the final size of the entire movie (or each frame). The selected tile size is the 100% size for the 100% zoom factor. A zoom value of 200 will thus double that size, etc. A zoom value lower than 100 is therefore not allowed.

### Step 3: create a path

Use the mouse to click on a few places in the picture to create a path. Then edit the values, taking the minimum zoom factor into account to avoid white edges in the output file. When no zoom and panning are required, skip this step. There must at least be *one point* available to start a mosaic movie creation. This is the (50,50) point with a 100% zoom (already indicated).

### Step 4: check the dynamics of the path

If a path is created, check the dynamics of the path. Press the **Camera** icon button (left button). Check, correct and check again. Edit, modify etc, the table to add or delete points, or click on the picture to add new points (will be appended to the last known point). Frequently check the path before doing any rendering.

### Step 5: select output type

Make sure you select the correct movie codec. The codec must be capable to *encode* the movie, rather than to *decode* the movie (at playback you decode). When frames are selected the frames will be saved in a sub-folder of the save path (see **Folders -> Open save folder** to detect what folder is used for default save). It's good practice to first save as individual frames and check the frames thoroughly in the windows explorer before to make the final movie. In most cases frames are a favourite choice to allow for post-processing and finalizing the movie for (commercial) purposes. Saving as a movie is useful for an initial assessment of the aspired effects. Also select the frames that need to be rendered: the first frame and the total amount of frames. Don't forget to also set the sensitivity of colour change between frames. A value of 60% is pretty good, but each project can have different wishes for dynamics. The best way is to start small (e.g. 200 frames at the time) and see the effect of the sensitivity setting (save as 'Movie' to effectively see the effect).

### Step 6: render the movie

Then press the **Create movie** button (the general start button on the left). The movie rendering will now start. First each frame is analysed, then each frame is created. Depending on the system each mosaic frame can take between 0.5 and 2 seconds to complete, and is of course heavily depending on the library size and resizing actions. Large video lengths with large sizes (e.g. 4K size), can take several minutes. Depending on the system a 10 second movie at 30 fps will take appr. 5-10 mins on an (Intel) i5 system.

### Some useful tips

Select the correct part from the footage. The application requires a first frame of the mosaic movie, and the amount of frame. Below an example is shown where a starting frame of 1 and 668 is taken, but the same length of the simulation (and movie).

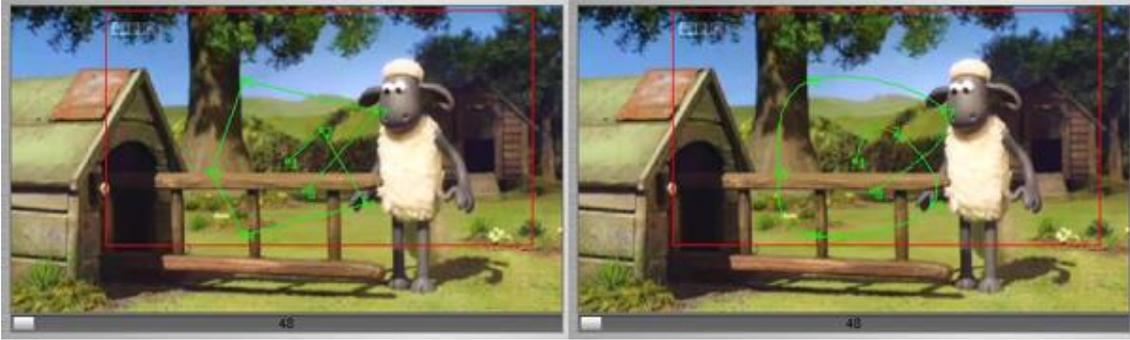
## Mosaizer XV Help



Cropping a movie is needed when the selected output size does not match with the footage dimensions. In the picture below it is demonstrated what will happen when you select a different format than the footage: a cropping area is created of 16:10 size in a 16:9 sized footage.



The path is created by anchor points. The path between each point is basically straight. In case sharp edges (corners) are made to pan left or right, the movement can behave jerky. To avoid jerky panning, the path can also be smoothed. This control is found in the [Tools -> general settings](#). Below an example of how the path looks like in these two cases. One word of caution: when smoothed, there is a risk that the zoom factor gets too small since it may move in a curve and not straight line, exceeding the edges of the footage frame. You need to increase the zoom values in that case.



Below you find an example of an error warning. In this case the zoom factor of data point #5 is too small, and should be at least 167 (in the path data table a factor of 100 was set). A safe value would then be 170 or even higher. The error code is 12 (see list at the end of this chapter).

ID	Frame	Hor	Ver	Zoom	Warnings
1	1	50	50	100	
2	48	61	44	135	
3	72	72	55	188	
4	96	63	69	167	
5	120	46	70	100	Zoom too small (min: 167)
6	144	39	58	135	
7	168	44	41	127	
8	192	53	32	163	
9	216	60	53	131	
10	240	54	61	134	

Type of output  
 Frames  Movie

Framerate: 25  
 24  25  30  60


 Warning: zoom factor too low to fit within the movie boundaries

### Error codes for mosaic video

The application checks if the path is defined in the correct way. It will check several situations that might result in a possible error or faulty mosaic frame. These are the codes that are generated:

1. Frame number cannot be negative
2. Horizontal position cannot be negative
3. Vertical position cannot be negative
4. Horizontal position cannot be larger than 100%
5. Vertical position cannot be larger than 100%
6. Same frame numbers found
7. Incorrect frame sequence
8. Zoom factor is not allowed for your type of license
9. Frame number exceeds available frames
10. First frame must always be equal to '1'
11. Zoom factor cannot be lower than 100%
12. Zoom factor too low to fit within the movie boundaries
13. Missing data in table

## 5. Tools

### Contents

This chapter describes some tools that may help the user to keep the session data ('Activity log'), and to explain about the terminology that is used in this application ('Glossary').

### Language selection

There are currently a couple of languages available for Mosaizer XV: English, Dutch, French and German are pretty well translated. The language is selected via 'Tools' -> 'Language', and then the language of choice. Only the language files that are available in the folder '/Languages' in the installation directory are loaded. These files must have the extension '.lan' to be recognised by Mosaizer XV. Anyone who can help us to translate the software needs to contact us. Please send us an e-mail and we'll contact you how to translate the software interface.

### Automated web-update

This version of Mosaizer XV is pretty new, small bugs may still occur once in a while, or customers notify us of an issue. A manual update can be done from the menu bar 'Tools'-> 'Web update' . The automated update is activated via 'Tools' -> 'General settings', and then check the checkbox 'Automatic update' .

## 5.1 Glossary

### Cell or Tile

A cell (or photo-tile) is a bitmap from the library of pictures that replaces a part of the source bitmap. When a photo mosaic bitmap is generated, the source picture is divided into small pieces, called 'cells'. You can set the number of cells in horizontal and vertical direction. Each cell is replaced by a photo (bitmap) that resembles the colour impression the best.

### Photo mosaic painting

A technique adopted and refined by APP Helmond to 'paint' photo's on a canvas from a library of photo's (bitmaps) that matches the intended colour of a source picture. This technique is also known as photo mosaic creation or Photo-Tiling, patented by Robert Silvers in 2000 (United States Patent 6549679). However, evidence of early photo mosaics is found back in 1993. Since the creation of Photo-Tiles is not protected, but the publishing and printing is, it's very difficult to assess the infringement when Mosaizer XV is used.

### Canvas

This is a professional term of the basis of the photo-mosaic result bitmap. It could be interpreted as a virtual canvas of a painter, ready to be painted.

### Alpha masks

Alpha masks are greyscale 24bit bitmaps/picture that are used to control transparency of another bitmap/picture. This picture (or 'mask' as it is called here) is applied as a transparency template, where black is regarded as opaque, and white is seen as transparent. Shades of grey are treated as degrees of transparency. When applying an alpha mask over a miniature picture in the photo mosaic picture, only the parts of that picture are visible that have a corresponding white (or grey for degrees of transparency) pixel in the mask.

### Filter

A filter is a graphical effect that can be applied to a bitmap to create a certain transformation. This can be colour (darken, saturate), overlay (buttonize, shadow)

or shape (distort, skew), or any other transformation. Famous filters are applied in graphic editors, such as Photoshop, for where a large community creates free filters. Some specific filters are hard-coded in Mosaizer XV, which cannot be changed by the user.

### 32Bit (overlay) bitmaps

Most bitmaps are created in the 24bit format, where every primary colour channel has a range 256 levels (R, G and B), in total a range of 16 million different colours. A 32bit bitmap has an extra channel (the 'alpha channel') with greyscale pixels to define the transparency of each pixel in the 24bit bitmap. You need special software to create alpha channels. Mosaizer XV can read 32bit bitmaps and use it for masker definition. A 'jpg' image has no alpha channel, and is 24bit by definition, so all 32Bit masks are of 'bmp' or 'png' format.

## 5.2 File locations

### Default installation locations for the application

- **XP:** C:\Program Files\APP\Mosaizer XV
- **Vista/Windows7/8/10:** C:\Program Files\APP\Mosaizer XV (32bit) and C:\Program Files(x86)\APP\Mosaizer XV (64bit)

**Windows XP:** Default location for the projects (\*.mpr), textures (\*.jpg) and sources (bitmaps), libraries (\*.lst; \*.dat) and masks (\*.jpg):

- C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Projects
- C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Textures
- C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Sources
- C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Libraries
- C:\Documents and Settings\All Users\Application Data\APP\Mosaizer XV\Masks

**Vista/Windows7/8/10:** Default location for the projects (\*.mpr), textures (\*.jpg) and sources (bitmaps), libraries (\*.lst; \*.dat) and masks (\*.jpg):

- C:\Program Data\APP\Mosaizer XV\Projects
- C:\Program Data\APP\Mosaizer XV\Textures
- C:\Program Data\APP\Mosaizer XV\Sources
- C:\Program Data\APP\Mosaizer XV\Libraries
- C:\Program Data\APP\Mosaizer XV\Masks

## 5.3 Licenses

With version 10 some new features are only available for licensed users. Our software appears to be frequently used in commercial work, where the user has chosen to illegally use Mosaizer XV and Mosaizer Video, and ignore the granted license for personal use only when installing the application.

For that reason some advanced features are locked or in demo mode until a valid license is purchased. These are:

- Super mosaics (no limit to 60 MB size, so super mosaics now possible)
- Number of levels in adaptive patterns not limited to 3
- Mosaic output size not limited to A2 (but to A0 and beyond)
- Mosaic video creation not limited to 5 seconds
- You can now also save tile details ('Tile\_Locations.csv')

### Purchasing a license

Simply send an e-mail to us. We will inform you about our licensing possibilities. Basically you can purchase an unlock file (lifetime validity) with or without a commercial license. A commercial license will not only unlock the extra Mosaizer XV and Mosaizer Video features, but also allow for commercial use if also a license has been purchased. The license is always valid for all the APP Helmond applications, including Textaizer Pro and Wordaizer, the unlock file can only unlock one application at the time. When only an unlock file is purchased you cannot commercially exploit the results made with our applications (e.g. sell the mosaic, printed or as picture file, offer services via internet).

### Validating the license

Via 'Tools' 'License information' you can load the license file into the application. The license will be in effect immediately. You'll notice a small '+' (plus) sign after the application's name. This indicates a valid license. The license file is stored in the same folder as the executable, and in case more than one .lic file is found, it will generate a warning and will ask you to only keep the correct .lic file, and delete, rename or move any other .lic file from that folder.

Here is video tutorial how to register the unlock (aka license) file: [YouTube](#).

