

AlfaCAD 1.6 This is about photorealistic patterns technology for filling areas, solids and traces, enhanced images features and some more

Enhancements in brief:

- *Hatching, solids and traces rendered with photo-realistic patterns*
- *Transparent PNG images*
- *Images import and storing in drawings extended by JPG format*
- *Enhanced 'Offset' and 'Trace / 2-line / 3-line' functions.*

1. Photorealist patterns

The pattern collection stored in the **Backgrounds** folder and its subfolders is made up of images in JPG, PNG or PCX format (JPG is recommended as the most compressed format with high image quality at the same time), which are seamless patterns of various materials and architectural elements, such as brick and stone walls, concrete, marble cladding and ceramics, pavements, roofing, plasters, views and sections of wood, plywood and chipboards, metals and metal products, soil, sand, gravel, stones, lawns, fabrics, carpets etc.

Catalogue of patterns is unlimited, and any custom pattern can be added to the collection. There are some formal conditions necessary to fulfil:

- Size of the pattern in each direction has to be a power of 2, and shouldn't exceed 1024 pxl. So legal sizes are: 16, 32, 64, 128, 256, 512 and 1024 pxl.
- Accepted formats: JPG, PNG, PCX
- Colour depth 24 or 32 bits (however 8 bit images are also accepted)
- The pattern on the image should be seamless, as all pre-installed patterns are.

Below is an example of seamless pattern (wall_cladding_stone_254.jpg).

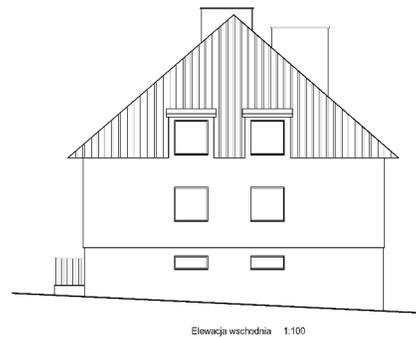
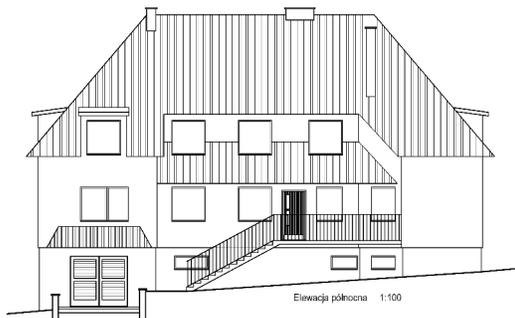


The image size is 1024x1024 pxl (so the maximum size) what is preferable for high resolution printing (the same patterns are used for screen rendering as well as output images saving and drawing printing).

Images are scalable in limited range, between 1/8 of the size to 8x the original size (counted in pixels). Due to used fast but rough rendering technology, scale factor is rounded to one of 14 available scale factors, both for presentation on the screen as well for printing. The pattern can be rotated by 90°, 180°, and 270°, origin point can be moved by 1/32, 1/16, 1/8, 1/4, 3/8, 1/2, 5/8, 3/4 and 7/8 of the size of the pattern, to mark elements stitching, making the pattern visually different.

2. What advantage brings this new feature?

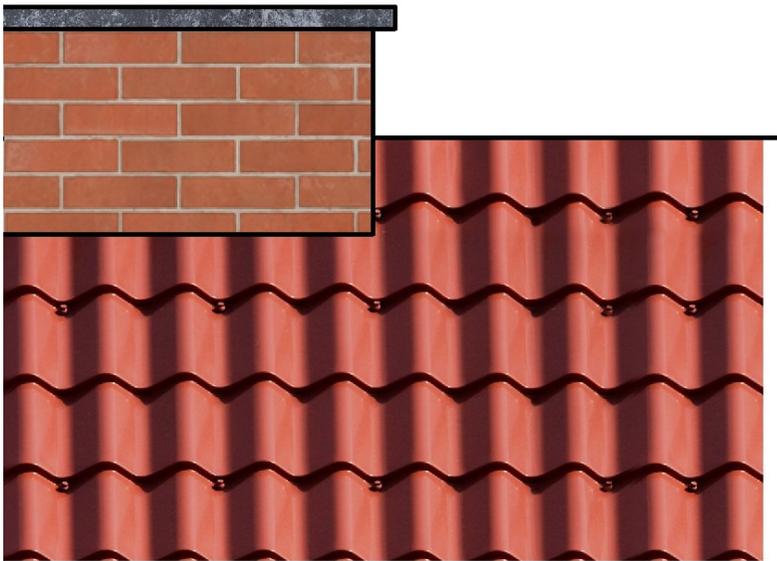
Here is an example. Simple drawing of house facades:



In few dozen clicks (actually 20 minutes work) it was transformed to this (background, clouds and trees were added “for fun”):



with such details:



There are many missing elements what would appear on 3D model, like gutters etc, but in scale 1:50 or 1:100 that is not visible anyway, so such „rough” rendering is more than enough to show facades more attractive than simple vector drawing. All that in just few dozen clicks.

Drawing can include the background image or images (including transparent images like clouds) as well as foreground images (e.g. partially transparent tree or other plants images).

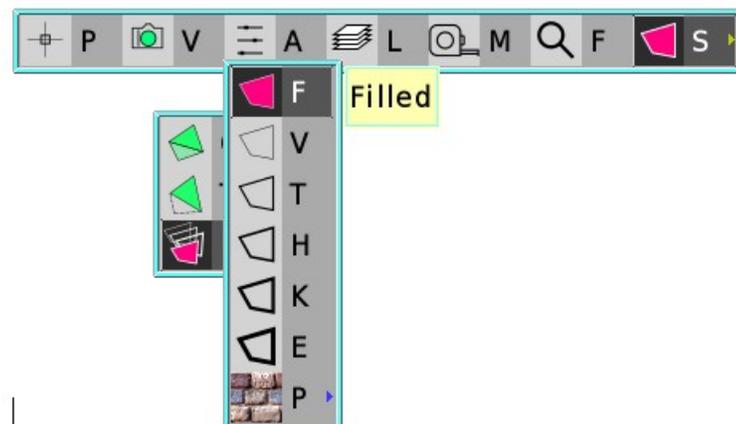
3. How does it work ?

AlfaCAD has now 3 types of objects able to be rendered by image patterns:

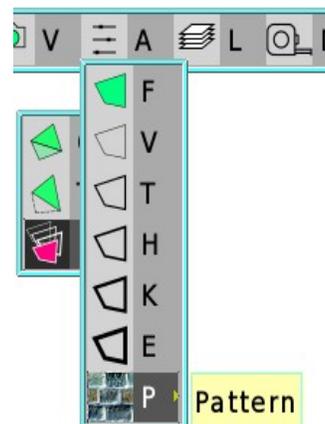
- Solids
- Traces
- Hatching blocks

Solids and traces (traces are built from solids creating regular path of fixed or variable width) can be filled with colour or can be just outlined by the lines of selected width, from very thin to extra thick. Now, starting from v. 1.6 AlfaCAD offers solids and traces filled by image patterns.

In auxiliary menu for Draw Solid function new options appear.



Next to „Filled” (with colour) or alternatively thickness of the solid outline, „Pattern” option is available:



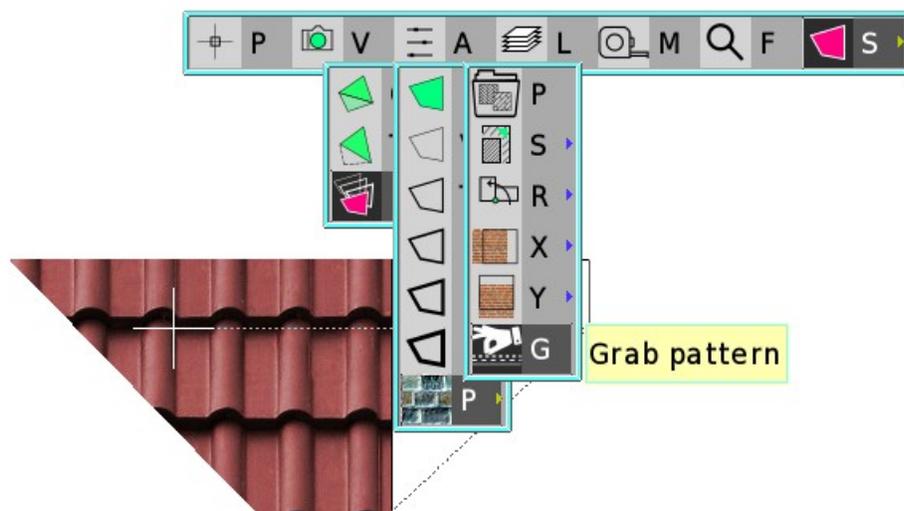
“Pattern” option with sub-menus is common for **Solids**, **Traces** and **Hatch** objects.

Pattern image is filling Solids, Traces and Hatch blocks in real time based on pattern image embedded in drawing. Once pattern is inserted, its original image is saved in the drawing file or exported block file. Opening drawing file or importing previously saved block including image pattern, loads the image from the file to fill object in real time during all screen operations and while file is printing or saved as graphic image file or PDF document. The only indicator of pattern is it’s named inherited from original pattern file name.

Replicating existing pattern

Image patterns are embedded in drawing. Pattern is replicated in all solids or traces (also hatching blocks) wherever it is used, and it is recognized by its name inherited from original pattern file name. If current catalogue does not contain such pattern anymore, or such pattern was never existed in catalogue, it still can be used, cos is embedded in the drawing anyway. Each solid or trace can use individual parameters, like scale, rotation angle or origin point shift, but the image pattern is common, embedded just once.

When new solid or trace or hatching block is created, existing image pattern can be used by choosing “Grab pattern” function from auxiliary menu:

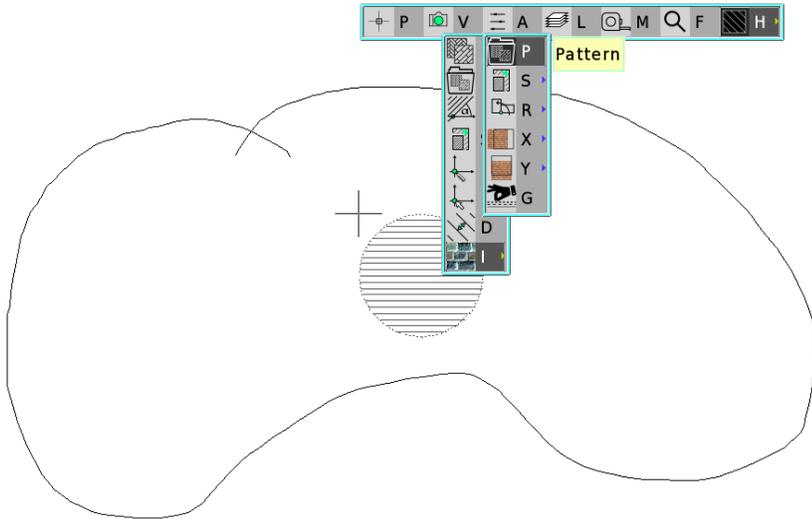


Click on existing patterned solid, trace or hatch block takes the name of embedded image pattern to be used in newly created object.

4. Hatching with image pattern

Hatching in AlfaCAD can be done using vector patterns or solid colour. In case of solid colour, the enclosed marked area is divided into the number of solid objects, filled with colour, stitched together including the hatching boundaries. Since version 1.6, due to extension of solid object specification allowing to fill solids (so also traces) with image pattern, hatching can be done using image patterns as well. No matter how many solids are used to fill in the regular or irregular enclosed area, the image pattern, using identical origin point for each chunk of hatching, is filling seamlessly whole area.

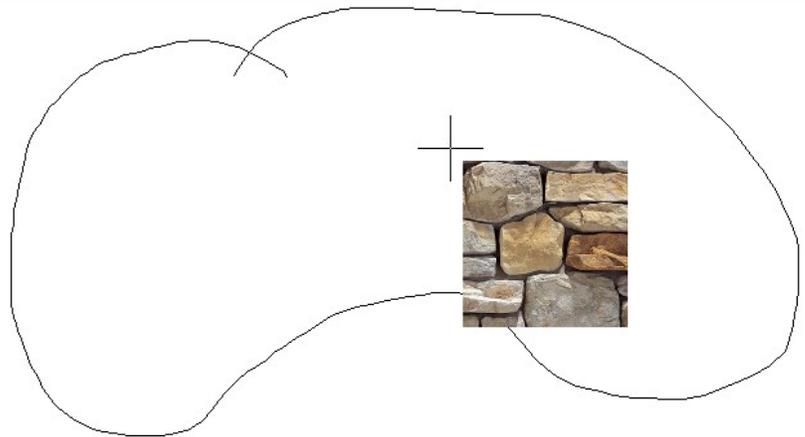
Below example of very irregular shape created using Sketch function, filled with some stonewall image pattern:



On identical way as in case of Solid or Trace objects, image pattern can be selected.

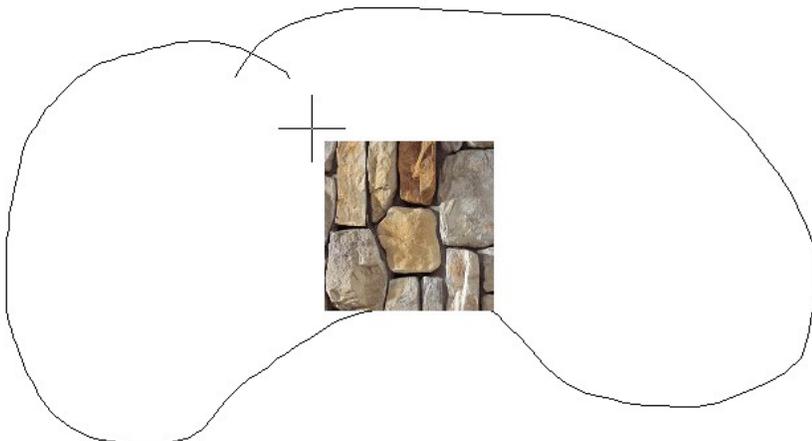
To properly evaluate if selected pattern and its parameters like scale and rotation angle matches the needs, next to the pointer the sample of pattern appears, in realm scale and rotation. In the upper row the instruction appears prompting to indicate the point inside previously selected enclosed area, or press **Home** key to rotate pattern 90° or **End** key to rotate -90°

Indicate the inside of the area (Home +90°,End -90°)
 Pattern : HATCH IMAGE



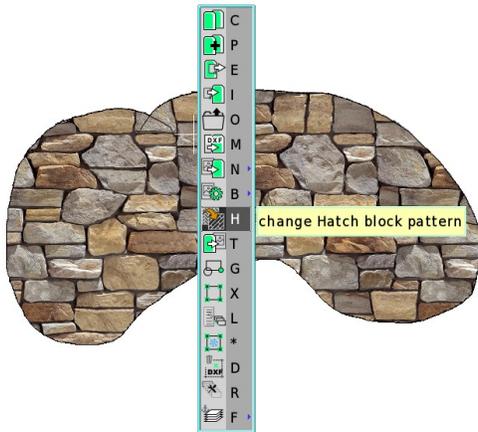
Indicate the inside of the area (Home +90°,End -90°)
 Pattern : HATCH IMAGE

The pattern rotated 90°



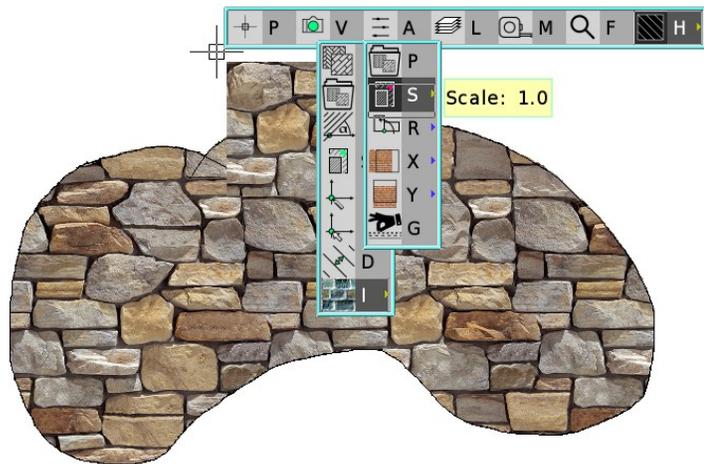
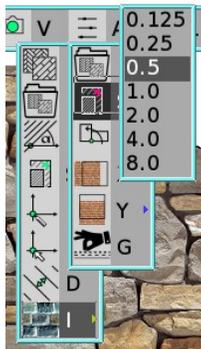
After clicking inside area, the shape is filled with pattern:

If the pattern seems to be not adequate, for example the scale is too big, it can be easily changed. In **“Block” “change block Hatch pattern”** function, existing hatching block can be re-defined.



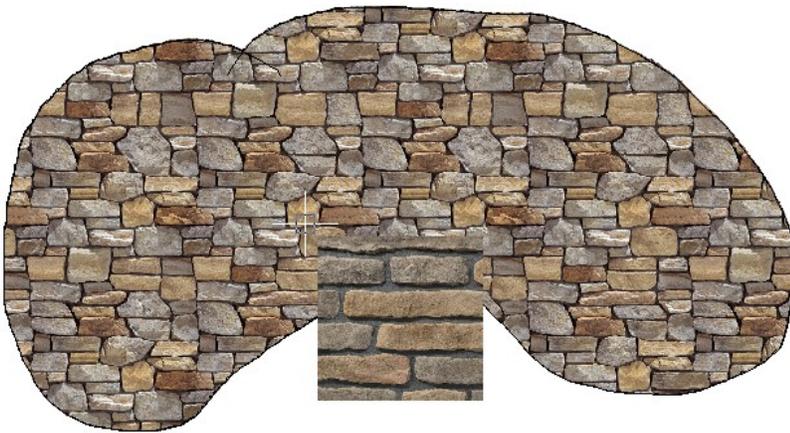
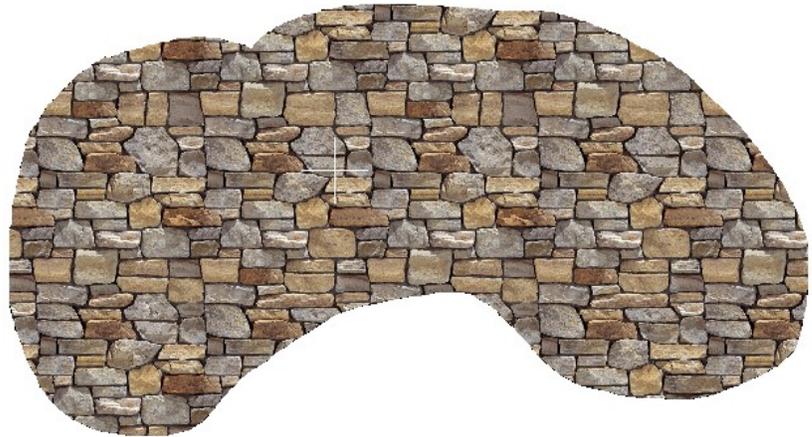
If it is too big, scale can be changed:

Indicate the inside of the area (Home +90°,End -90°)
Pattern : HATCH IMAGE



The sample next to the cursor shows the image in changed scale, and now it's just necessary to indicate the hatching block which is going to be replaced by new scaled pattern:

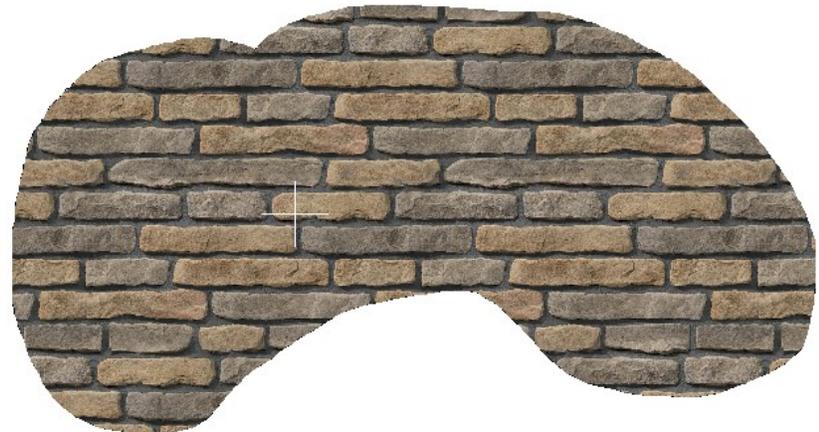
If the selected pattern is not really satisfying, it can be change on the same way, but this time instead of changing scale or rotation angle, another pattern can be selected from catalogue, or grabbed from another object like hatching block, solid or trace. Below different stonewall pattern is selected:



Sample next to the cursor is changed to new patterns, honouring previously selected scale. Any parameter can be changed with instant view in the sample, what helps to evaluate if hatching pattern fits the needs.

After clicking again inside existing hatching block, whole area is rendered again with new pattern:

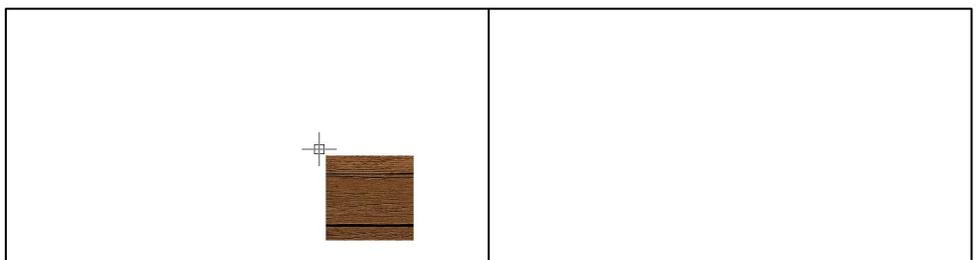
This procedure can be repeated until reaching desired view of hatching block. Once hatching block is created, the boundaries of the enclosed area are included in the block, so changing pattern or its parameters in “Block” – “change block Hatch pattern” function does not require redlining the area boundaries.



Origin point shifting

These two parameters (separately for the X and Y axes) are used to define the boundary between two areas that need to be distinguished from each other, which do not constitute a whole, but rather border each other.

Below is an example of a wooden floor pattern filling two adjacent rectangles.



Using zero X and Y origin point shifting:



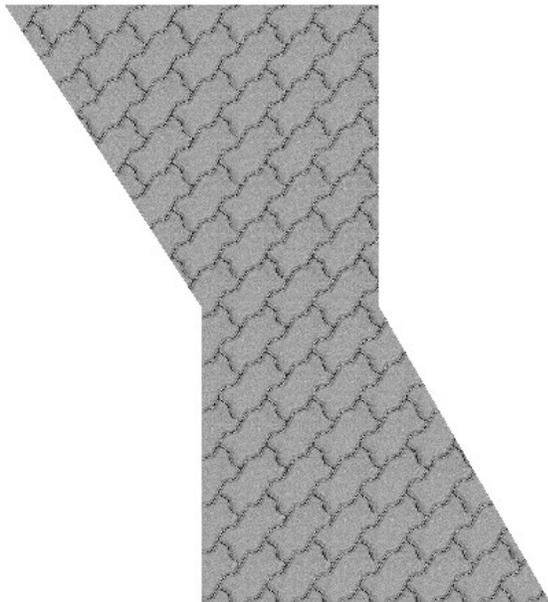
Using 1/16 Y axis shifting for right rectangle:



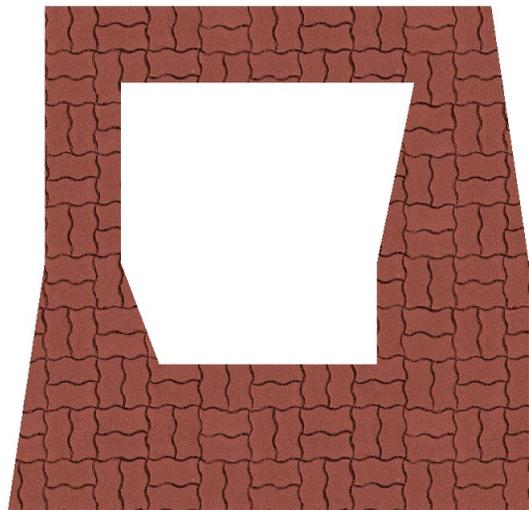
5. Solids and Traces with image pattern

Traces are built from the chain of solids, creating quasi polyline block or individual unblocked objects, depends of selected option (by default trace solids are blocked within each chain).

Here is an example of solids filled with image pattern:



and multiple width trace with the similar image pattern:



Modifying existing pattern for solids and traces

Every object in AlfaCAD can be easily modified. The same applies to solids and traces. Each solid or trace segment, which is always quadrilateral solid, can be set as outline using one of 5 available line widths (very thin, thin, thick, very thick and extra thick), or can be filled with colour, or can be filled with image pattern. Those 7 options can be chosen for new solids or traces or modified for existing ones at any time, using appropriate submenu “Solid” or “Trace” in auxiliary menu. Changing the type of existing solid or trace segment is considered as actually re-writing the object, due to changes necessary to do in object data structure, not just amending some variable values, especially when simple outlined or colour-filled solid is replaced by image patterned one, where each solid contains the set of parameters like the identifier (name) of the image, scale, angle as well as origin point shifting in X and Y direction.

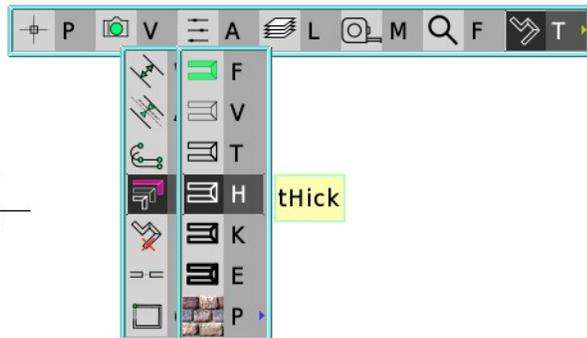
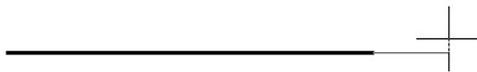
Below the very simple example how modification of solids or traces can be done, during creating the complex trace, or after creation is completed. Every change can be based on manually selected type or pattern, or can be taken directly from existing object like solid, trace or image pattern hatching block.

Let's start the trace. First click marks trace start point.

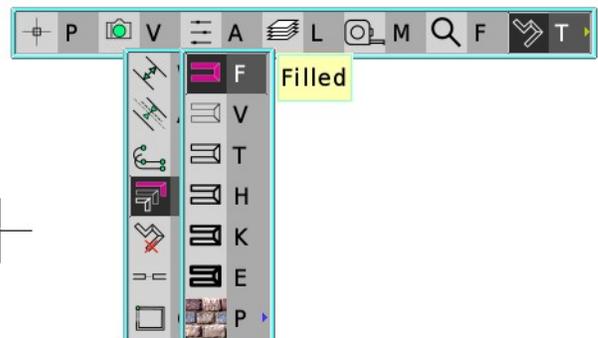
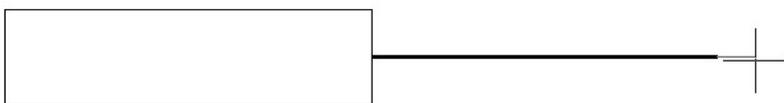


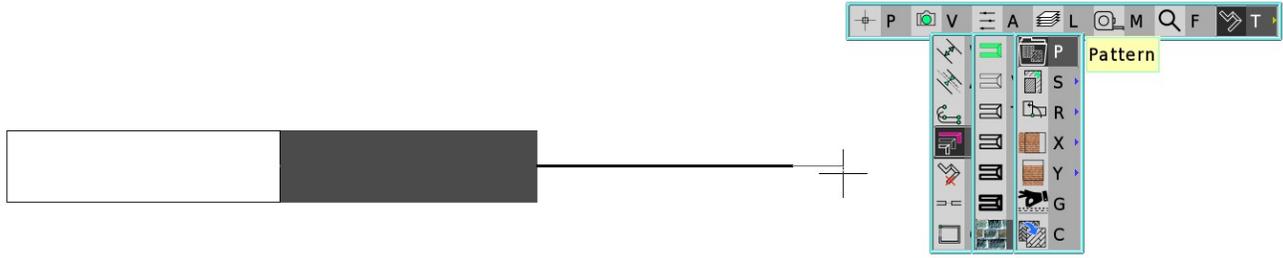
After the second point is marked with another click, thick line appears showing the direction of trace segment, however solid is not created yet, due to unknown angle between current segment and the next one. As long as the solid is not yet created, we can select type of solid.

Let's select thick line, to get empty solid outlined with thick line:



When next point is marked, outlined segment is created and next segment is marked with tick line. Now, in auxiliary menu, let's select Filled type



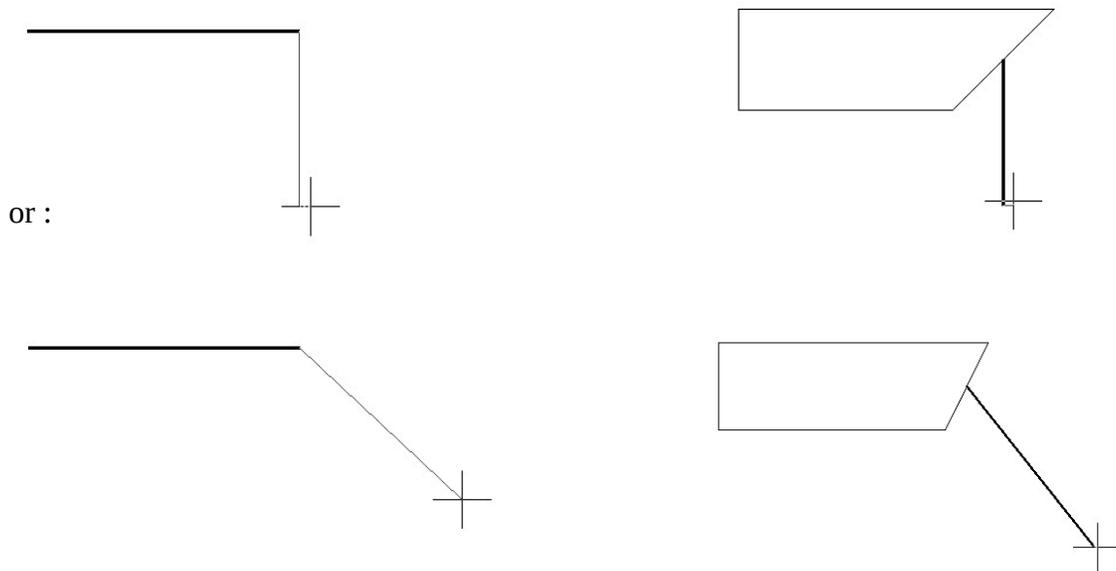


When another point is marked, second segment is created as filled solid, then let's select image pattern for next segment.

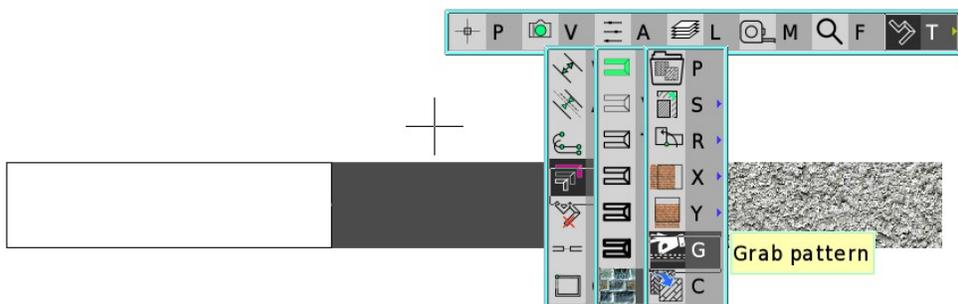
Finally, escaping the function, whole trace is completed.



Each segment is created as a solid after next segment is marked with the thick line. Direction of next segment defines the angle between segment. In case of not collinear segments, the trace can look like this:



Let's modify that trace created above. In the same function (Draw Trace) in auxiliary menu Trace, then Fill/line width – Pattern, choose “Grab pattern”

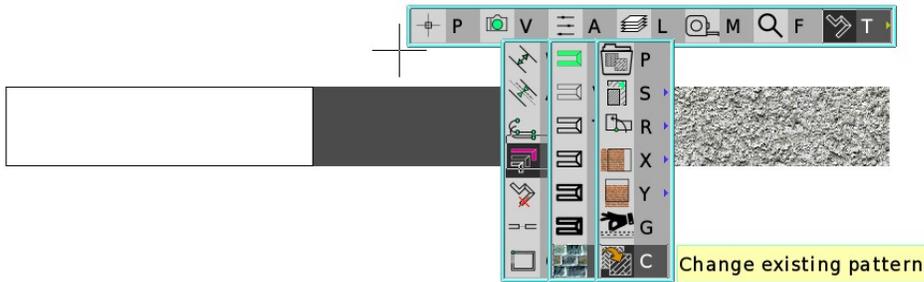


Pick a pattern filled area, solid or trace to determine fill parameters



then let's select image pattern solid.

Once the pattern is selected, or in general – solid type (can be also outline with particular line width, or colour fill), lets come back to the last menu and let's choose the last option: “Change existing pattern”



Now let's select solid (trace segment) to change:

Pick a pattern filled area, solid or trace to change fill parameters



When first segment is changed, what doesn't finish the function yet, program is waiting for another choice, or ESC key (or right mouse button click) to leave the function.

Pick a pattern filled area, solid or trace to change fill parameters



So let's select another segment:



Finally the job is done:



Notice:

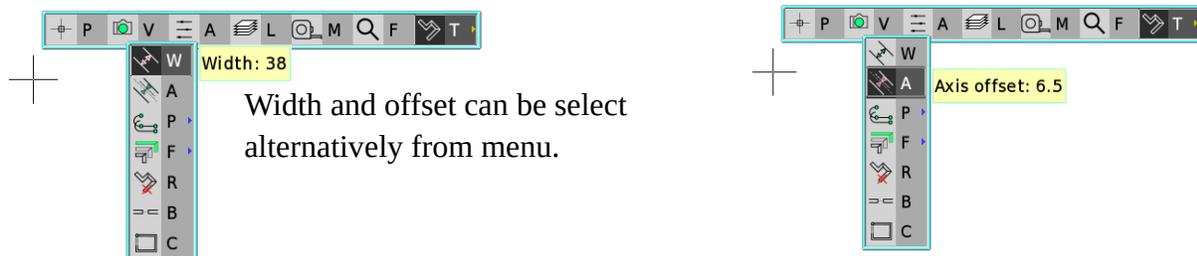
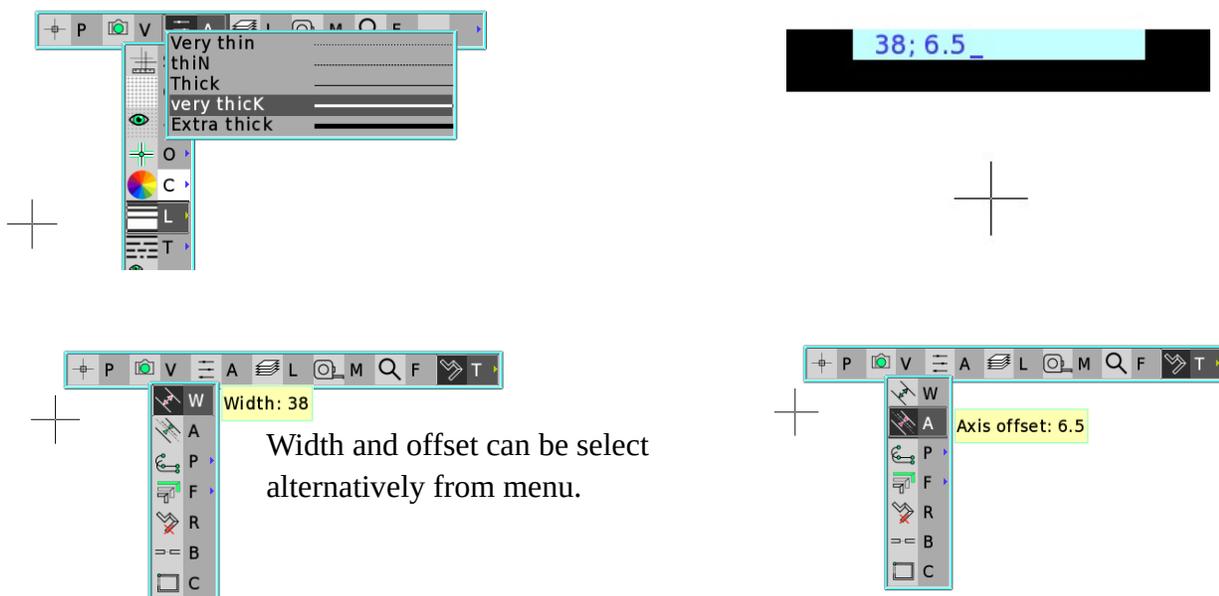
Changing solids and traces types and patterns are only available in Draw Solid and Draw Trace function, however there are limited change possible in “Edit” - “Change” function. Image patterned solids and traces are not changed in “Edit” – “Change”, where many object can be selected to change properties at a time. This is due to the protection against accidental change of patterns, so they can be only done one by one, however, work on the most optimal, but safe method is still ongoing.

6. Traces, 2-line and 3-line complex quasi polylines

Trace, 2-line and **3-line** are quite strong tools to create quasi polylines built respectively with solids, two bordering lines or 2 bordering lines and axis line. There are many uses of these functions, and one of them can be sketching a wall plan of a designed or inventoried building. Such a plan usually consists of a closed chain of wall sectional fragments with window and door openings shown. The function allows only a sketch of walls, without taking into account such details as window embrasures, therefore such a sketch must then be subjected to some modification. The main purpose of the sketch is to create the skeleton of the plan in real dimensions.

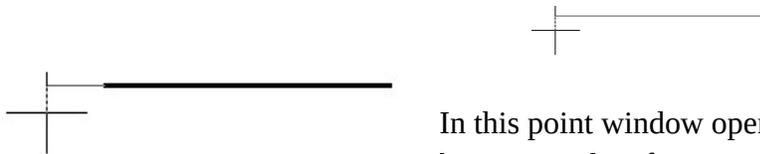
Here is a simple example. Let’s assume the building’s wall is multilayer wall containing structure ceramic hollow clay tile of the thickness of 250 mm, external insulation board of 65 mm and Tuscan brick cladding of the thickness of 65 mm. Let’s assume also we base of dimensions marked by axis of structural wall, so if the thickness of all layers of wall is $250+65+65=380$ mm, while axis of structural part lays in the distance of 125 mm from internal edge, this axis is shifted by $380/2 - 125 = 65$ mm from the geometrical axis (centre) of wall. The value is positive, if we are going to draft the wall anticlockwise (same direction as angle value is increasing). If the direction of drawing the wall is clockwise, the axis shifting relative to the centre of wall has to be negative.

Let’s start. We select very thick line type for cross section edges of the wall. Selecting “Draw” – “3-line” using TAB key we can edit trace thickness and axis offset: 38; 6.5 (in cm, due to selected units).

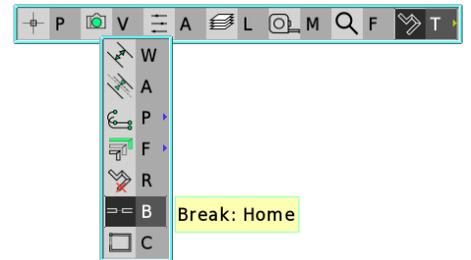


Let's start 3-line: first point

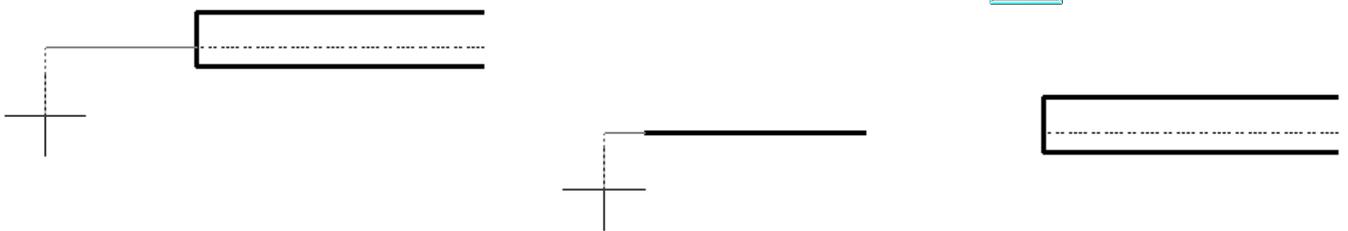
Then second point:



In this point window opening should begins, so let's press **Home** button or select from menu:



So this way window opening is marked on the right side.



Continuing drawing supported with numeric values of the length of lines, or single value in orthogonal mode, or length and angle in non-orthogonal mode, in edit field reachable after pressing **TAB** key, window opening is marked, then another line, and after pressing **Home** key again to make another break both ends of wall section will be added together with edges and wall axis:

Now angle is changed, orthogonal mode is off (**Ctrl-Q**), next wall section is the beginning of the oriel.

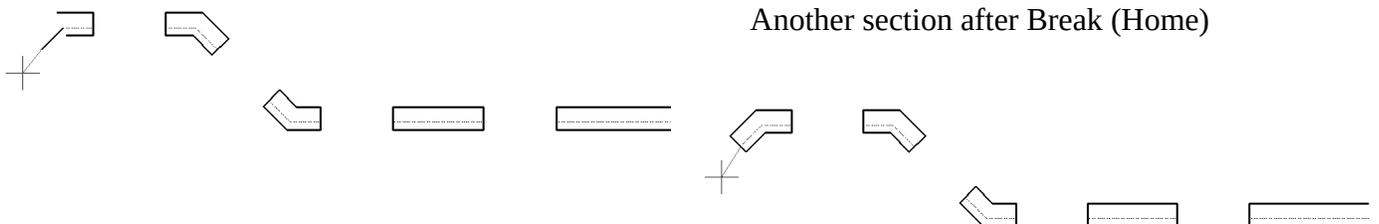
After pressing **Home** key or choosing "Break" option from auxiliary menu, first wall section of oriel is finished.



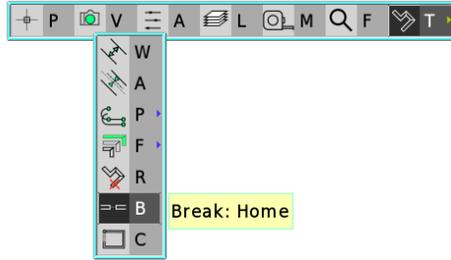
Procedure can be repeated till the end of oriel, where always **Break (Home)** closes the wall section, and stating another one, or marking opening.



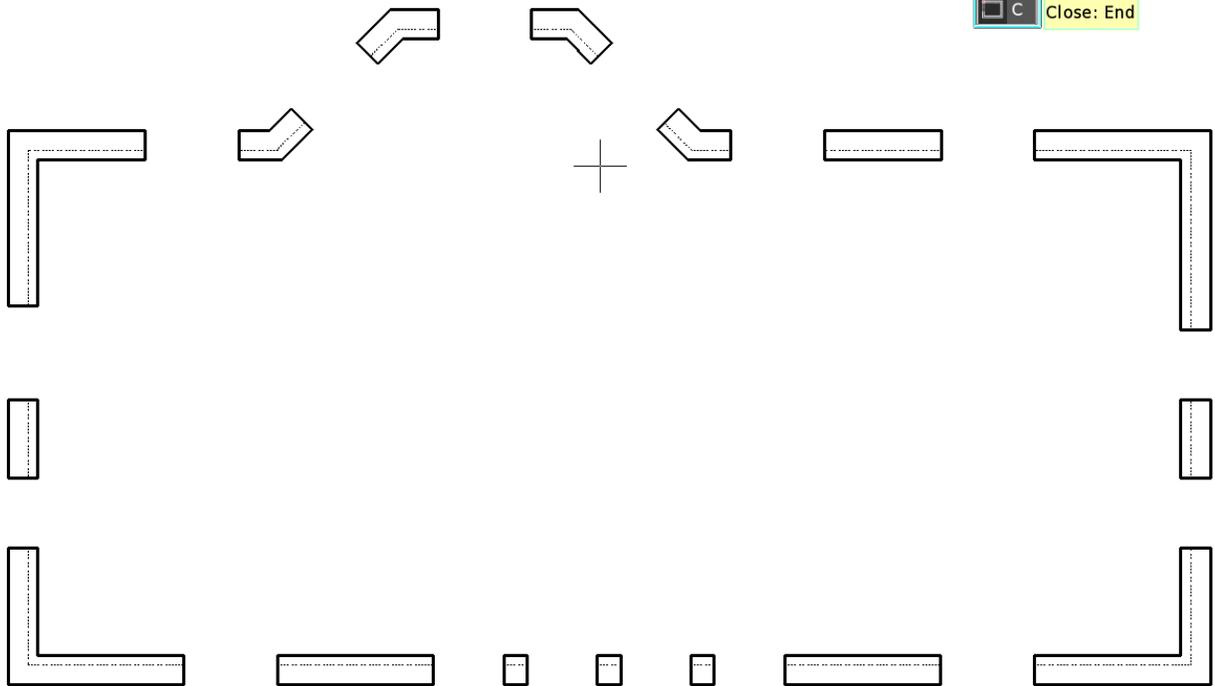
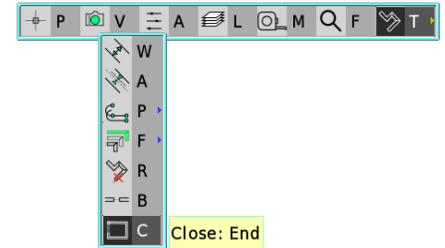
Another section after Break (Home)



Continuing around the whole building, using numeric values, and breaking wall with Break function (Home key):

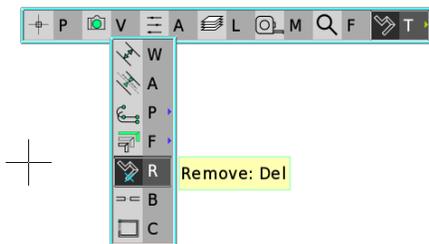


the last segment can close the wall, using **Close** function (or **End** key)



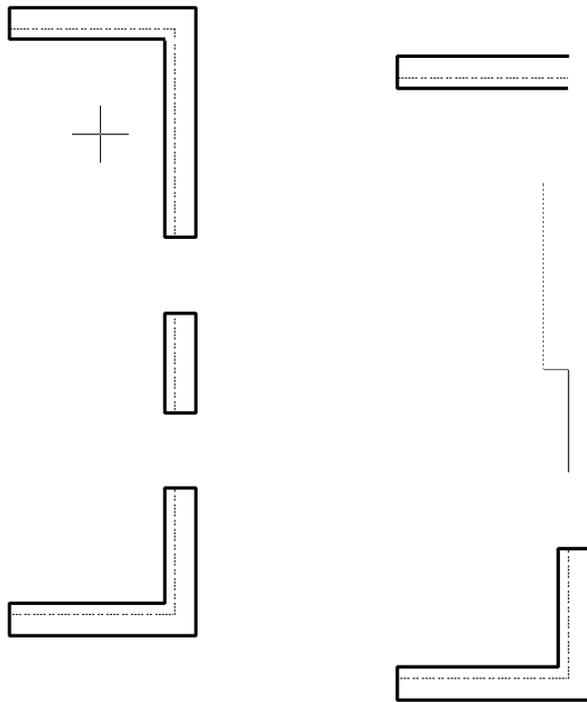
Deleting last segment

Each previously created segment of quasi polyline (trace, 2-line or 3-line) can be successively deleted one by one in function **Remove** (or **Delete** key).



After each section deletion, new section can be added, wall section or opening. When trace is already close, and mistake was discovered, it can be still open using Remove function, to remove the last segment, re-modify first segment to the shape before trace was closed (closing trace usually leads to the change of first segment too).

Here is a fragment of the trace after closing:



and after using **Remove** function:

Wall is open again, every previous section can be removed too, one by one up to the very beginning, and replaced with proper segment, in case of mistake in shape, dimensions and so on, and so forth.

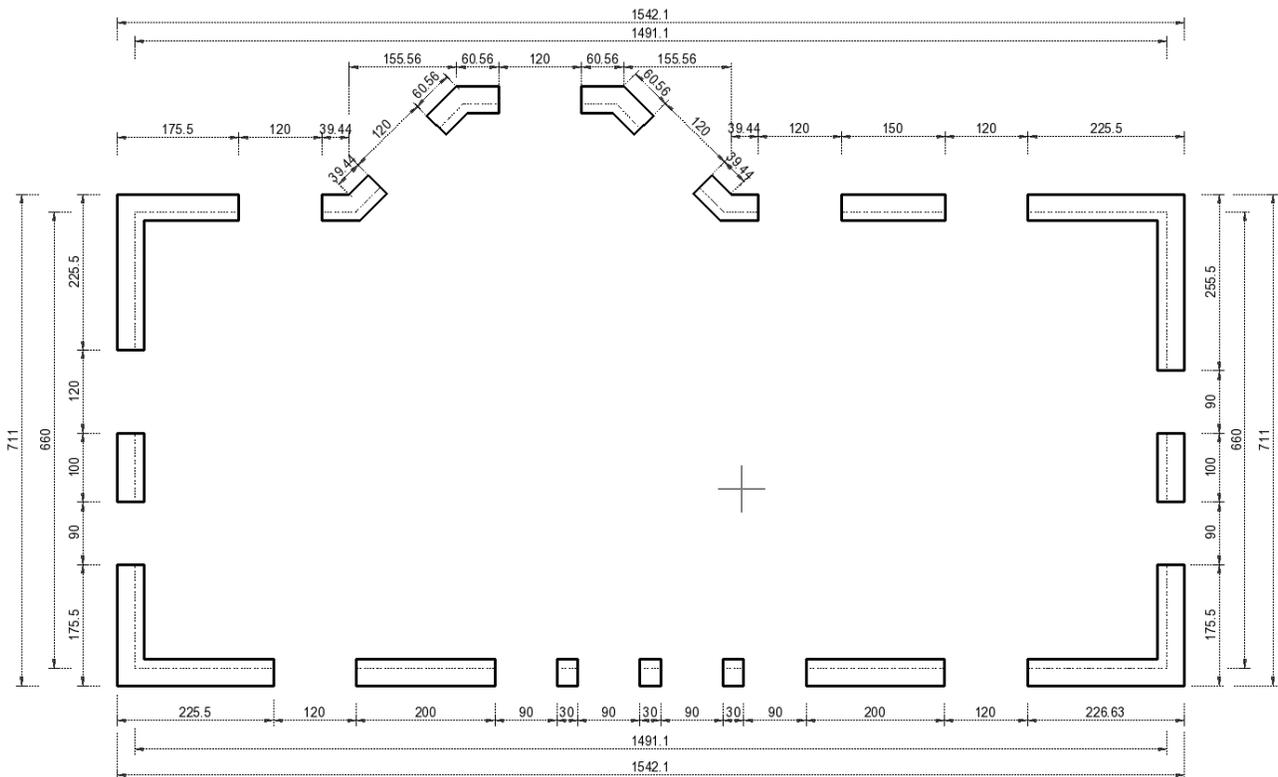
Removing last segment of trace (closed trace) is only possible before next trace drawing began.

If this opportunity is missed, whole trace can be edited on the regular way, like every object in AlfaCAD.

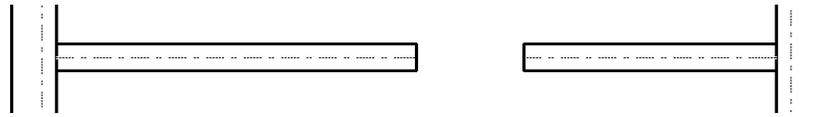
Small mistake can be always corrected aftermath, using Stretch function, Break, Trim, Extend etc.

Editing of the polyline, what trace actually creates, is slightly limited, as long it is a polyline block, what is useful in some editing function, like moving, rotating or copying. Exploding that block allows editing every object individually without any limitation.

At the end let's see what we got, dimensioning traced wall:



On the same way internal structure walls can be drawn. Finishing trace with ESC key leave polyline open, so there internal wall:



The same wall after some amendments (breaking lines and extending axis):



AlfaCAD is not designed to specifically support architectural design. Trace, 2-line, and 3-line, however, can make it much easier to draw building plans. Works on extending the Trace functionality is ongoing. Such extensions will appear in future versions of the program.

7. Transparent images

AlfaCAD can import now transparent PNG images, with 32bit colour depth (24bit colour plus alpha channel). Images can be imported as another images in PNG, JPG or PCX format. Other formats are converted to PNG, JPG or PCX depending of chosen option in Block – Insert image function:

Insert Map is considered as importing image file in compressed, lossless PCX format, mostly 1 and 8 bit colour depth, however also 24bit files can be inserted. This function is dedicated mostly large scanned documents, like cartographic maps and geodetic sleepers.

Import compressed lossless PNG format files including transparent 32bit images.

Import highly compressed, lossy JPG format files. JPG doesn't support transparency.

insert Map

insert photo as PNG

insert photo as JPG

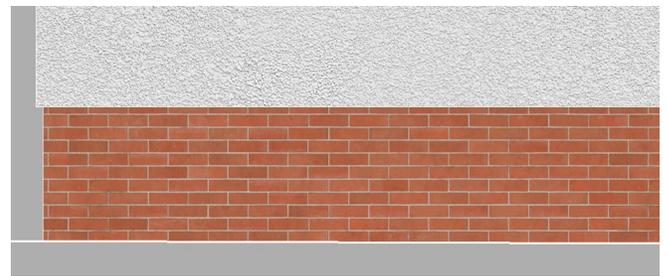
When AlfaCAD is displaying drawing on the screen or printing on printer as well as creating the image to be saved in graphic format file, first background images are drawn (maps and other regular images), next image filled and colour filled solids (also traces), next all vector objects, like lines, arcs, circles, text (vector and True Type Fonts) etc. At the very end, “On Top” images are drawn. Every image can get “On Top” attribute, to be drawn on top of other objects, however it is mostly used in case of transparent images.

Below the example of applying plans transparent PNG image on top of existing objects, in this case image pattern hatched area:



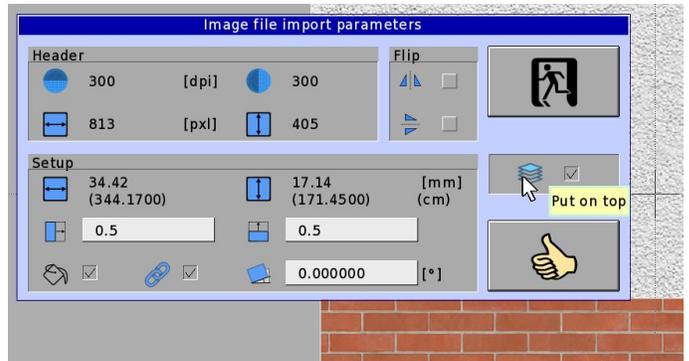
Facade made with lines, and image pattern hatched polylines, so clay cladding tiles, plasters, roof tiles, window glass, concrete, wooden balcony door and window frames.

In Block - iNsert image – insert photo as PNG we choose transparent image with plants:



In parameters size is set and “Put on top” is tagged

Image is moved to the front of the wall.



On the example cloud image is set as regular image (not on top) so is considered as the background, then all hatching solids are drawn, then all primitives like lines, arcs etc, at the end “On top” images a drawn completing whole image both on the screen as well as on the printer or image file.

8. Printing the background

When background colour used in drawing, needs to be printed (what in some situations makes drawing printout more attractive or expressive, especially on drawings where images are involved, like transparent images of clouds, plants) in printer dialogue window new option enables this to happen.



In the current version, only colours from the standard 255-colour palette can be used as a background, both on the screen and on the printer. This palette will be expanded in the near future to any colour from the True Colour palette (24bit).



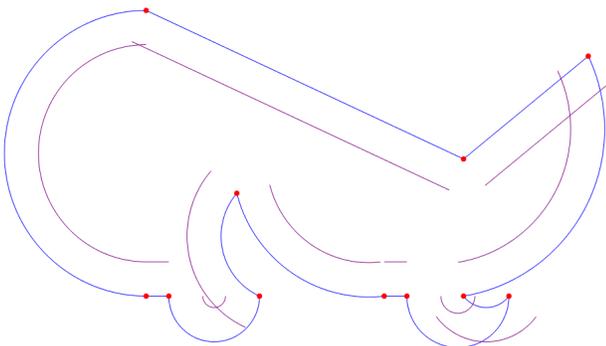
9. "Offset" function extension for polylines

"Offset" allows to make a parallel copy of line, arc, circle, and now also complex object like polyline, which is the combination of lines and arcs bound into the chain.

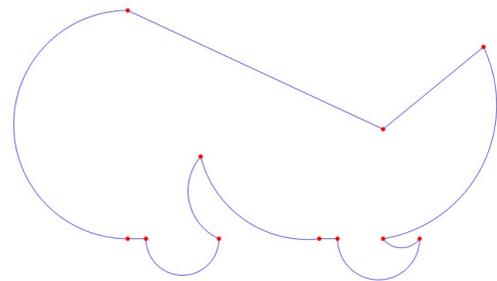
Polyline "Offset" uses quite complex algorithm, based on work of Liu et al ([1] Liu, X.-Z., Yong, J.-H., Zheng, G.-Q., & Sun, J.-G. (2007). *An offset algorithm for polyline curves. Computers in Industry*, 58(3), 240–254.) with further works Jedidiah Buck McCready (<https://github.com/jbuckmccready/CavalierContours#references>) then further extension by AlfaCAD author.

Polyline "Offset" is done in few steps:

- Generates raw offset segments from the indicated polyline.



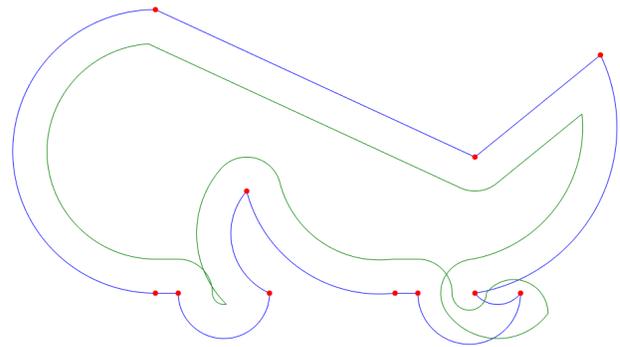
Raw offset segments generated in purple (Step 1)



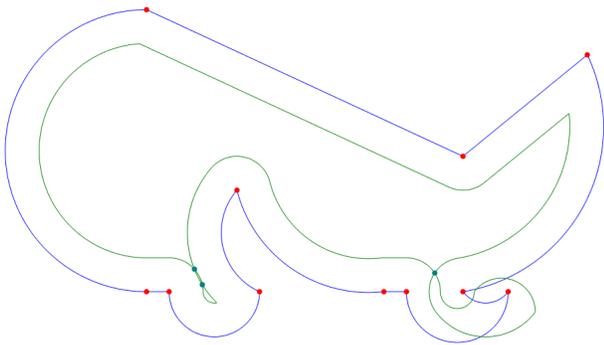
Original input polyline, pline in blue, vertexes in red

- Creates the raw offset polyline, pline1, by trimming/joining raw offset segments acquired in step3.

- If the input polyline, has self intersections or is an open polyline then repeat steps 1 and 2 with the offset negated (e.g. if the offset was 0.5 then create raw offset polyline with offset of -0.5), this is known as pline2.
- Finds all self-intersects of pline1. If step 3 was performed then also finds all intersects between pline1 and pline2. If pline is an open polyline then also find intersects between pline1 and circles at the start and end vertex points of pline with radius equal to the offset.

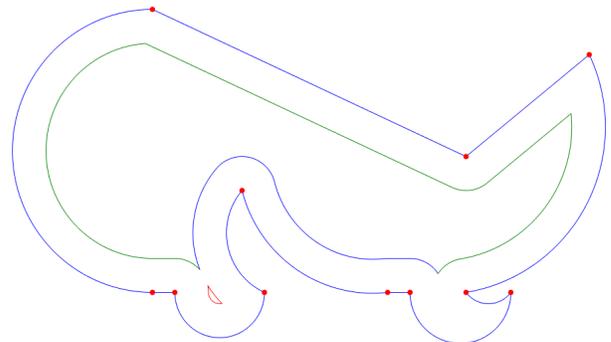


Raw offset polyline created from raw offset segments, pline1 (in green) (Step 2)

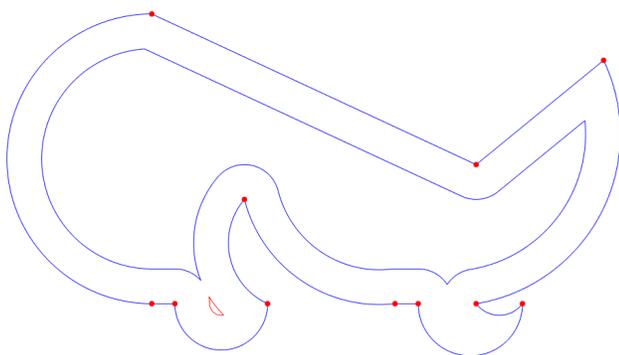


Raw offset polyline self intersects (dark cyan) (Step 4)

- Creates a set of open polylines by slicing pline1 at all of the intersect points found in step 4.
- Discards all open polyline slices whose minimum distance to pline is less than the offset.



Valid open polyline slices created from self intersects (in green, red, and blue) (Step 5 & 6)



Open polyline slices stitched together (in red and blue) (Step 7)

- Stitches together the remaining open polyline slices found in step 6, closing the final stitched results if pline is closed.

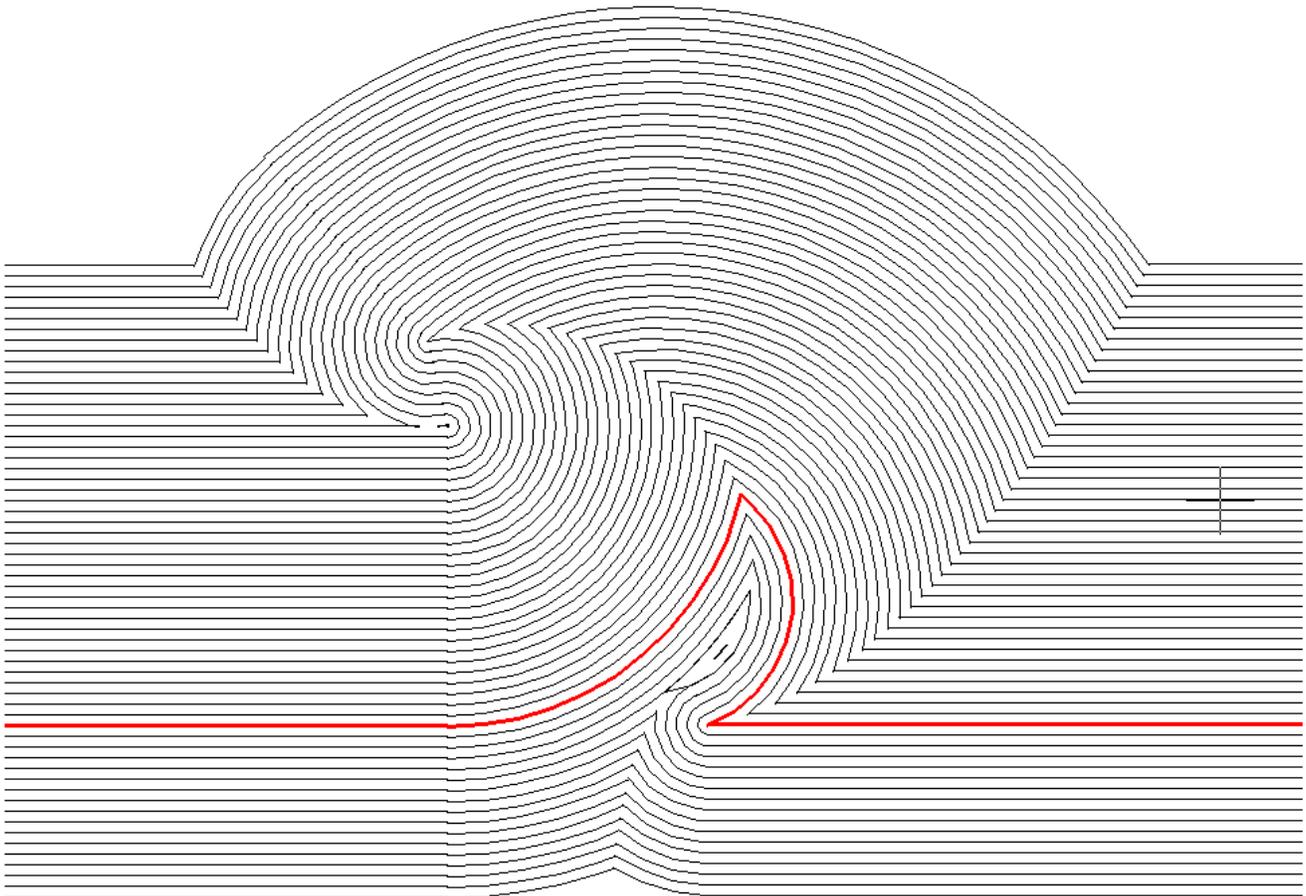
The key differences comparing with original algorithm:

- When raw offset segments are extended to form a raw offset polyline they are alternatively:
 - extended to be stitch together (it means, the rectangle is copied as a bigger rectangle (“Normal” method – extension by AlfaCAD author) however, only in case of regular polygons such transformation will be homothetic.

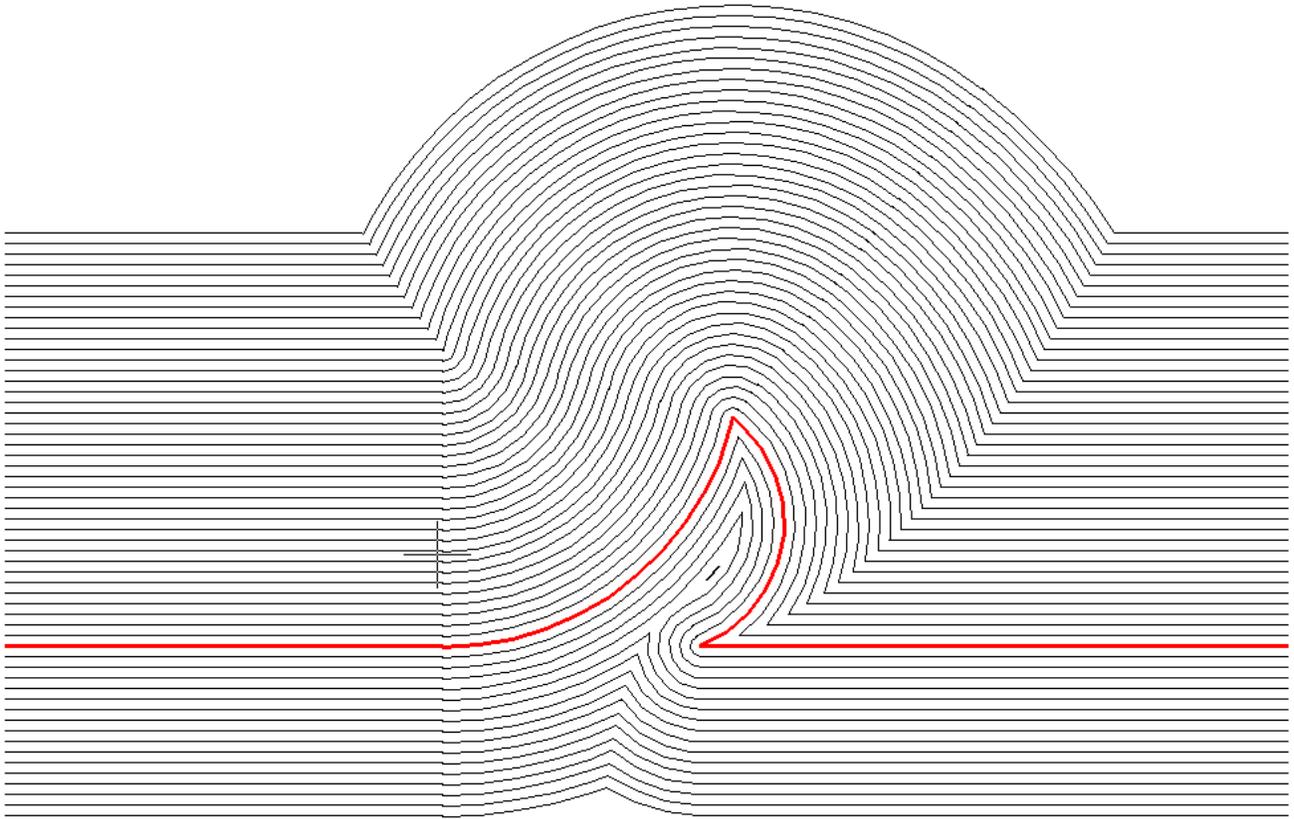
- joined by an arc to form a rounded constant distance from the input polyline (“Smooth” method)
- Dual offset clipping is only applied if input polyline is open or has self intersects, it is not required for a closed polyline with no self intersects.
- If the polyline is open then a circle is formed at each end point with radius equal to the offset, the intersects between those circles and the raw offset polyline are included when forming slices.
- Collapsing arc segments (arcs whose radius is less than the offset value) are converted into a line and specially marked for joining purposes.

Below testing example of multiple offsets using identical distance from each other, of the complex polyline with Normal and Smooth methods. The original polyline is marked with red, thick line. The offsets were done internally and externally.

In any case, offset is performed in the declared distance from original polyline and the side which is indicated by the pointer, or the offset copy nearest polyline segment comes trough the point indicated by the pointer. In the example “distance” was selected, so each offset copy is located in identical distance from previous copy.

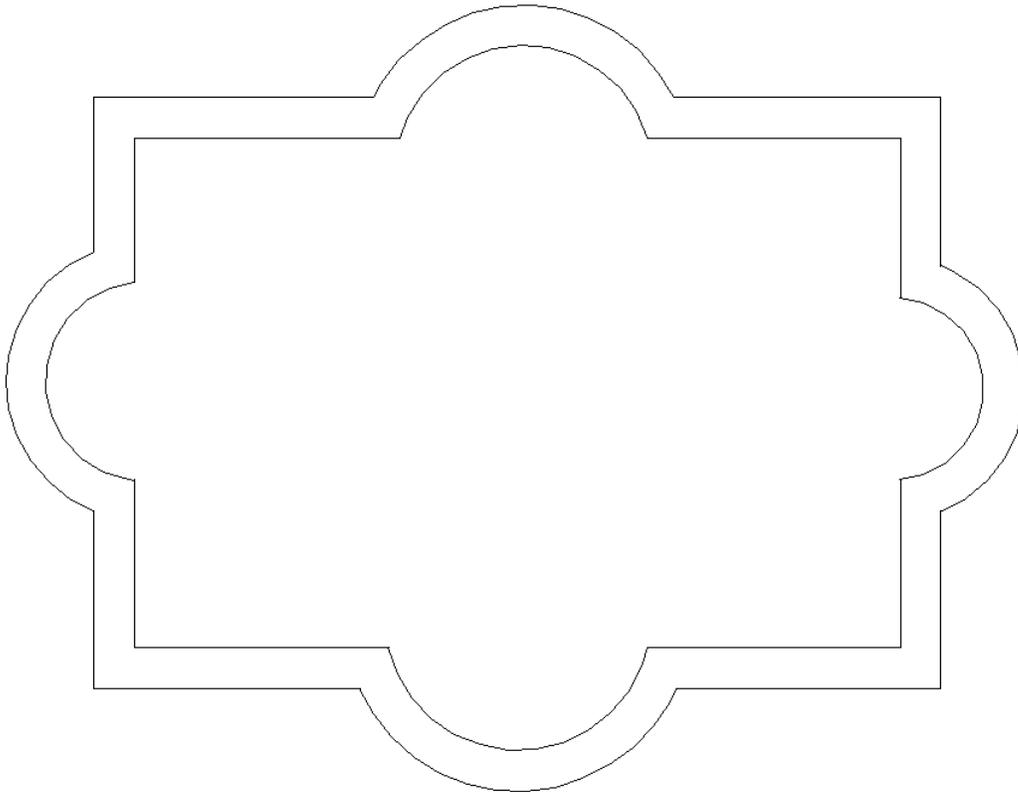


Normal method



Smooth method

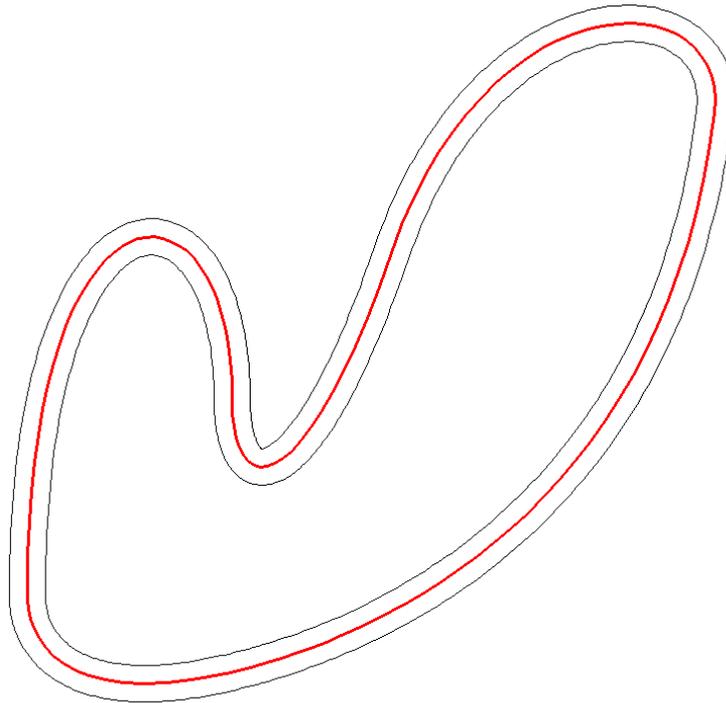
Here more practical polyline offset example:



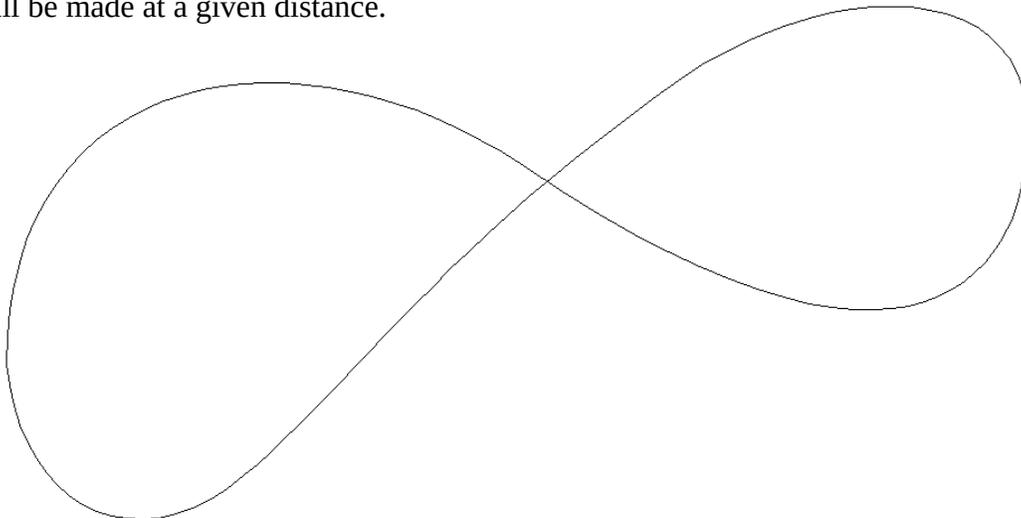
9.1. "Offset" function extension for Bezier spline and Bezier splines polyline

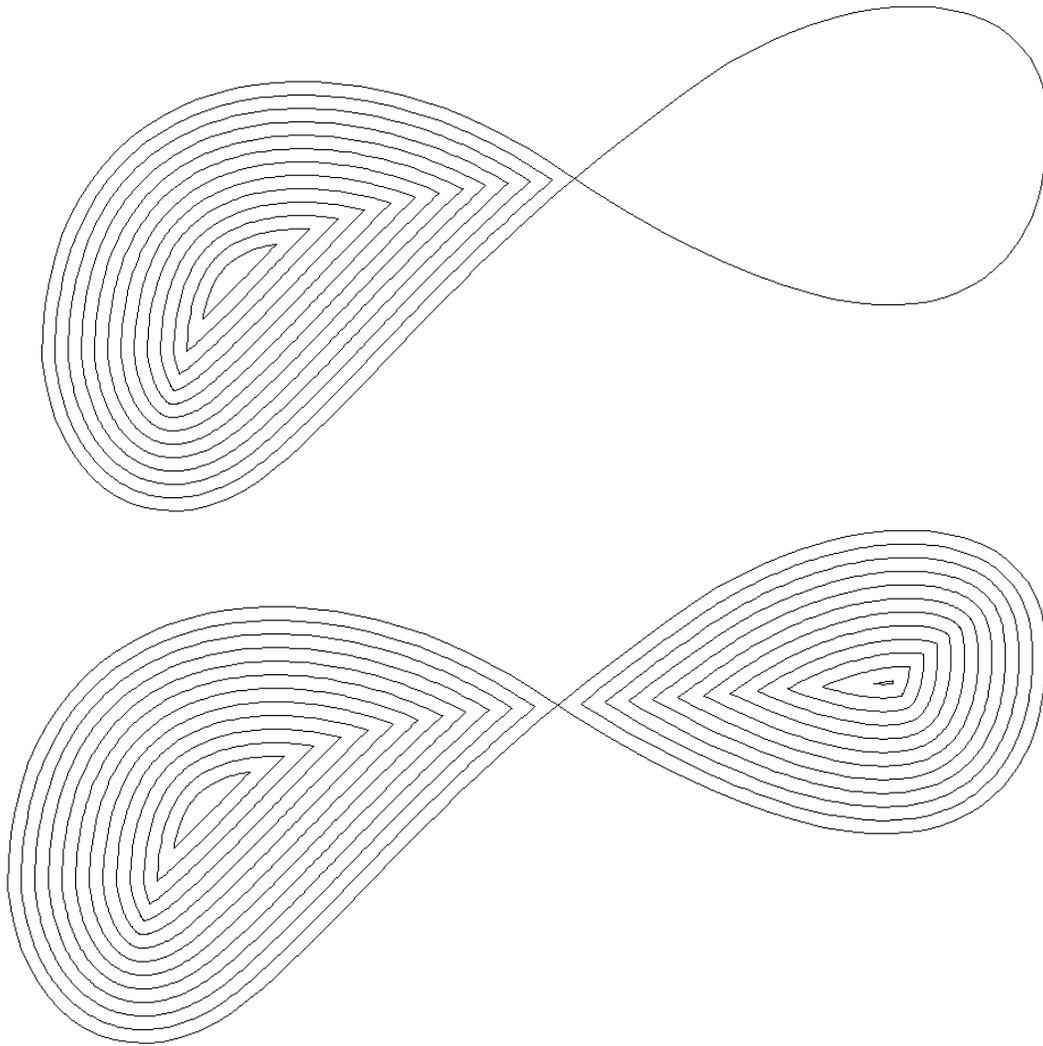
A Bezier curve (spline) can be drawn as a single curve or unblocked joined curves, or as a polyline where each segment of the polyline is a spline (the default).

It is not easy to generate a Bezier spline where every part of it lies in the same distance from the reference curve. In many cases it is not analytically feasible, therefore AlfaCAD uses the shadow method, where the "shadow" of the Bezier curve in the form of a polyline consisting only of vectors approximating the model curve, allowing the use of the polyline offset algorithm. By the same method, the "Break" function is performed, where the Bezier curve is replaced with an approximating polyline, and then cut to form two disjoint polylines. In the case of the Offset function, the curve is not replaced with a polyline, it remains a Bezier curve, although its offset copies are based on an approximating polyline (shadow).



Polylines containing intersection points are treated as disjoint polylines, and each cut portion is offset independently, depending on the indicated point through which the copy is to pass, or the side on which the copy will be made at a given distance.



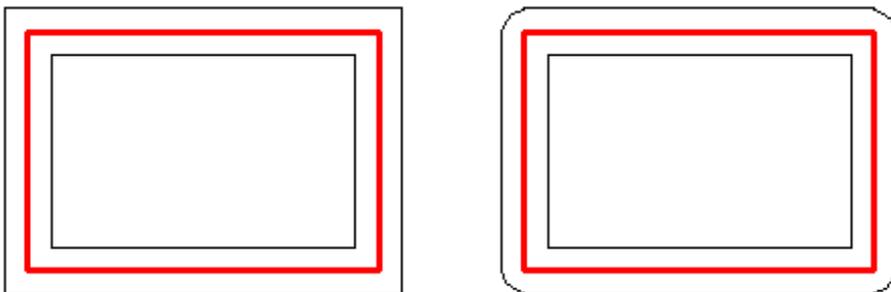


9.2. "Offset" of rectangles, polygons, ellipses and elliptical arc

Rectangles, ellipses and elliptical arcs in AlfaCAD are polylines, created during their construction based on respectively vertices, main axis and secondary semi-axis or centre and two semi-axis, or circular arc scaled with different X and Y scale factors.

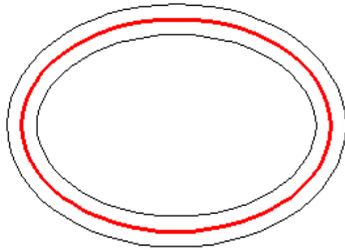
All those object can be freely cut in the Break function or trimmed in the Trim function over its entire length. The "Offset" function is performed in a similar way, identical to any other polyline.

Rectangle offset with Normal and Smooth methods

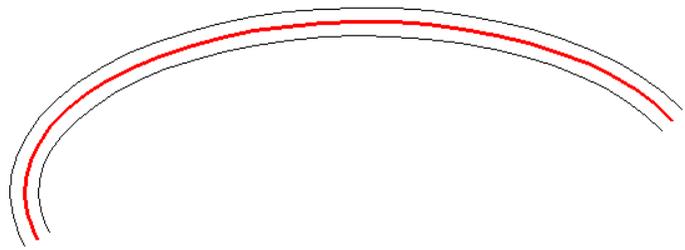


In examples the reference object is drawn with red, thick lines or arcs. Those attributes were changed aftermath for presentation purpose. When "Offset" is performed, reference object attributes (colour, layer, line width and type) are inherited by offset copies.

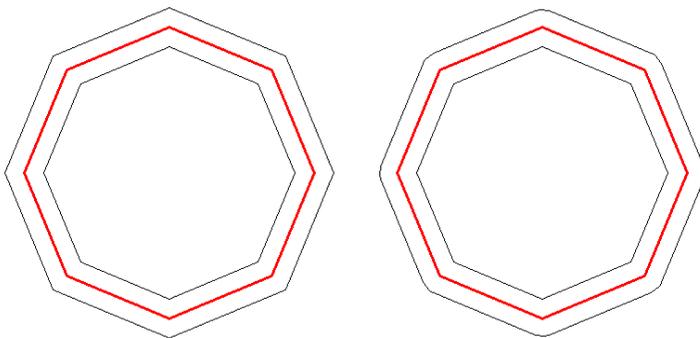
Ellipse offset



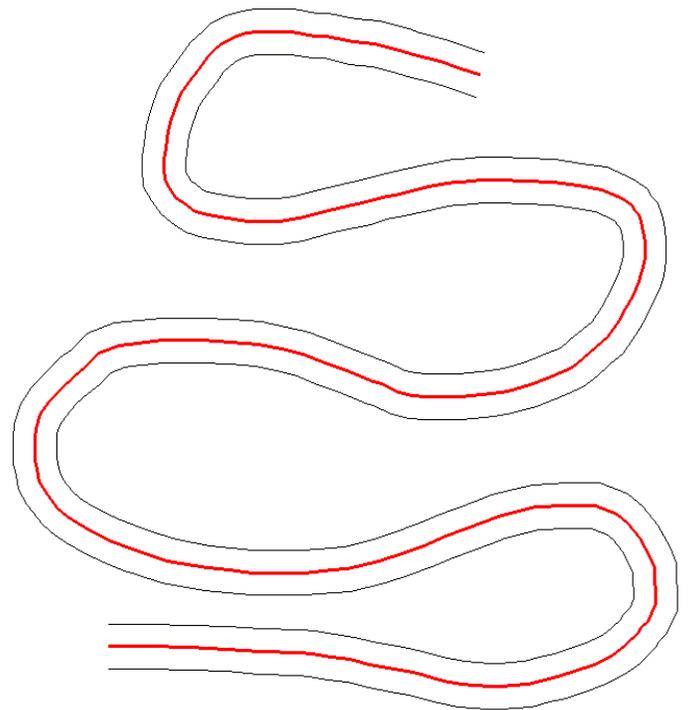
Elliptical arc offset



Polygon (regular octagon) offset using Normal and Smooth methods



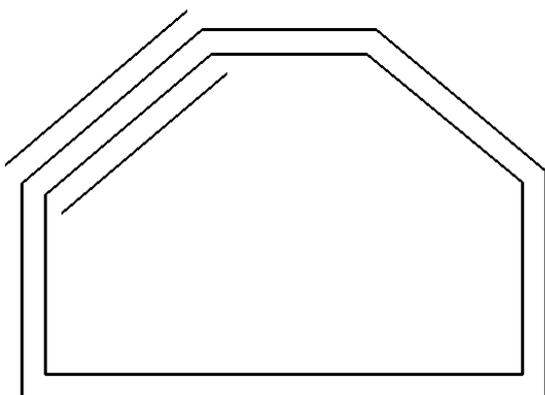
Sketch offset



9.2. "Offset" of 2-line and 3-line trace

2-line and 3-line trace is built from lines and depending of chosen option, whole trace creates quasi-polyline, or is just a set of unblocked lines.

In case of quasi-polyline, offset is limited to the single indicated segment, as shown on the example below:



10. Missing bricks.

ALfaCAD, like other CAD programs, is a complex program. New objects introduced in version 1.6, such as areas, traces and hatches filled with photorealistic patterns, do not have their counterparts in earlier versions of the program. AlfaCAD has always been and will always read all drawings made in previous versions, but drawings made in version 1.6 can only be read by the new version. The older version will read the drawing, but it will not display or print new elements, replacing them with areas filled with colour.

AlfaCAD saves drawings and blocks in proprietary format. The functions of exporting a drawing to DXF format allow you to read the drawing in other programs, although not all objects have their counterparts in this format, and must be replaced by similar or graphically identical but not identical objects. Version 1.6 does not save new objects in DXF format yet, although such a supplement will be included in the next version. However, due to the many corrections to the previous version, the author decided to publish the new version before the completion of the said supplement.

Closing remark

The author stands on his toes to clean up each released version of the program from bugs, which, however, cannot always be avoided, especially when the new version includes new features and significant changes. The author will be grateful for any comments and reports about noticed bugs, which can always be sent to the e-mail address published on the website, and visible in the information about the program.

AlfaCAD was conceived as a simple, universal graphic editor that allows to quickly make all kinds of engineering and other drawings, where 3D modelling is not required. One of the basic assumptions is a kind of minimalism, in particular expressed in a simple interface whose task is to allow access to all program functions in a way that limits as much as possible the amount of information reaching the user from the screen. It is the user who decides what he wants to do and how, and the interface is to facilitate this choice without cluttering the limited space with information unnecessary for the operation being performed.

The author believes that this goal is being achieved, therefore all new functions and extensions fit into the nature of the adopted method of communication between the user and the program.

AlfaCAD does not copy what other similar programs present, both those simple ones and those most complex. Even not long practising using the program, pays off with mastering one or two-character commands that initiate functions fast, many shortcuts and techniques to speed up the drawing process.

The author hopes that this concept finds supporters, who in response find in program the source of their own inspiration.

Enjoy AlfaCAD

Author