AlfaCAD v. 1.4 This is what ergonomics and efficiency are all about… *New features and interface enhancement are also here...*

This particular version (as a successor of ver. 1.2 and 1.3) is dedicated ergonomics, speed and efficiency, together with new features and interface enhancement.
There is one basic rule: the CAD application should deal well with a large number of different interactions, in particular, inserting new objects in most cases located in relation to the position of other objects, repeatedly changing the view scale or panning the drawing on demand, searching the drawing to edit the indicated object and redrawing the content after corrections or significant changes. AlfaCAD makes every effort to achieve this goal.

Interface, menu and shortcut keys

AlfaCAD has many dedicated key shortcuts, or single letters (for all commands as an alternative for choosing function from menu) or key combinations, never exceeding 2 keys, where one can be only so-called shifting key, what can be Ctrl key, Alt key or Shift key, combined with function key F1 to F10, as well as few letter keys and space key. Also single function keys F1 to F11 are involved. Function keys and key combinations are sorted into 4 logical sets:

- single function keys **F1** to **F10** to operate directly on objects (also **F11** for mouse freeing or catching, as alternative to **Ins** key)
- combination with **Ctrl** key for view control
- combination with **Alt** key for alternative choice (like changing property based on the property of another indicated object)
- combination with **Shift** key for positioning object in relation to other one (there are also alternatively single letter keys available for that purpose)

Majority of modern computer applications are equipped with set of functions accessible by clicking buttons or icons or choosing from menu, list boxes or combo boxes settled in tool bars, side bars etc. located around document window, where all creative action takes place. It is convenient, but also has some negative sides. If the screen is small, mostly on notebooks, document window space is stolen by tool bars or side tool window. As long we are thinking about word processing, so text editors, such an interface is mostly very welcome, but when we come to such application like CAD or any graphic editor, including image processing or desktop publishing (DTP) where page layout is created, is quite difficult to create whole layout seeing just a part of the page in reasonable scale to edit it. High screen resolution helps but also for a price, especially for those with not perfect sight. On the very big screen where is probably enough space for document window and for all tool side windows, tool bars or menu bars, during the designing process you make miles and miles of the mouse pointer to reach the function or option and to come back to the place in document window to continue the action after making a choice. Every application is equipped with set of shortcut keys to make that distance shorter. It's simple: using shortcut keys is highly effective, especially in case of the application which we do not use occasionally, rather every day, as our working tool. CAD application is such a tool for designers, engineers, creators, builders, no matter what sort of creation they perform and for what reason.
AlfaCAD's interface is quite different than others. There are no side tool windows, tool bars or menu bar. If there is no action, only document window is visible, where just pointer movement and mouse wheel is involved in the perception of whole document, to see it and analyse for further development. Moving pointer to the edge of window triggers panning the drawing into indicated direction, rolling mouse wheel changes the scale of presentation, where always position of the pointer is the centre of transformation (dilation).

Such a minimalism of the interface is developed for a reason. No not necessary information comes from the screen, leaving the creator alone with his or her creation, to analyse it then decide about the next step to be done, or to go ahead with new objects, or to edit existing one.
At this stage is not necessary to see anything else but the drawing, as big as it can be.
Nearly each function or option in AlfaCAD is represented by a specific icon. Such a small image is easier to recognize than text message like name of the function or option to choose.
AlfaCAD has about 650 icons, each represents different operation. It's impossible to show them all around the document window. Well, not every operation has the same importance or is used as often as another one. Most programs offer mostly used functions settled in menu bars or side bars or tool bars, some programs offer creators to choose most important functions or options to show there. AlfaCAD doesn't do that. There is no menu bar, no side tool window, no bottom tool bar. All functions are located in floating vertical menu or auxiliary horizontal menu accessible by the single click of the left mouse button in any place of the document window. Menu will appear close by, no need to focus you sight on another part of the screen, all is close to the pointer. When menu is open, position of the pointer doesn't change, leaving menu leaves pointer position unchanged, still in the place located before, where possibly object is moved, or specific point is marked for creation of the new object or editing existing one.

When any menu is open, the menu bar is moved by mouse independently of the previous position of the pointer. Mouse wheel is also engaged to scroll menu, if is scrollable, together with arrow keys **Up** and **Down**, to move the bar by one position, or **PgUp** or **PgDn** to move “the page” of scrollable menu or **Home** and **End** to move to the beginning or the end of menu. In all cases, when menu is showing icons, there is a shortcut key indicated to choose function or option simply by pressing a single key. In the practice, no menu has to be specifically open (it will be open momentary anyway) to choose any function. To start drawing a line is enough to press D then L keys. D (Draw) L (Line). To draw circle: DC, to edit text: ET, to hatch area: H, to save current file FS (File Save) etc. No needs to observe menu on the screen. It's enough to just press two, one by one, in few cases even one single letter, to access desire function, and all without reposition the pointer, which is mostly focused on the object being created or modified.
Right mouse button (RMB) or Esc key leaves the function, aborts any action, if there is staging action, RMB or Esc leave the stage moving back to the previous one. Successive Esc pressing returns to the start position where there is only you and your drawing.
Whatever action is undertaken, any function, there can be auxiliary menu engaged with extra options, open by Space key or middle mouse button (MMB) pressing. There are plenty of actions and options available at any time, like point locations, zooming, changing object properties, searching for texts, changing drawing methods, what is object type dependent, so in part it's the context menu.
Since version 1.4 this feature is even enhanced to let you choose directly the context auxiliary menu for current operation by pressing **Ctrl-Space**, which is going directly to the set of options or functions specific for current action, like drawing specific object, or editing it on specific way or any other.
During any action, drawing, editing or block operation, set of auxiliary shortcut keys is engaged, and the set can vary depending of the action. Auxiliary keys can be the single letter or function key (F1...F12) or it's combination with shifts, so Ctrl, Alt or Shift keys.
It's quite easy to memorize them all. At any time key F1 shows “Help” menu with all key shortcuts, not only to remind what each of them is doing, but also actually doing it, by choosing the key from menu, even for demonstration purpose, what is practical anyway when specific key combination was forgotten.
If the specific icon in menu doesn't suggest well enough what that it is for, is enough to wait 2 seconds until tip window will appear next to the menu position with short explanation what this function is for. If the position of the menu is dedicated for the change of any parameter, the currently set parameter value is shown in tip window too.

When so many shortcut keys are available for choosing functions (commands) or options, and when using those keys is the fastest way to progress the work on drawing with minimal effort to do that, it's quite typical to use both hands during your work with AlfaCAD. While one hand operates with mouse and its 3 keys and wheel, second hand is engaged with keyboard (like shown on the image below), choosing functions by single letter key, or selecting options or functions normally accessible from hierarchical menu (mostly not exceeding two levels of the hierarchy, just in rare situation reaching 3 levels) by single function key or its combination with one of the shifts.

(photo by Ariel da Silva Parreira)

It reminds the HOTAS (an acronym of **H**ands **O**n **T**hrottle-**A**nd-**S**tick is the concept of placing buttons and switches on the throttle lever and flight control stick in an aircraft's cockpit):

in the jet fighter:

and racing aircraft:

(photo by Joerg Mittel)

While editing single line text (DT (Draw Text) or ET (Edit Text)) or multiple line text (DM or EM) it's very likely both hands are engaged with keyboard, like on the image below:

(photo by Ariel da Silva Parreira)

This is basically the only moment when one of hands (right for right-handed) is loosing contact with the mouse, except of the situation when shortcut keys combination brings a problem to use it with single hand, for example Crtl-F10 on the keyboard with single left Ctrl key. There is simply too far away from one key to other in the combination, if you are not trained piano player. Even any combination of function keys with shifts brings some troubles, what eventually can lead to abandoning the idea to use such combination and choose function from the menu, what is not that effective as using shortcuts. If that would be the issue, so-called “sticky keys” bring the solution.

Sticky keys

“Sticky keys” is not a new idea at all. This feature was introduced yet in 1988 in some graphical user interfaces originally designed to assists users who have physical disabilities or help users reduce repetitive strain injury. It serializes keystrokes instead of pressing multiple keys at a time, allowing the user to press and release a modifier key, such as Shift, Ctrl, Alt, or the Windows key, and have it remain active until any other key is pressed.

Sticky keys functionality is available in Microsoft Windows and macOS as Sticky Keys and on Unix/X10 systems as part of the AccessX utility.

Sticky Keys was first introduced to Mac OS in System 6. In 1994, Solaris 2.4 provided sticky keys. Microsoft introduced Sticky Keys to the Windows platform in Windows 95.

As a feature designed for users who have physical disabilities is mostly ignored by other users. In fact, this feature not necessary is very welcome in some applications, so quite often it has to be switched on and off (e.g. by pressing Shift key 5 times).

To not interfere with system sticky keys, AlfaCAD is now equipped with own sticky keys system, very easy, and applicable only in the situation where is useful.

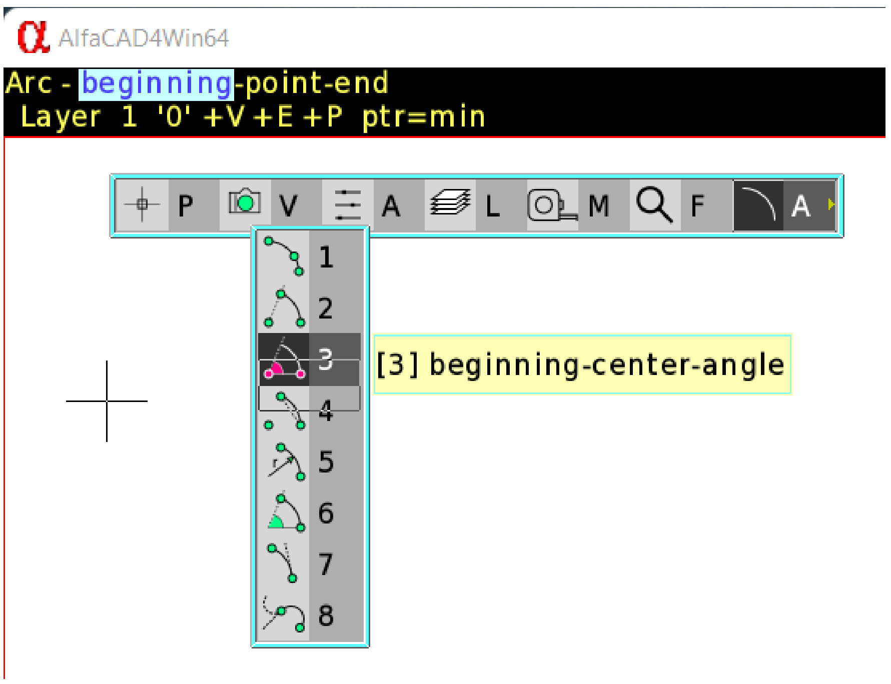
Pressing Ctrl, Alt or Shift key (no matter right or left one, also AltGr as an alternative to Alt key) the icon of the particular key is shown at the bottom left corner of the desktop. Program takes in to account already pressed shifting key, so is enough to press second key from the combination to execute function. If any other key which doesn't belong to any possible combination is pressed, or shifting key is pressed again, icon disappears and no action is undertaken.

Same sticky keys are active while text is edited both for single or multiline text, however, the purpose of using them is to simplify selecting special characters, like Greek letters and extra maths symbols.

Using sticky key is optional, always regular combination of shortcut keys can be used, like Ctrl-10 for View Drawing, or Ctrl-Q for changing orthogonality and so on.

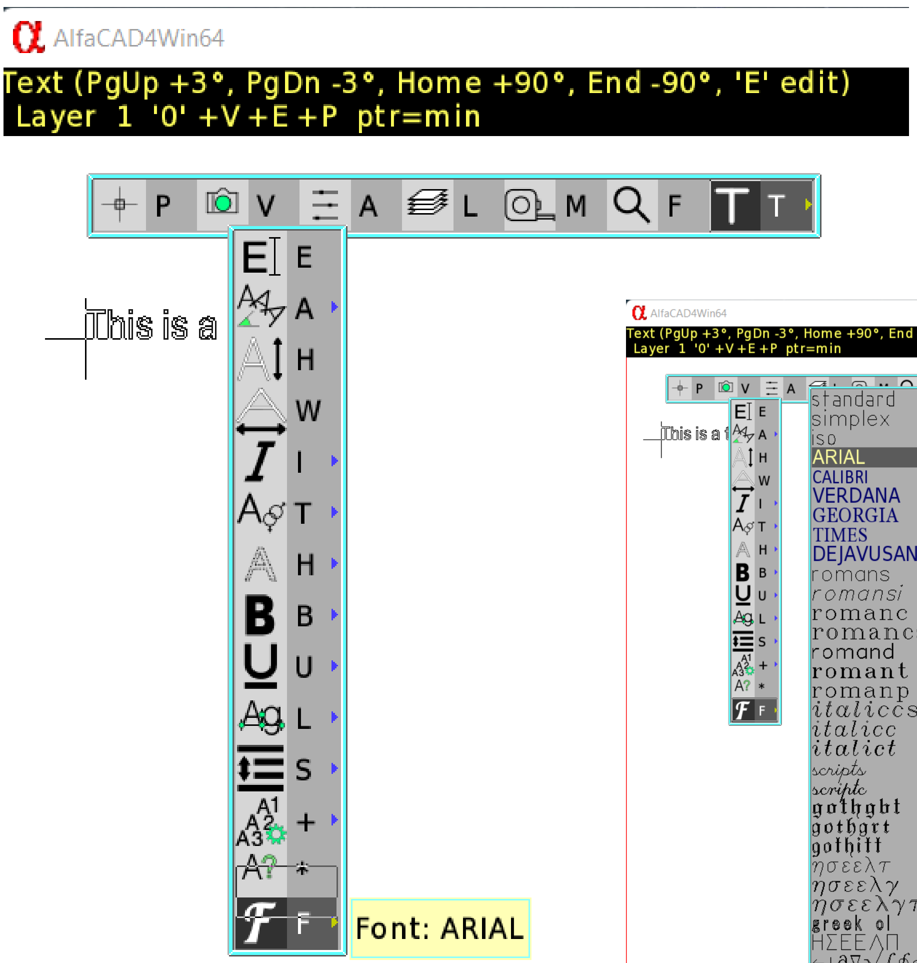
Auxiliary menu

Almost every function comes with set of options and auxiliary procedures, accessible in auxiliary menu triggered by **Space** key or middle mouse button **MMB**.
There are fixed set of groups of functions: **Point, View, pArameters, Layers, Measure** and **Find** as well as context group of functions appearing as a last option on auxiliary menu.
Context menu can contain drawing, editing options or selecting options and many others.
For example, function Draw Arc comes with set of options to construct arc on 8 different ways:

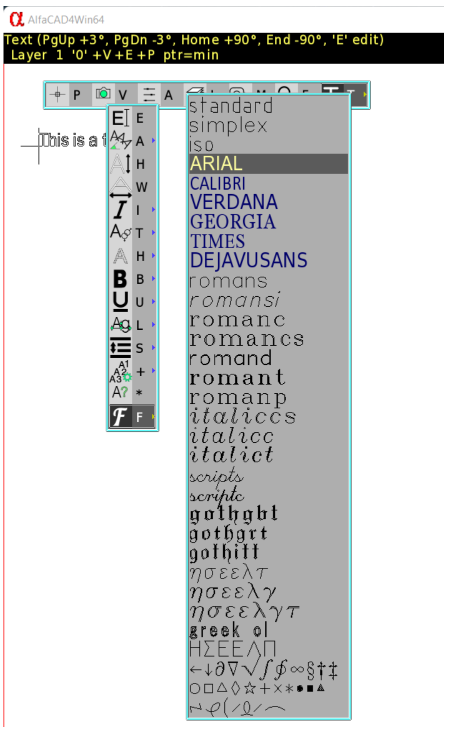

AlfaCAD4Win64

Arc - beginning-point-end
Layer 1 '0' +V +E +P ptr=min

To see those options, Space or middle mouse button ({MMB}) has to be pressed, then last option (A – for Arc) chosen. The shortcut key varies depends of currently performing operation.
In another example, Draw Text, to change text parameters the same Space or {MMB} has to be pressed then T – for Text last option menu chosen:


AlfaCAD4Win64

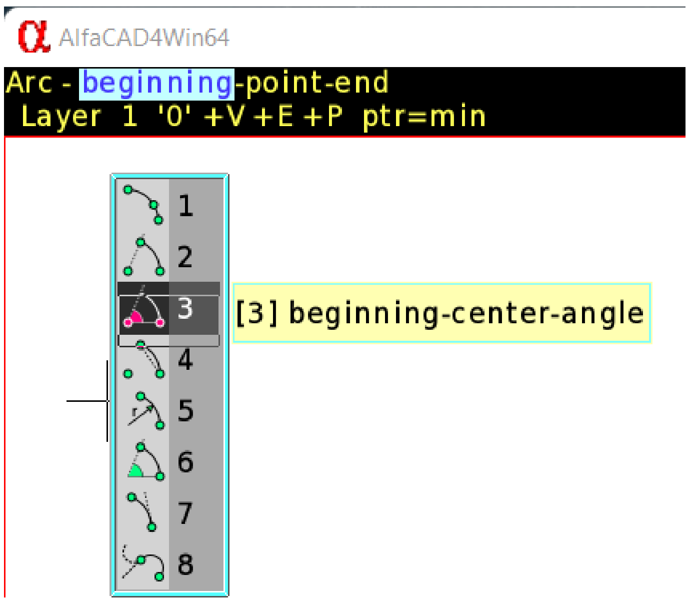
Text (PgUp +3°, PgDn -3°, Home +90°, End -90°, 'E' edit)
Layer 1 '0' +V +E +P ptr=min


AlfaCAD4Win64

Text (PgUp +3°, PgDn -3°, Home +90°, End -90°, 'E' edit)
Layer 1 '0' +V +E +P ptr=min

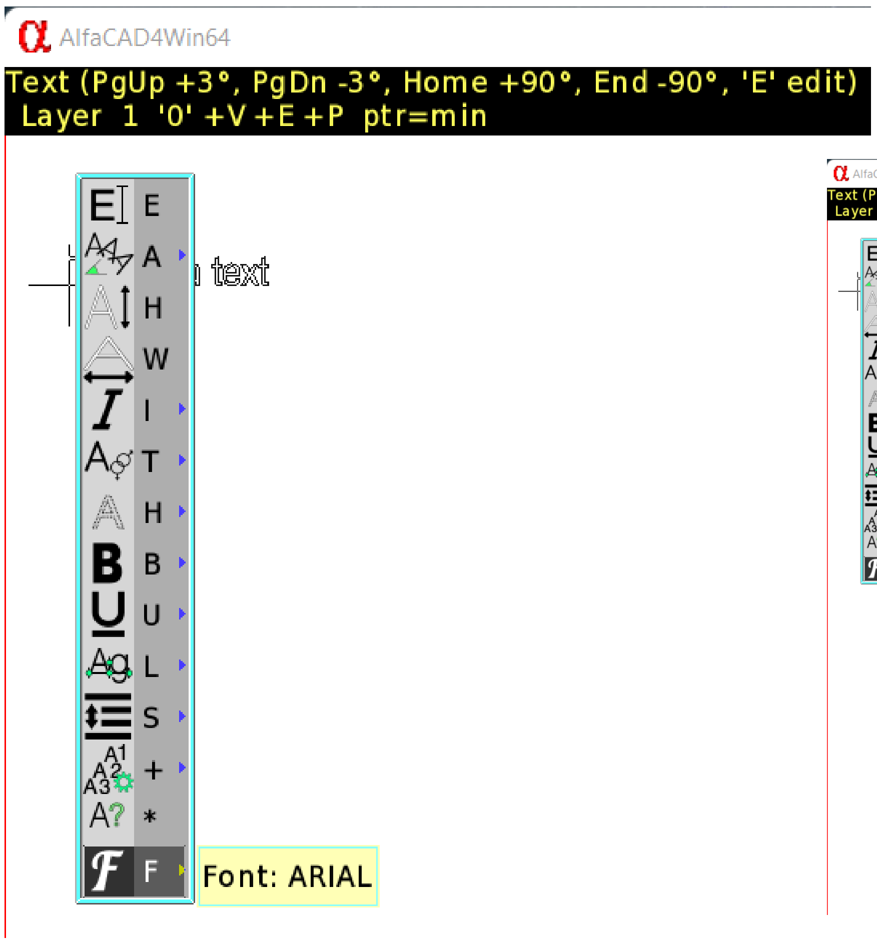
and then can be selected any parameter to change, e.g. font, as shown on the miniature image above.

To make such choice easier, is enough to use **Ctrl-Space** shortcut, which is opening context menu directly, omitting auxiliary menu. So when Draw Arc is a current function, when A option appears in auxiliary menu, Ctrl-Space brings options for arc:

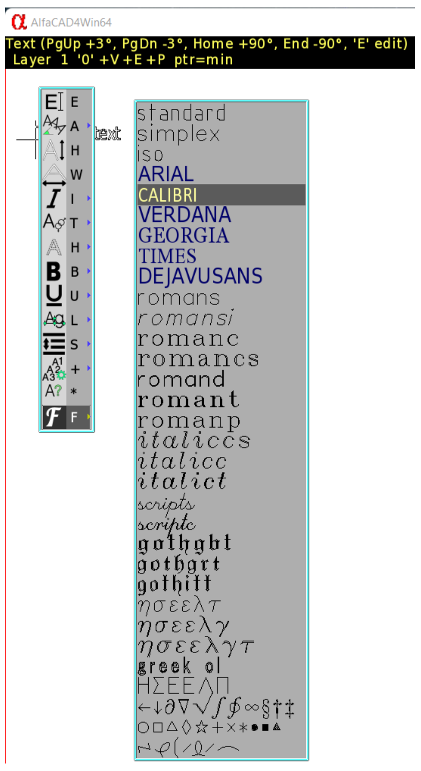


```
AlfaCAD4Win64
Arc - beginning-point-end
 Layer 1 '0' +V +E +P  ptr=min
```

and for Draw Text:



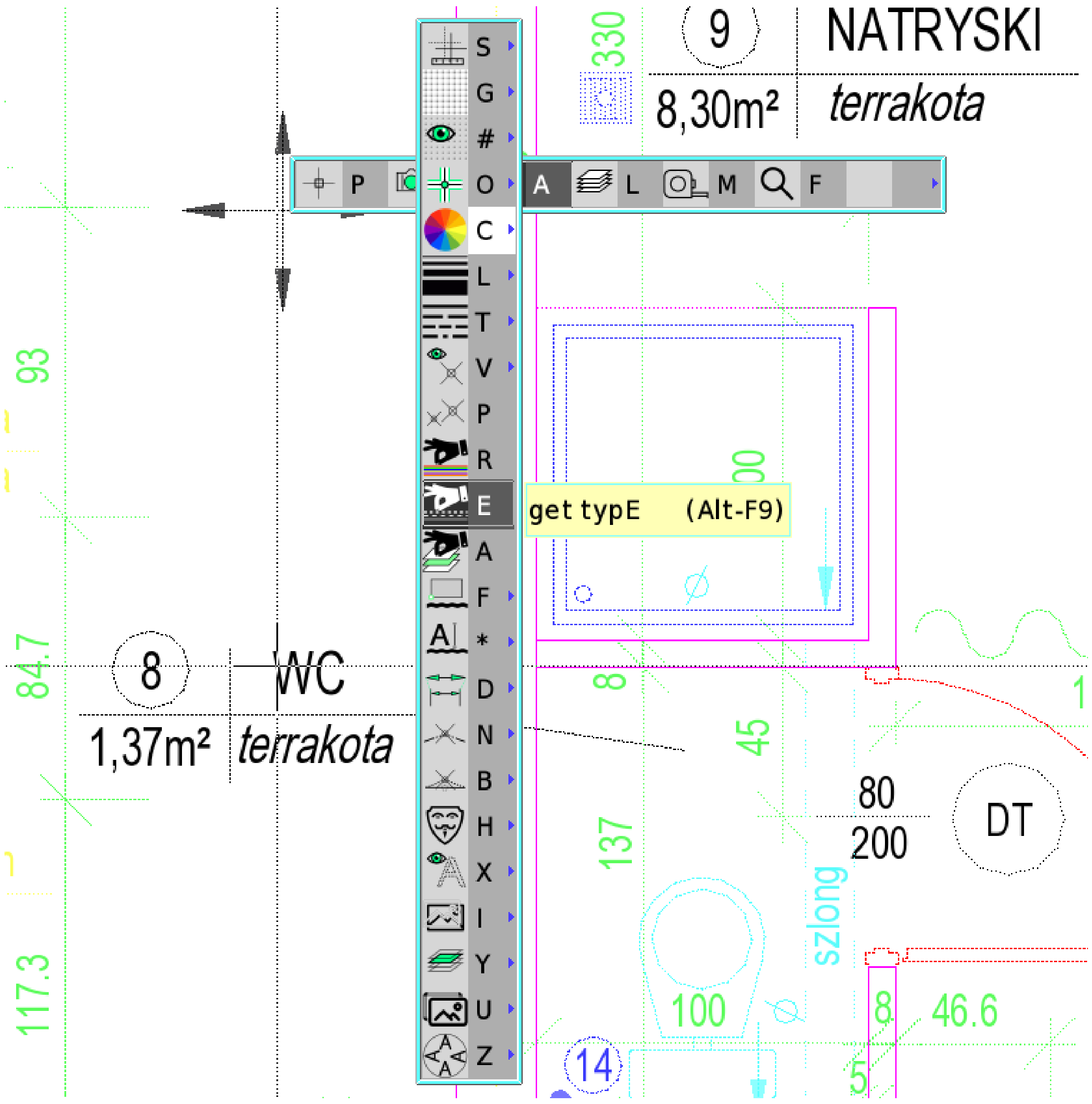
```
AlfaCAD4Win64
Text (PgUp +3°, PgDn -3°, Home +90°, End -90°, 'E' edit)
 Layer 1 '0' +V +E +P  ptr=min
```



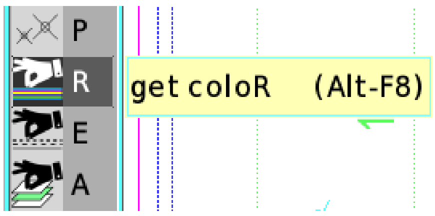
```
AlfaCAD4Win64
Text (PgUp +3°, PgDn -3°, Home +90°, End -90°, 'E' edit)
 Layer 1 '0' +V +E +P  ptr=min
```

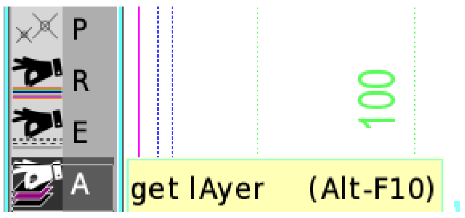
where like in previous example any option for example font style can be selected, as is shown on the miniature image above.

A large number of parameters are available to change. They are separated, can be selected one by one, but there it a very rare situation we have to change many parameters at once. If however it is necessary, maybe more practical is to use “get typE” function from auxiliary menu:



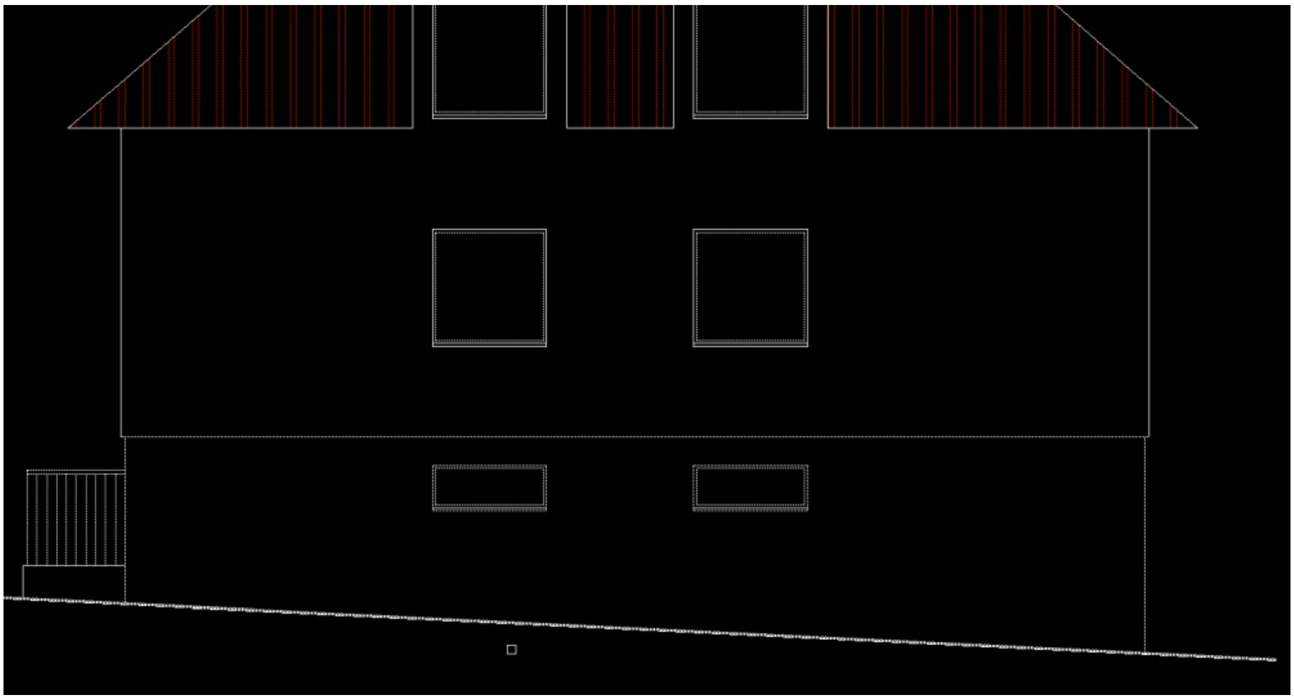
then you can grab all parameters (of the text, or line, or arc or circle, or any other existing object), except of the layer and colour, what is reserved for another two functions in the same menu: “get colouR” and “get lAyer”. The capital letter always is indicating the shortcut key, no matter the word is orthographically correct or not, because not always first letter of the function name can be used as a shortcut key. It's just to remind the key especially for users with developed visual memory. In this example on the image above „get Type” cannot be used because function “line Type” already uses that letter.



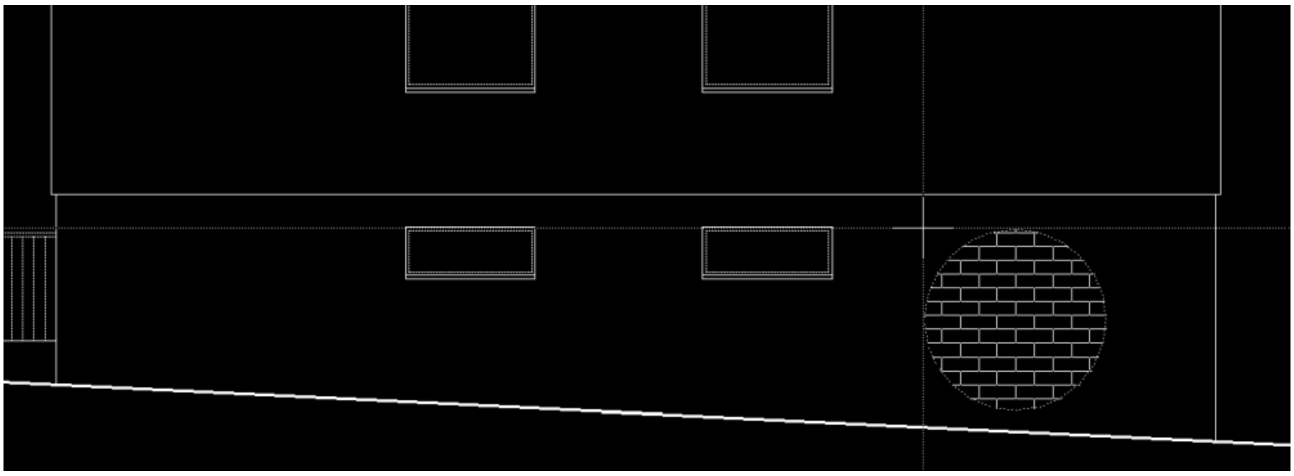


Hatching and hatching editing

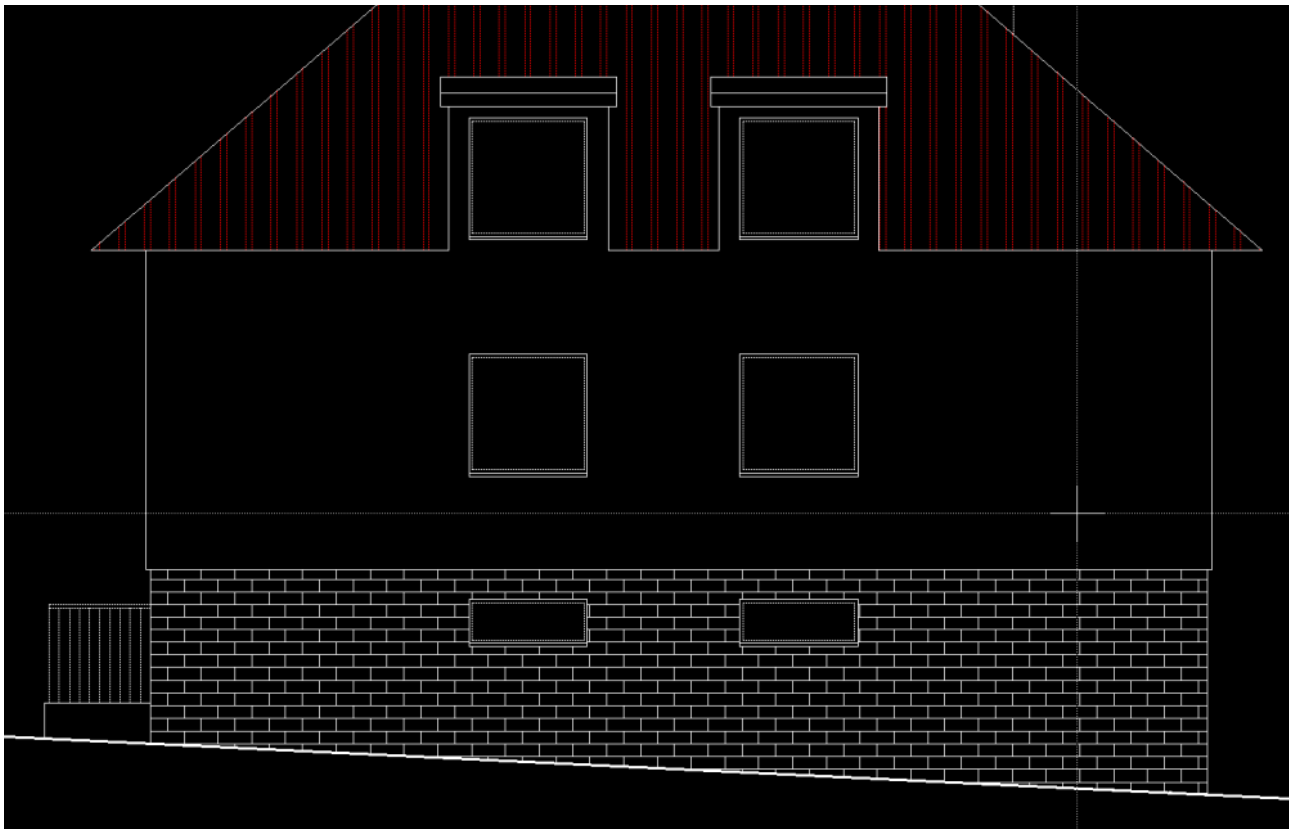
Since ver. 1.2 AlfaCAD is equipped with enhanced hatching function, which not only allows to hatch with selected pattern or fill with selected colour the enclosed area of practically any irregularity while still mono-consistent.
The improvement concerns the preservation of the boundaries of the hatch area as the hatch was originally created. As a reminder - hatching is created as a block of lines drawn according to a selected pattern. Now this block also contains hatch boundaries, although region boundaries have an attribute that allows you to display or hide region boundaries. The boundaries of the area, regardless of being displayed or hidden on the screen, are not printed or plotted, but are only used to re-edit the hatch, which in practice means automatically deleting the existing hatch block and replacing it with a block generated according to a different, selected pattern. This relieves the user from re-marking the boundaries of the hatch, speeding up the entire operation, especially for complex boundary shapes.
Let's do an easy example. Already selected boundaries of hatching the wall are marked with dotted lines:



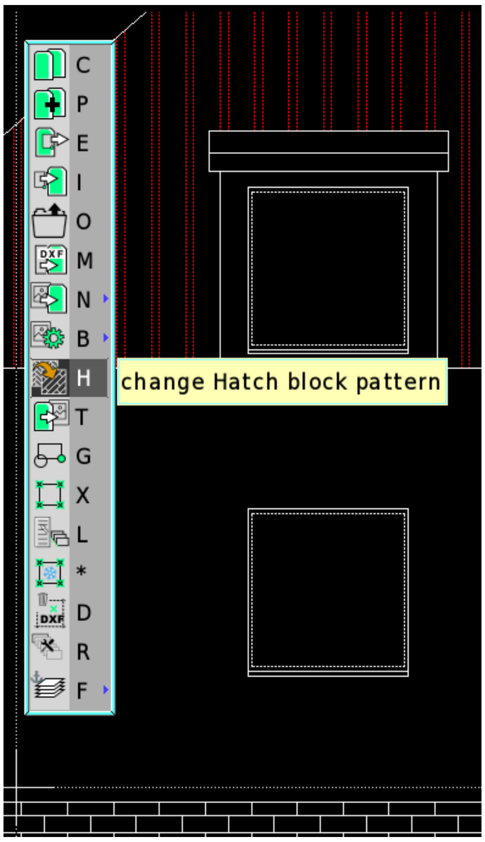
Selected “AEC-brick” hatching pattern appears next to the pointer, and AlfaCAD is waiting for indicating any point inside enclosed area, as a seeding point for hatching:



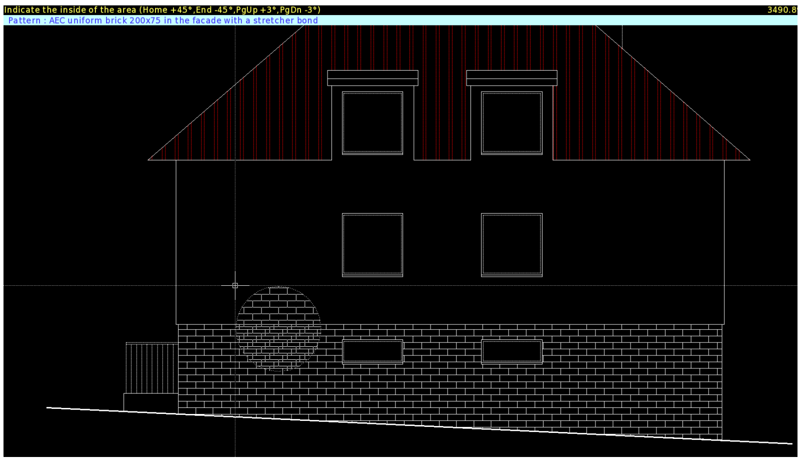
A masonry bricks hatching is generated:



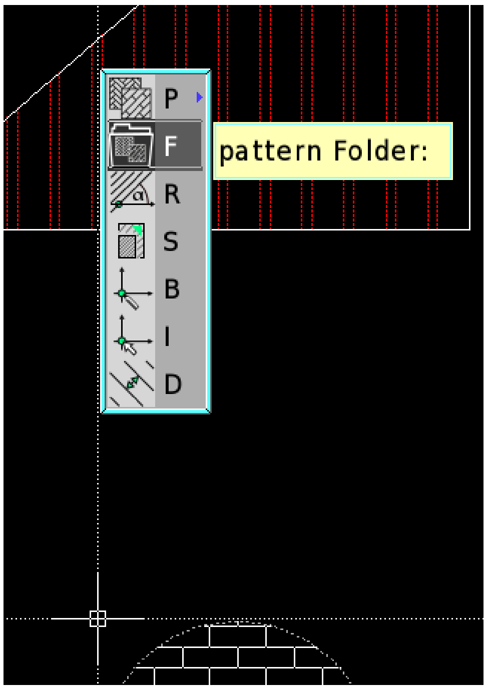
When we change minds, and instead of masonry bricks we prefer to use stone-masonry, is just enough to select “**Block**” – “**change Hatch block pattern**” function:



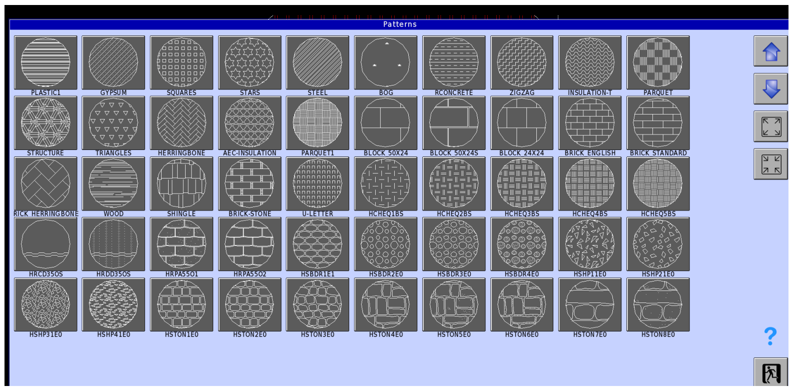
Initially the same (or rather previously used) hatch pattern appears next to the pointer, also offering quick change the angle of the pattern by pressing Home (+45°), End (-45°), PgUp (+3°) or PgDn (-3°) key:


Indicate the inside of the area (Home +45°,End -45°,PgUp +3°,PgDn -3°)
3490.8
Pattern : AEC uniform brick 200x75 in the facade with a stretcher bond

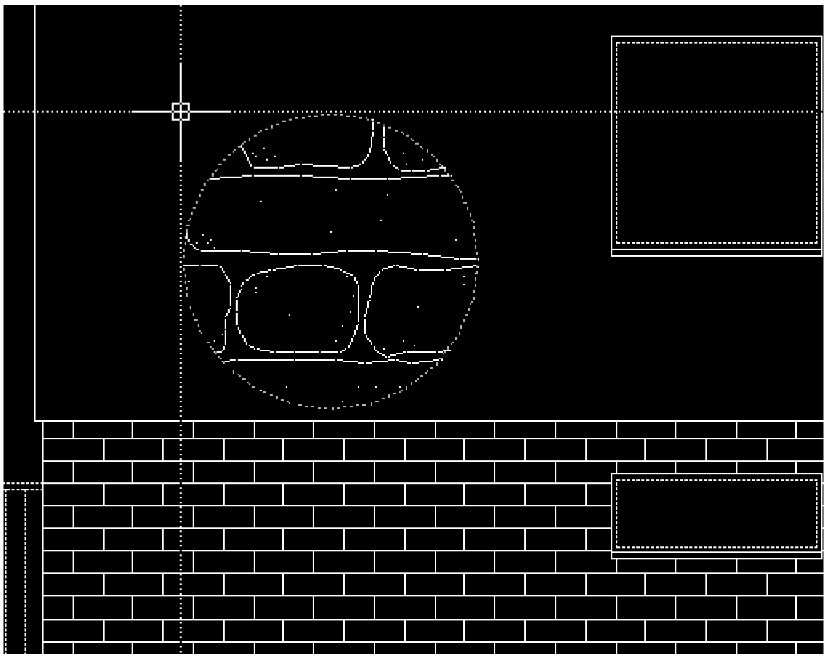
To change any parameter, including hatch pattern, auxiliary menu needs to be open by pressing Space or MMB, then **Hatching** option, or simply Ctrl-Space to open Hatching menu:



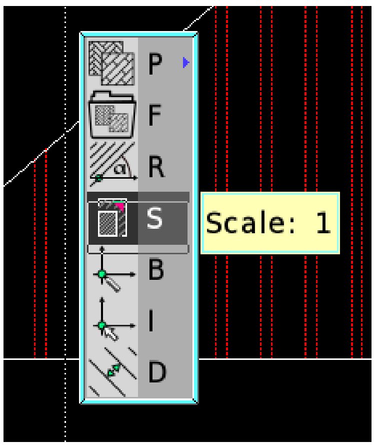
New hatching pattern can be selected from the list (Pattern) or from the catalogue of patterns (pattern Folder):



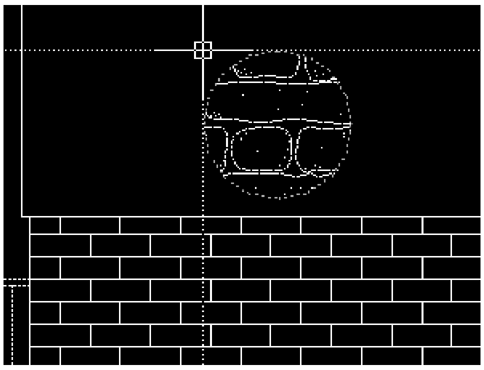
Let's select HSTON8E0 pattern, as a stonework patter with rounded stones and mortar with sand:



Visual evaluation tells us the pattern is too big. We can select Scale from the Hatching menu:

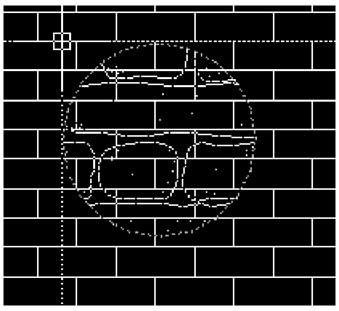


Changing scale down to 0.5 seems to be enough:

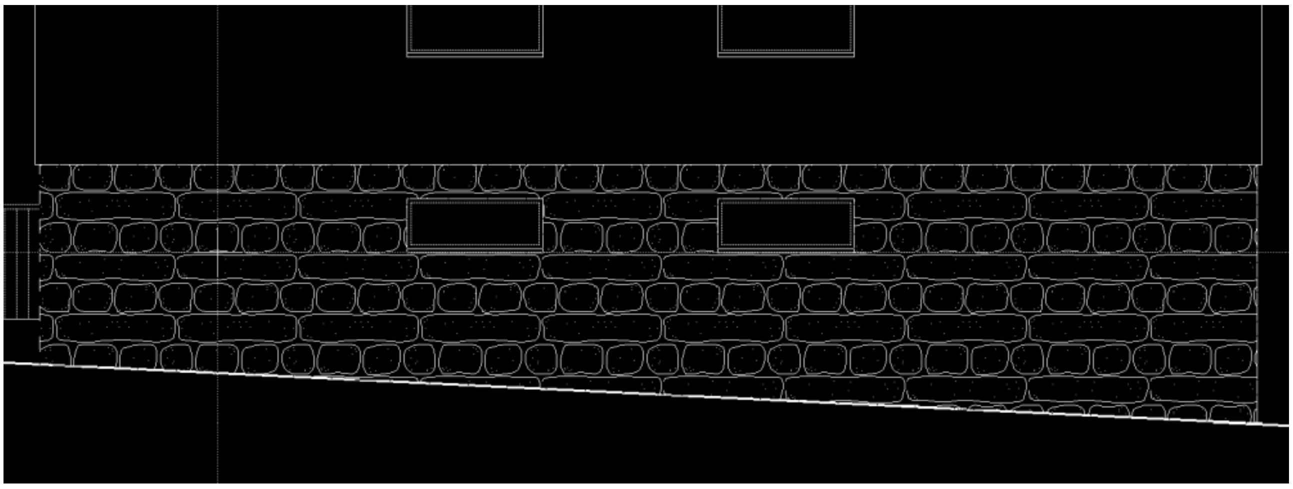


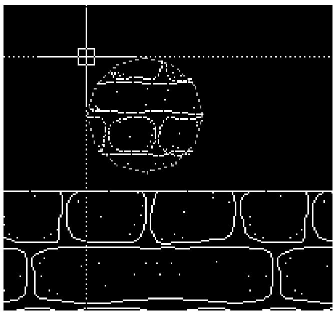
So now time to replace existing hatching block with the new one, based on new pattern. To do that, it's enough to select any line of hatching block, using pointfinder box.
Like always, with any selecting operations, the size of pointfinder can be changed. For that purpose 2 keys are dedicated: **PgUp** to increase and **PgDn** to decrease the size of the box. But here is an exception: PgUp and PgDn keys are already used to change an angle of the pattern. Here one trick comes with help in case of necessity to change pointfinder box size.

To divert the meaning of PgUp and PgDn keys from changing pattern angle to changing size of pointfinder box, is enough to select **Point** from auxiliary menu, and select e.g. **Endpoint**.

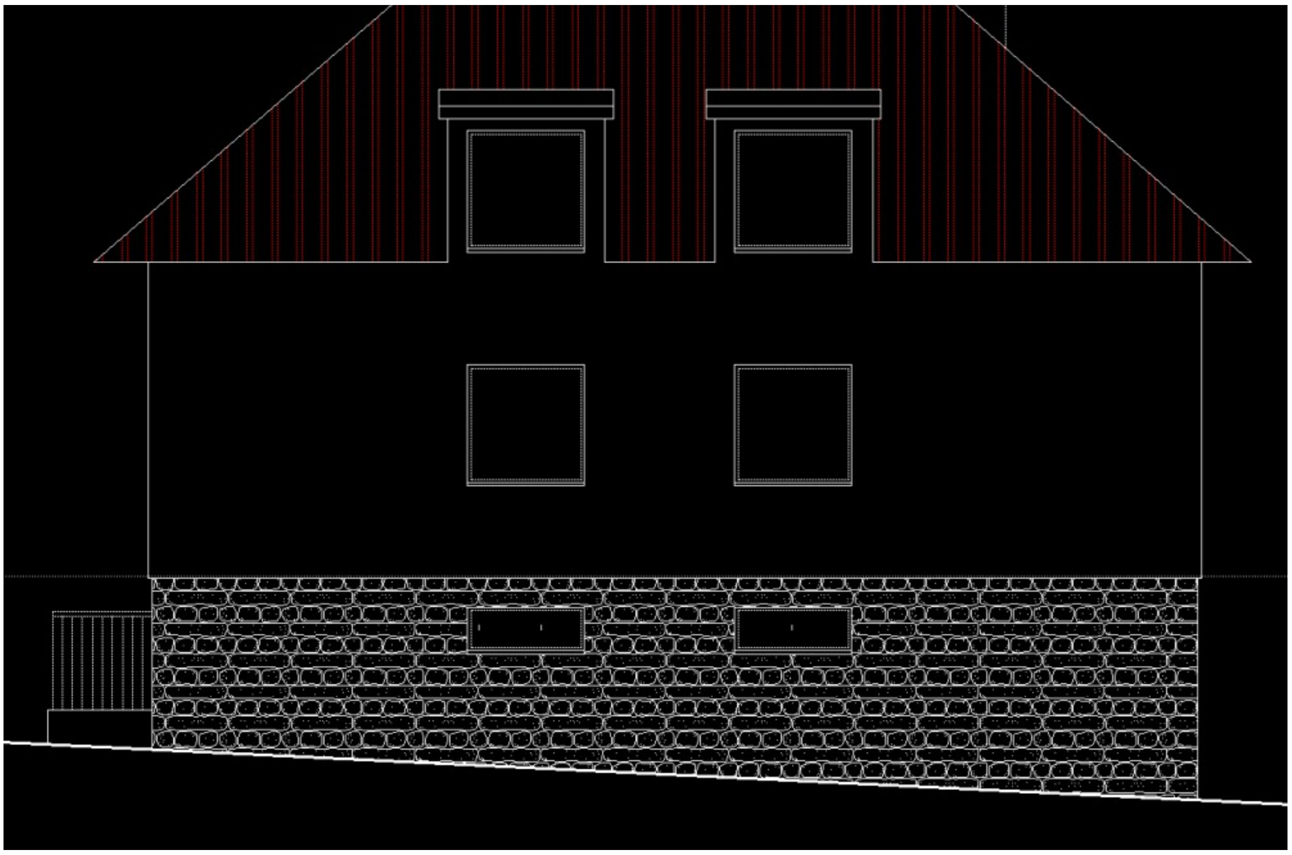


When any line from hatching block is selected:
all lines of the block are going to be removed, boundaries defining an enclosure are going to be found and used to generate new hatching block. Not only new pattern will be used, but entire set of parameters, including scale, angle and base point, however, current line colour, current line type and width, layer number on which previously generated hatching block was placed will be preserved.
And here is an effect:



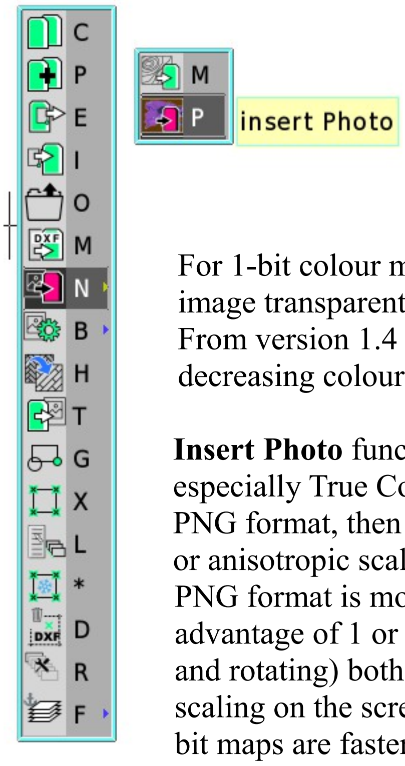


If original hatching block is an element of another bigger block, all block structure will be preserved too. In case of complex block structure in which hatching block is involved, changing hatching pattern can take much longer than new hatching creation.
If generated hatching is still not satisfying, any parameter can be changed again (e.g. Scale to 0.3) and the procedure can be repeated.
Here is a final effect:



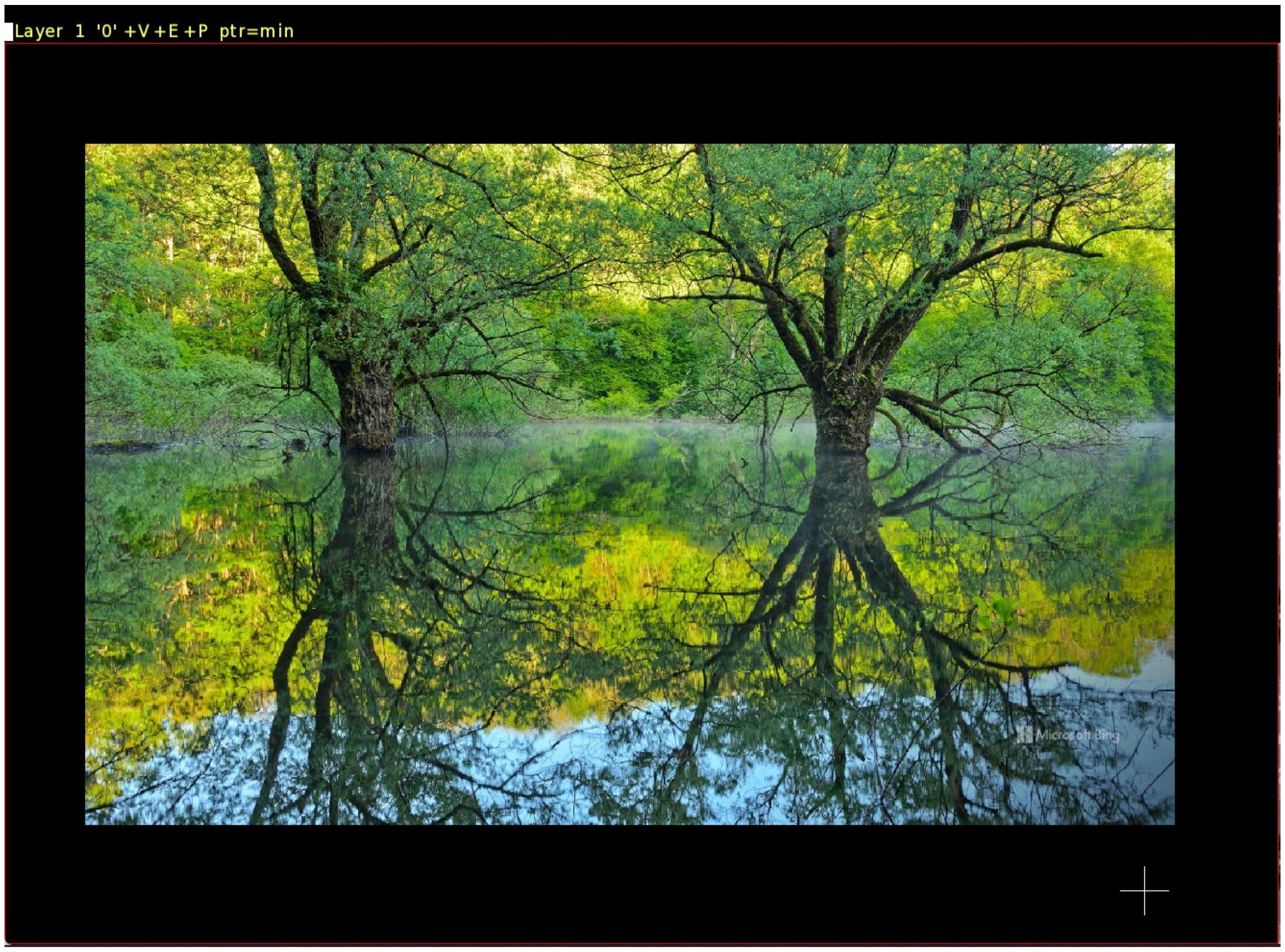
True Color images

Function **Block** – **iNsert image** is splitted now into two function: **insert Map** and **insert Photo**.

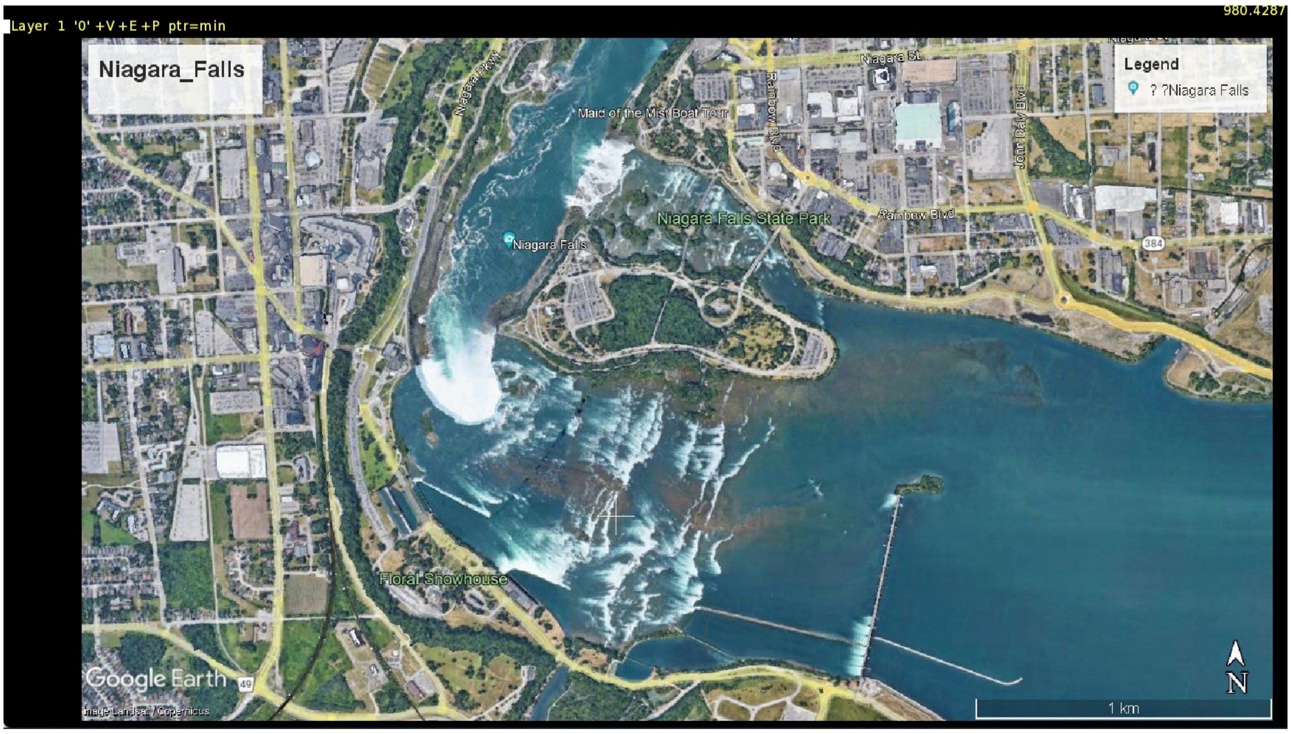


Insert Map function is identical with previous **insert Image** function, and is intended for import 1 or 8 bit per pixel images as geodetic sleepers and maps, where any graphic format file is converted in lossless compression to PCX format, then imported to the drawing, with set parameters like isotropic or anisotropic scale and angle.
For 1-bit colour maps sometimes ignoring background colour is practical, making the image transparent, when only pixels other than background are displayed.
From version 1.4 also 24bit colour images, as “maps” can be imported without decreasing colour depth.

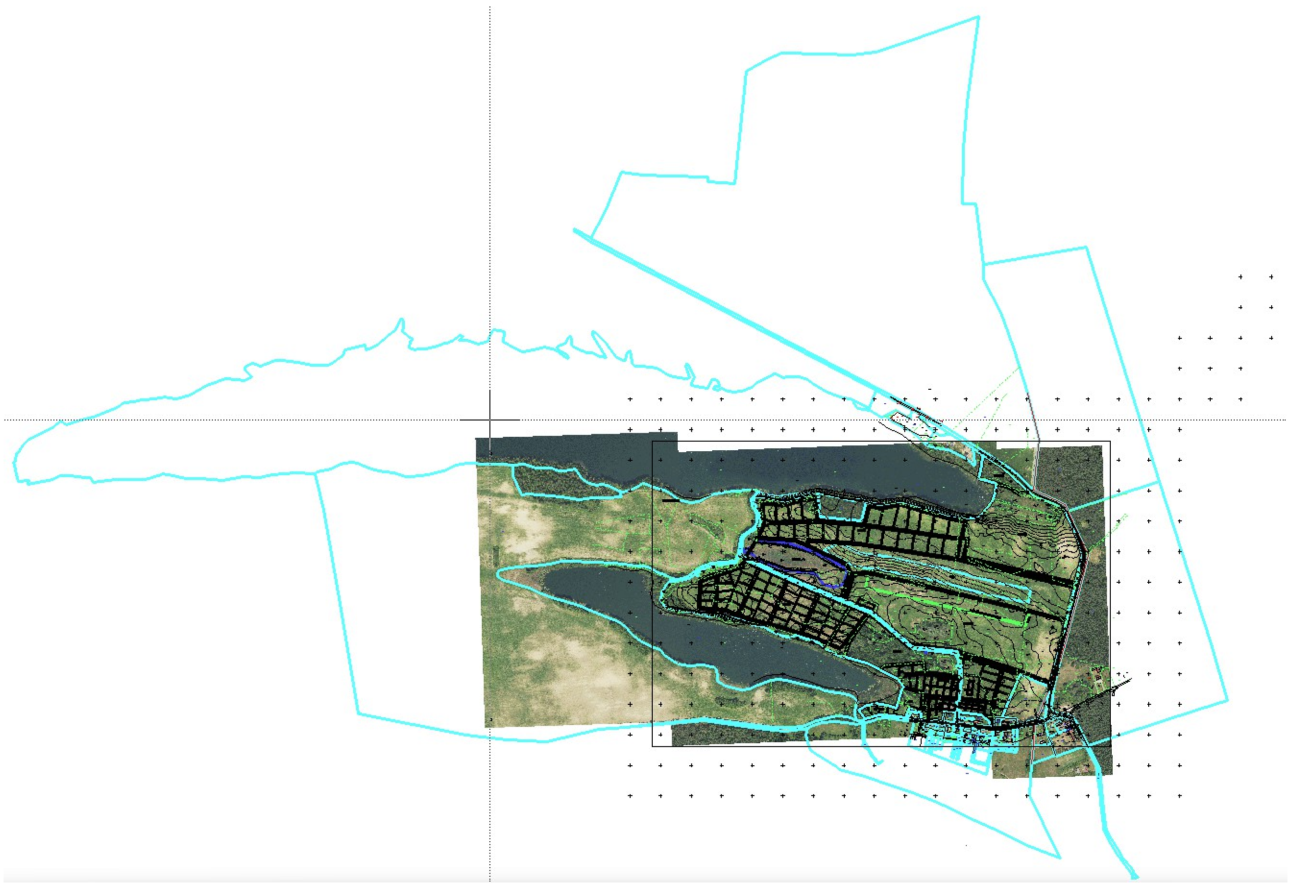
Insert Photo function is intended to import any graphic file, in any colour depth, especially True Color photos, which will be converted in lossless high compression to PNG format, then – similar to insert Map - imported to the drawing, with set isotropic or anisotropic scale and angle.
PNG format is more versatile than PCX with higher compression ratio, however, the advantage of 1 or 8 bit maps is slightly higher precision of transformation (scaling and rotating) both on screen operations and printing, as well as higher range of scaling on the screen with extreme zooming operations. Additionally, transparent 1-bit maps are faster in display functions.
Both formats are subjects of image buffering dramatically speeding up screen operations like panning and zooming. Here is a test image:



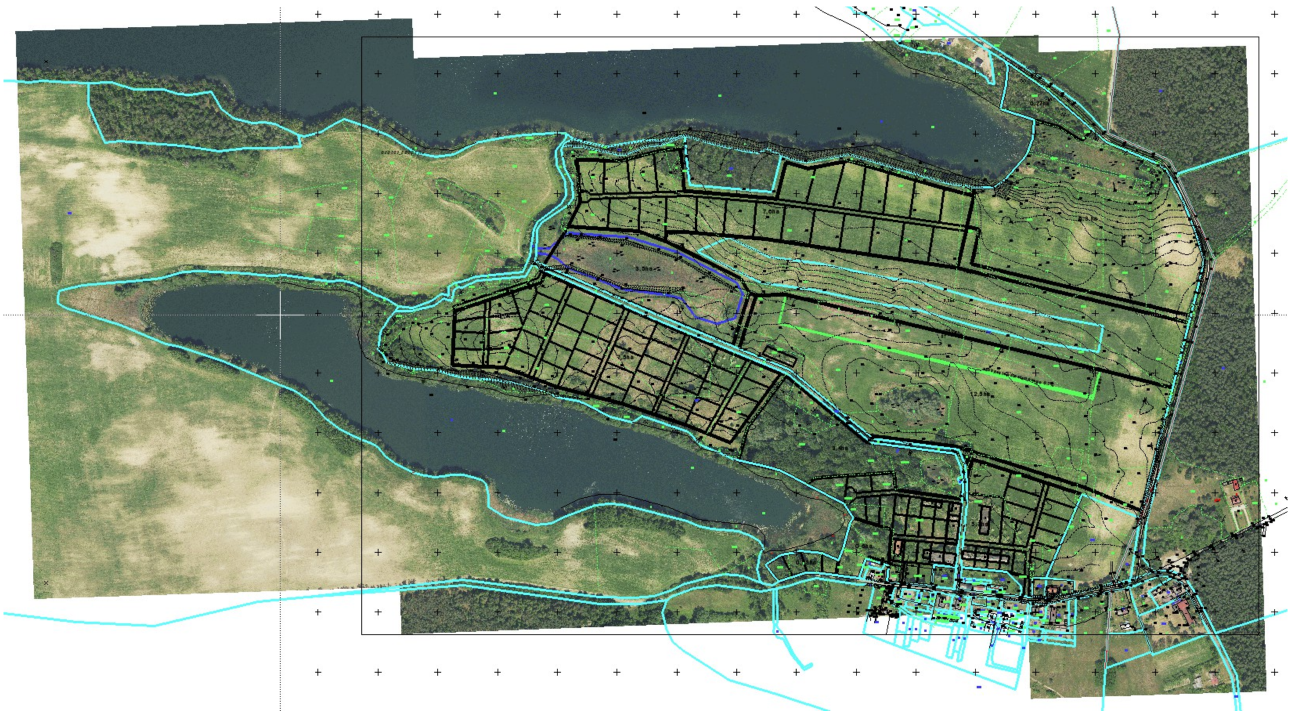
or from Google Earth:



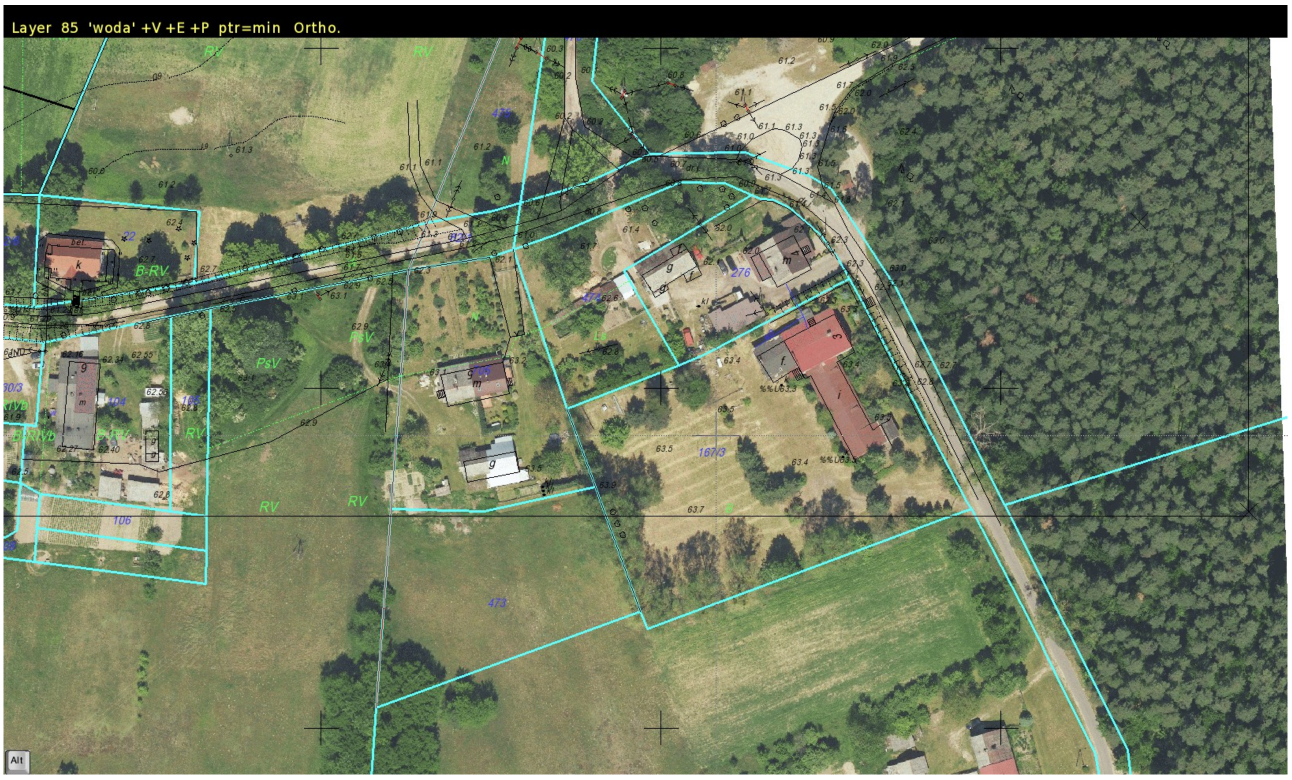
or something really serious, like photogrammetric or aerial photos for surveying combined with digital map:



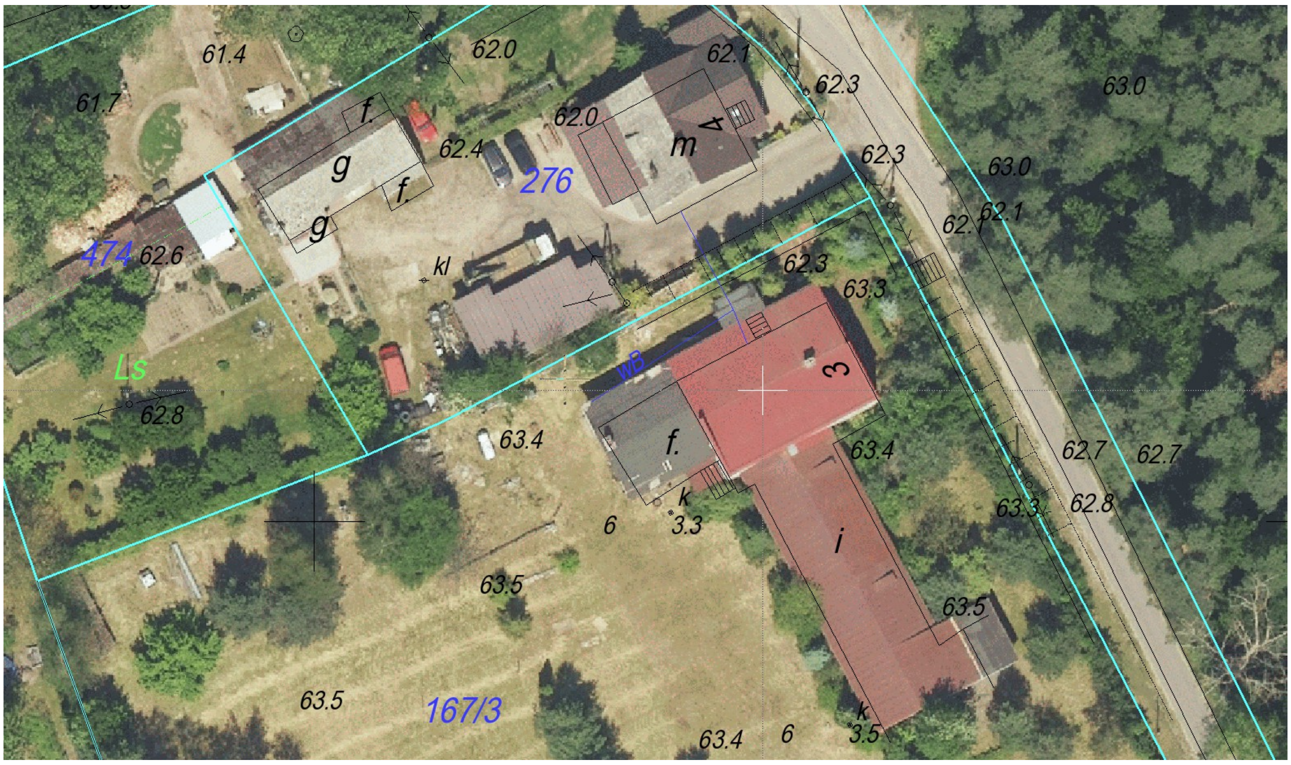
In this example, 3 aerial high resolution photos were stitched together (using calibration offered by AlfaCAD), where anisotropic scaling and rotation is involved. High resolution images 8,500 x 12,000 pxl, 13,700 x 12,000 pxl and 5,000 x 13,500 pxl are buffered to make panning and zooming operations lag-less.



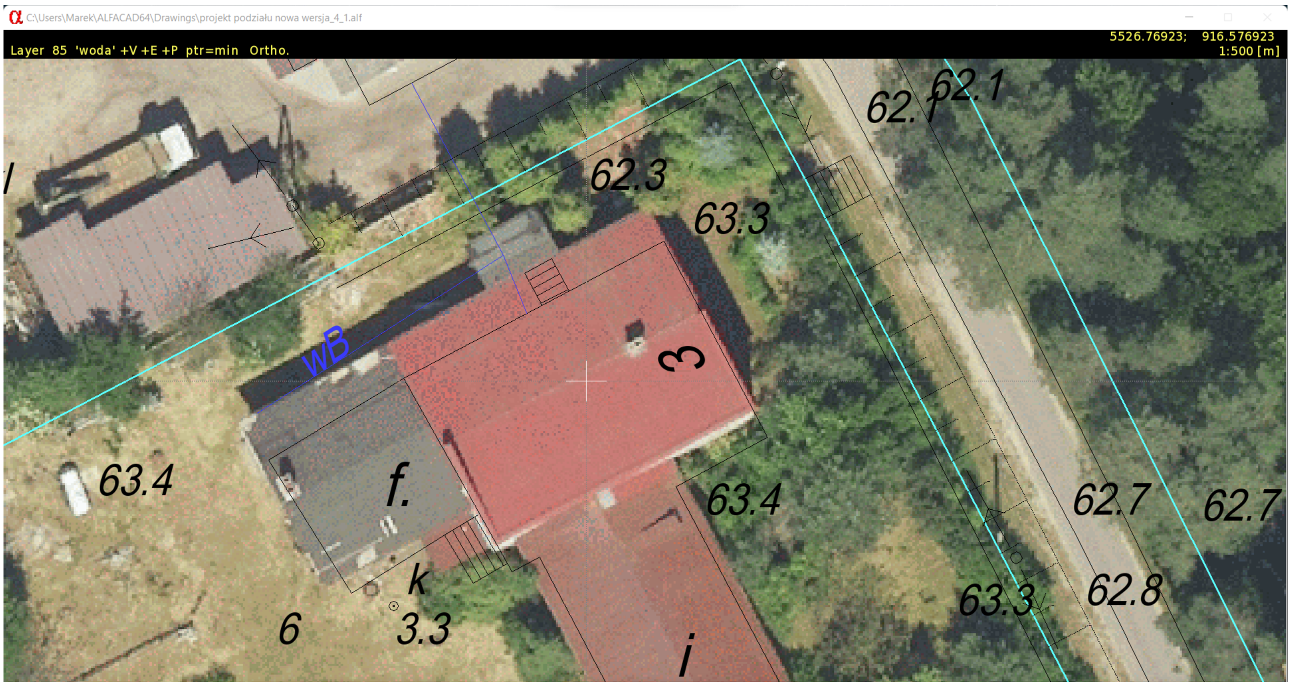
Zooming operations let come close…



and closer …



even closer with buffering (buffering brings some large-scale inaccuracies).

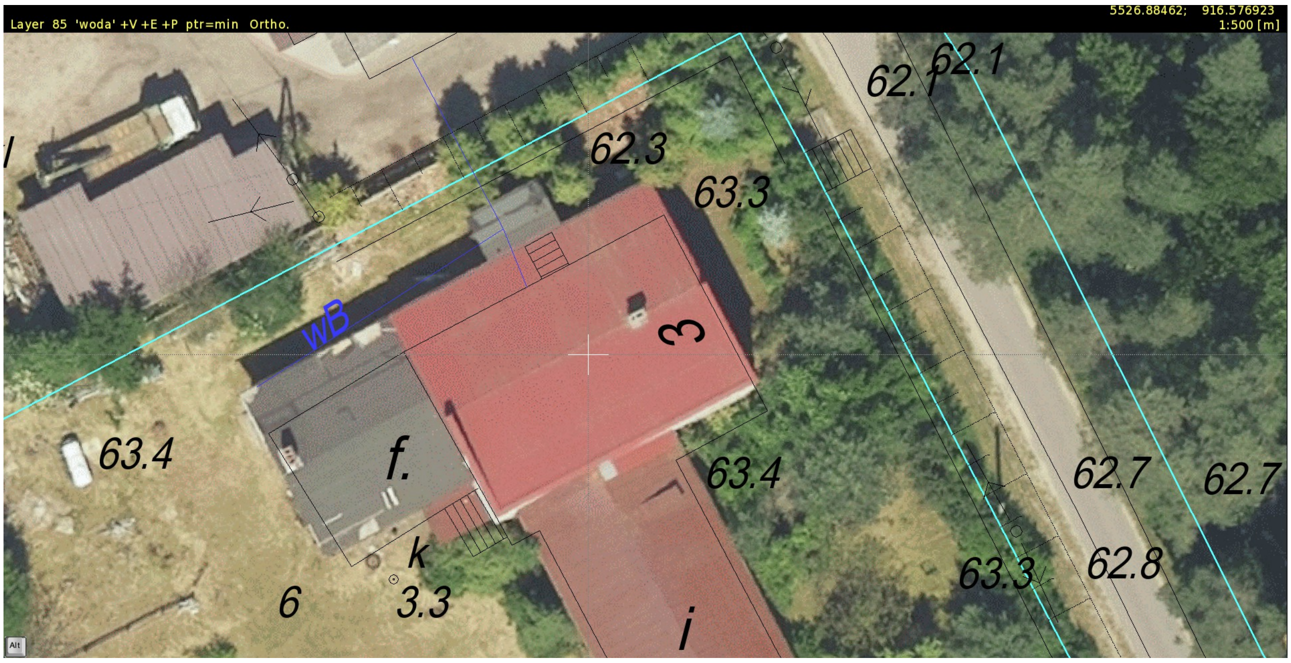

C:\Users\Marek\ALFACAD64\Drawings\projekt podziału nowa wersja_4_1.alf

```
Layer 85 'woda' +V +E +P ptr=min  Ortho.          5526.76923;   916.576923
                                                                1:500 [m]
```

Maps (so 1 or 8 bit colour images) are displayed and printed more precisely than photos, especially when large images are involved, displayed in large scale and printed on high resolution printers or saved in high resolution image files or PDF documents.

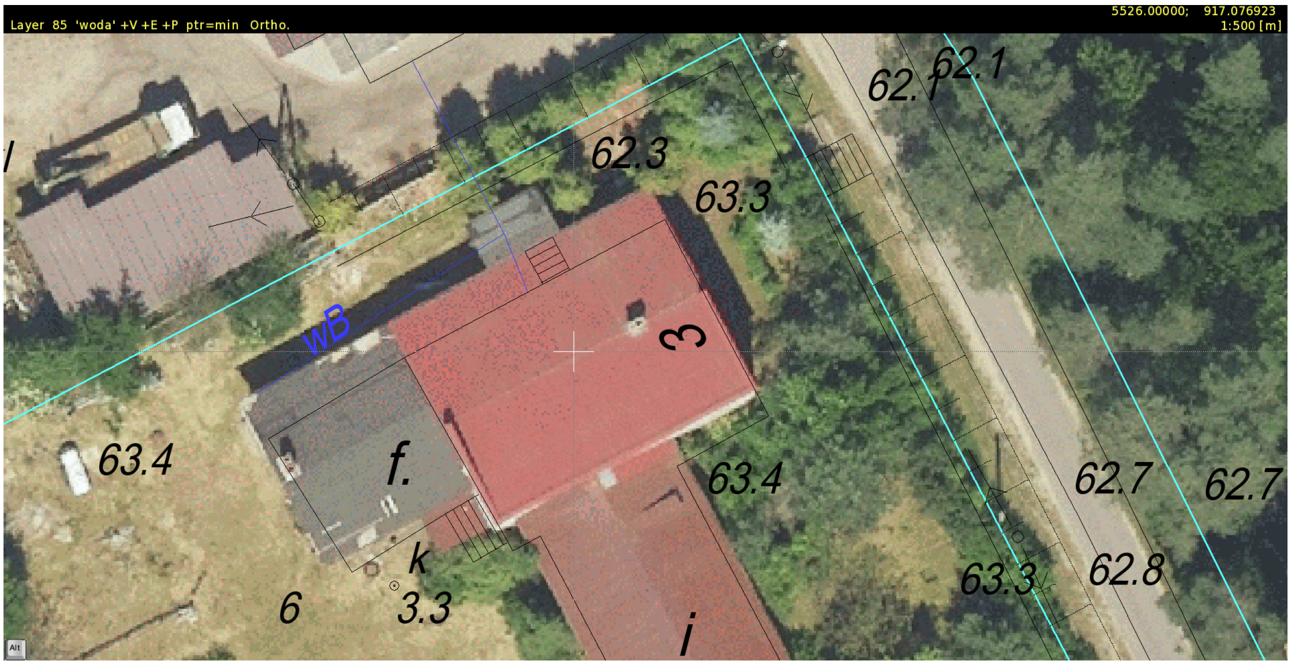
Is necessary to mention that images are stored in drawings in their original form, and they are scaled and rotated in real time, or such transformation is done ony once on so-called virtual screen, if images are buffered. Scales (scale X and scale Y) and angle can be always modified, or image can be re-calibrated, without losing any of original accuracy.

Here is an example of using „photos” instead of “maps”, however the same PCX images, converted to PNG were imported into the same locations with identical parameters. No matter it's “map” or “photo”, even in very large scale of displaying the image the difference between always very precise vectors location (coming from digital map) and image pixels location is virtually identical. Below is the image without buffering (also identical with printing method which is based on precise calculation of the position of each pixel). This is most accurate presentation.


Layer 85 'woda' +V +E +P ptr=min Ortho.

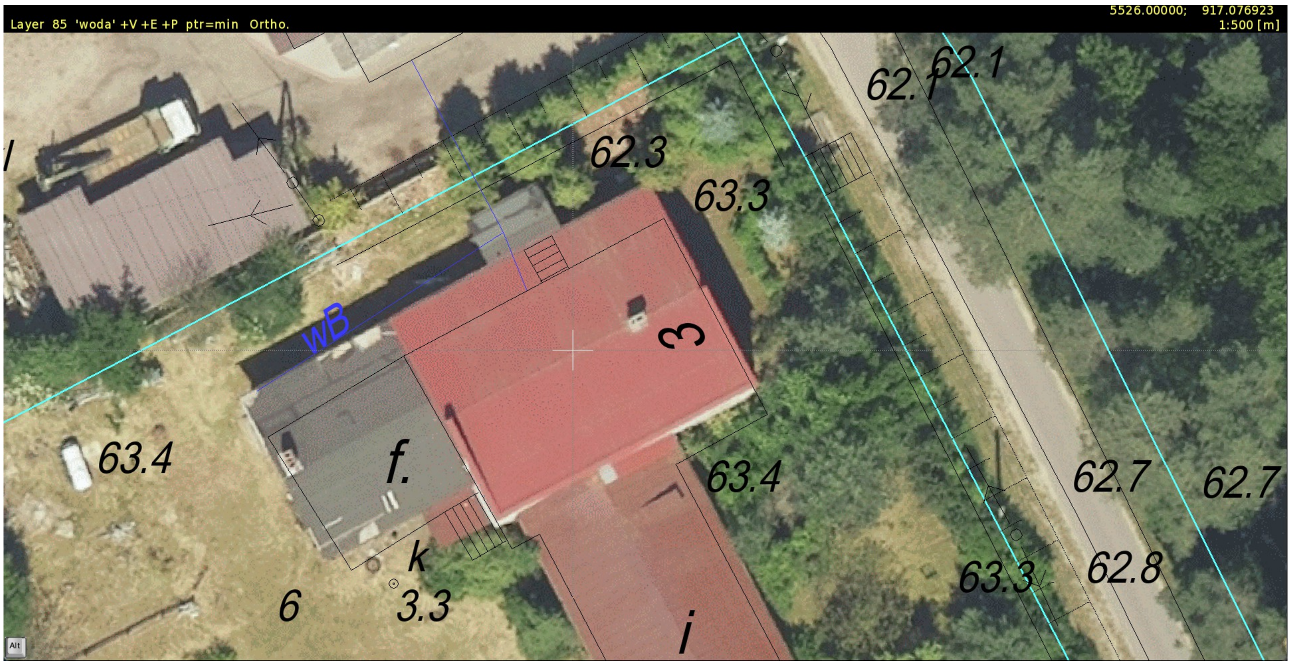
5526.88462; 916.576923
1:500 [m]

Here is the same image but imported as “photo” (there is no colour depth difference, because PNG were converted from PCX images extracted from drawing in **Block - image exporT** function) buffered (so slightly simplified).


Layer 85 'woda' +V +E +P ptr=min Ortho.

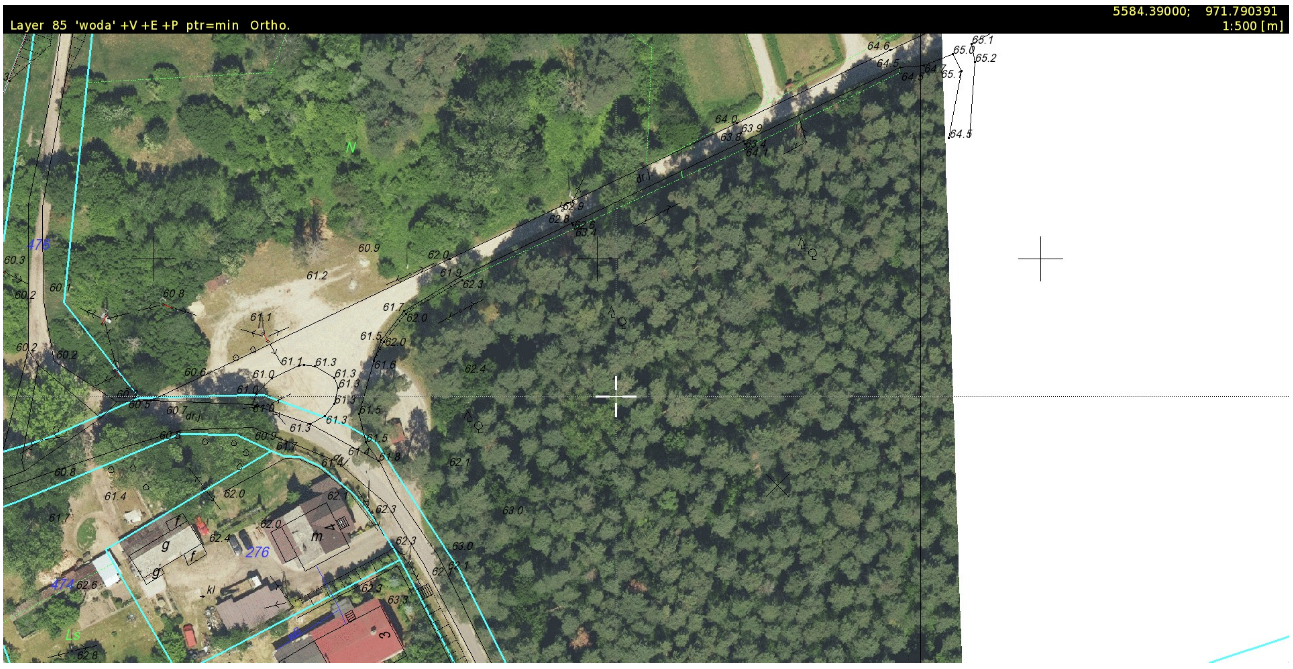
5526.00000; 917.076923
1:500 [m]

and not buffered (below).

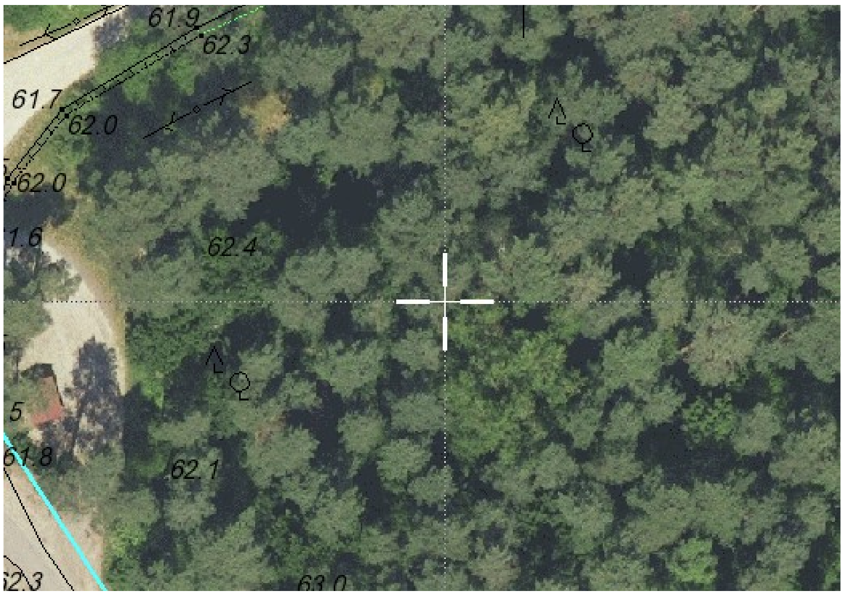


Smart pointer

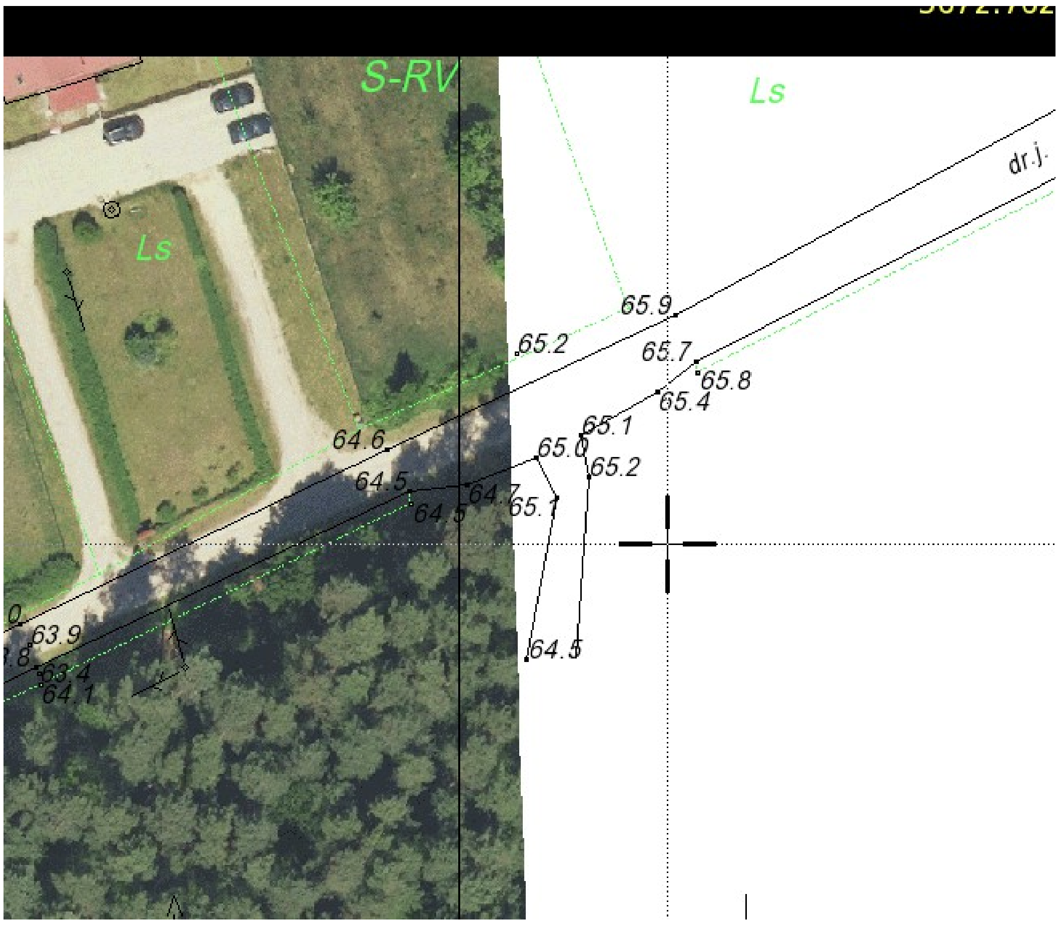
Cursor (pointer) was re-designed in ver. 1.4. Instead of simply reversible colour lines (drawn in so-called XOR mode) now AlfaCAD makes all effort to make the pointer as much visible as possible independently of the colour of the background. And background colour can vary, especially where maps or photos are placed, sometimes covering all drawing (like geodetic sleepers). Is highly recommended to use white colour of the pointer, however any other colour is also available.
When pointer is moving trough the drawing, with every piece of moving the background is analysed, and when visibility of the cursor is doubted, so the calculation is telling the intensity of white or black pointer is close to the white or black intensity of the background, pointer colour is inverted.
Below is the situation where pointer is moving over the relatively dark area. It becomes white.



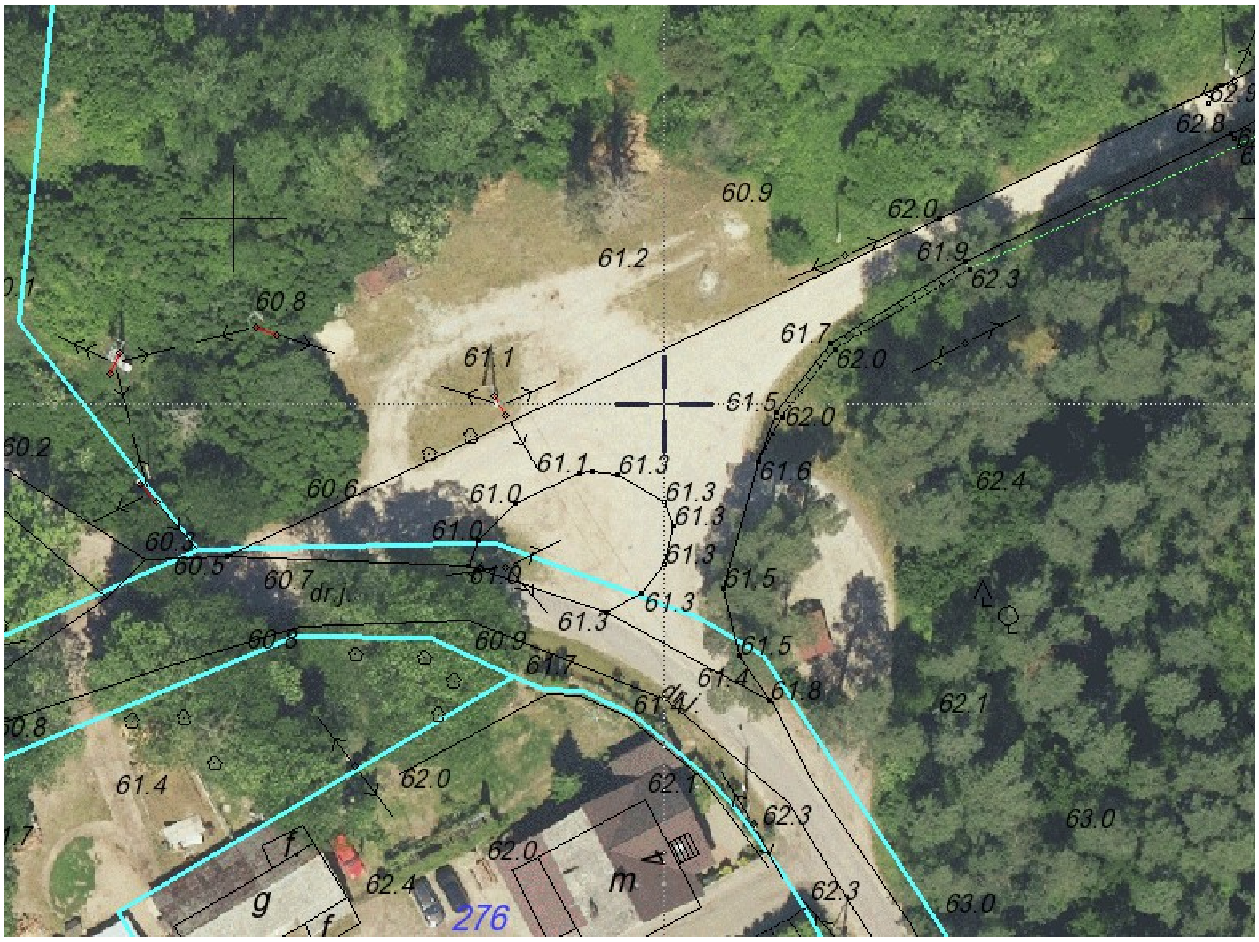
Below is closer view of the situation. Cursor is white, because it's visibility in black would be much lower. There is one fundamental rule: if cursor is declared as white in **Options** (what is highly recommended) it is or snowy white, or charcoal black. In just rare situation part of the pointer can became grey and it can only happen on the edge between light and dark area.



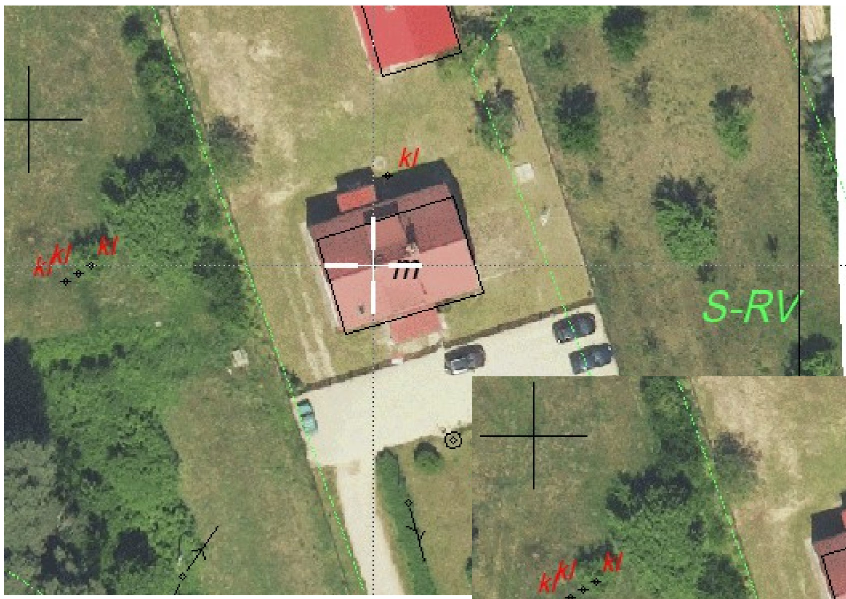
Changing position to white area makes pointer black:



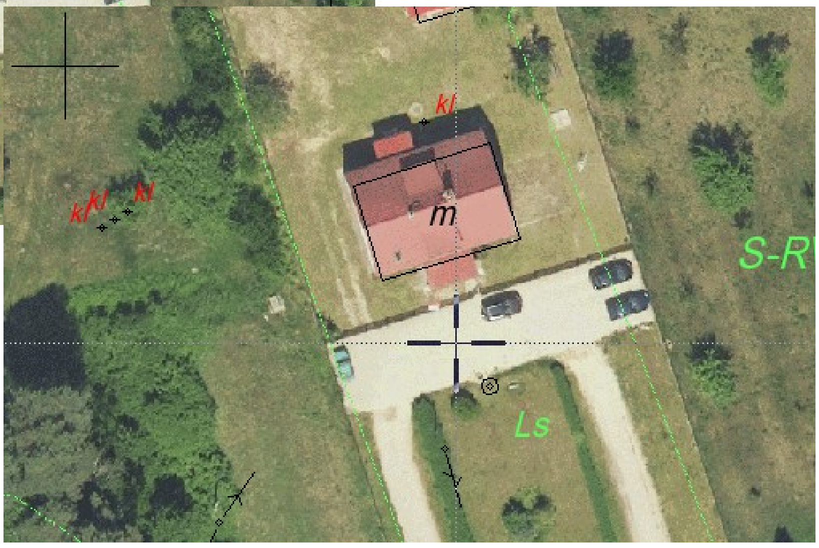
Not necessary background must be snowy white to make cursor black. Also in areas where background is light enough to rather show pointer black than white is better for better visibility:



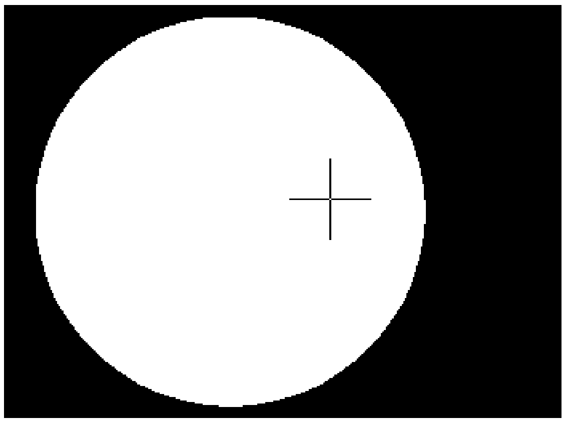
Pointer over red roof is better visible in white:

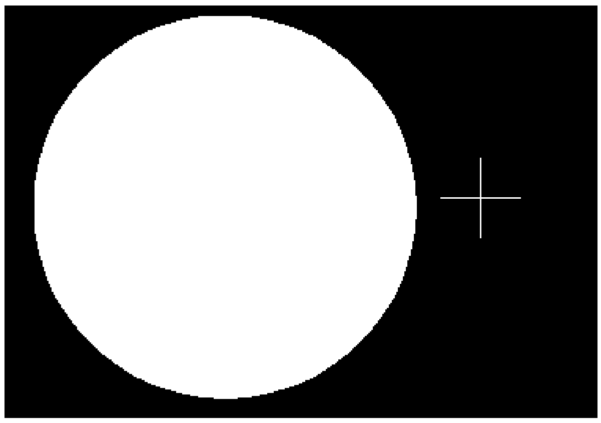


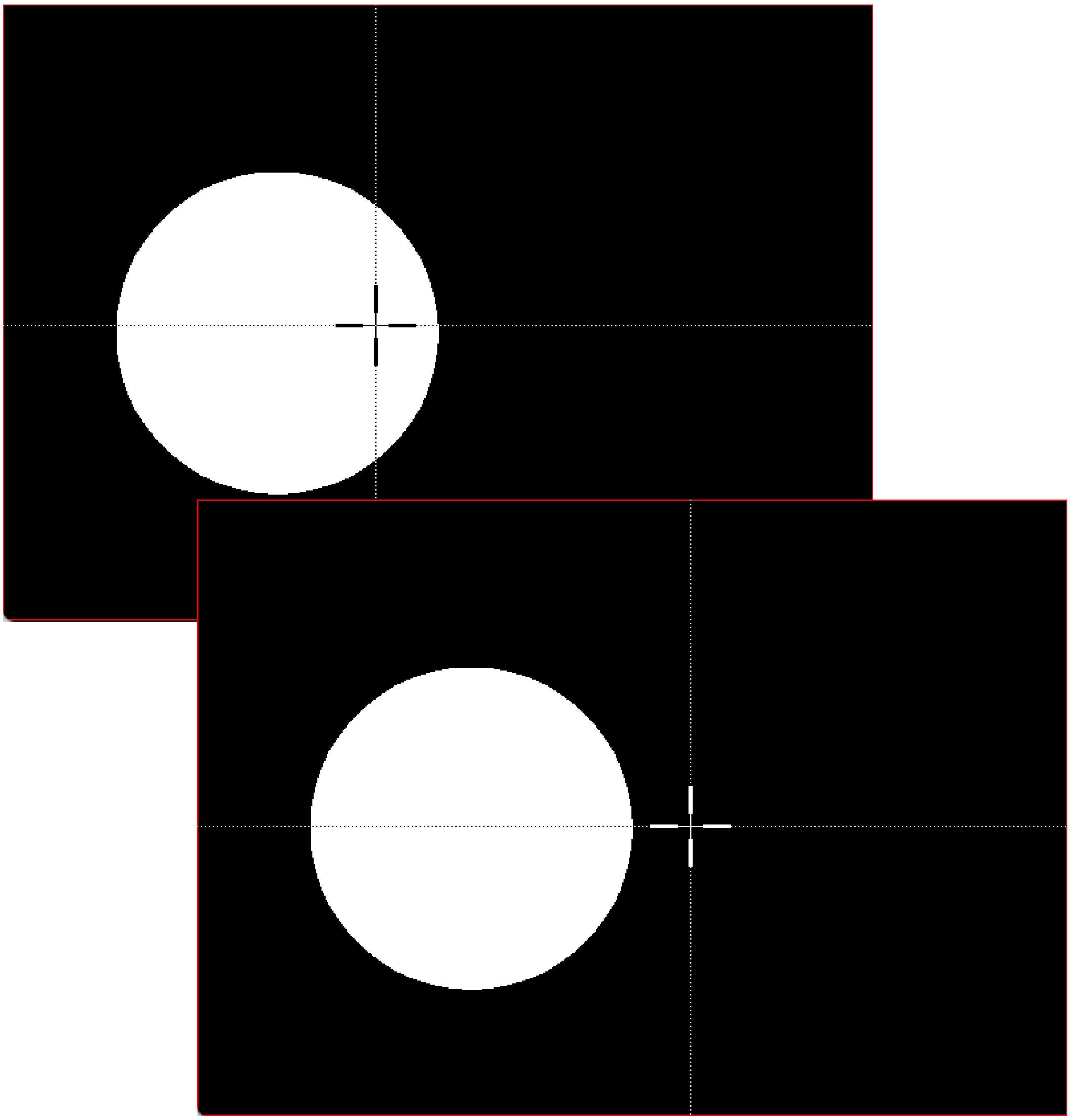
while over the light area is better visible in black:



Here are some obvious examples, common for both 1.4 version and its predecessors:







Enjoy AlfaCAD
author