Big thanks to the Inprise company for Borland C ++ Builder. I want also to thank all creators of the FastLIB library and especially of class FastBMP. Thanks to A. Milukov, the formulas of some effects were taken from his program AviEdit.

For bugreports I thank:
Alexander Milukov
Artem Mkrtychev
Ilya Artushin
ZDNET
Roman Gusev

Thanks to authors of effects:
Fingerprints - Roman Gusev
Twirl&fingerprints - Roman Gusev
Twirl 2 - Roman Gusev
Fun - Roman Khimov

And special thanks to Alexander Rabtchevich
for help in manual translation.

This program build lacks the correct and full filter ability.
That's why their set is rather small and they work in a strange way. But you can add your own filters, set the matrix and filter the image/region, chosing filters from the list.

The preview window shows the full image. By clicking on its arbitrary point you can see the image part in the editor window, starting from this point. All tools apply to the editor window. You can set the magnification of the editor window and resize it, as well as resize the preview window. If region is selected, all actions (filters, gamma adjusting and various tools) are applied to the selected region. You can undo any action with the "Undo" command from "Edit" popup.

You can choose the color by three different ways: by picking it from the image (see "Tools"), or by adjusting its channels, or by choosing from standard Windows list. To choose the background color, press the "BackColor" word at tools panel previously, otherwise press "ForeColor".

You can edit the image and select regions with different tools:

1-st group: Brush, Pencil and Eraser

You can paint with the help of theese tools. The brush differs from the pencil by flatting the painted line. The eraser paints with the background color (it differs from the brush, which paints with foreground color). You can choose different options for these tools in the "Options" window, namely their type, transparency, size (diameter for circles and side extent for squares) and the shape (circle or square).

2-nd group: Picking the color from the point.

It works in "Edit" window.

3-rd group: Filling

You can fill single colored object or being surrounded with the background color with the foreground color. The second is convenient when erasing the "trash".

4-th group: Color changing

There are two tools, one changes the chosen point color to background color, another changes color of all other points.

5-th group: Region selection

You can select either single colored object or region (rectangular or arbitrary)

Tools from 3,4,5-th groups are used often jointly with the color primitivization.

6-th group: Moving the selected region.

In this case the cutted region is replaced with the background color.

7-th group: Text adding

You can add text of foreground color. It can be with the smoothing or not. You can set the text transparency, using "Options", as well as for the tools from the first groop.

8-th group: Gradient

It's one among the most beautiful tools. It lets you create the smooth gradient from foreground to background color. You can also set the transparency for this tool.

One of the collest features of this program is the color primitivization rule. It means that with the use of scroll bars you can change the primitivization value for each channel. The primitivization rule consists in rounding the current channel value with tunable precision. Of course, other editors have a tolerance, but they require to input it numericaly and don't have the ability of preliminary preview. The primitivization affect the color change tools, filling and selection single coloured region tools. Despite its simpleness, this ability worth when editing images with large palette size. I advise you to examine this mode - it's very useful.

This program version includes a few interpreted distortion filters. Each deformation affect all image if nothing is selected, otherwise - selected region. Background uses background color :-).

Even if you are not a professional programmer and lack mathematics, try to create these effects. It's really a very engrossing amusement, and it doesn't require much time. You can even write down random equations (they must be of right syntax) - the most interesting effects are often created in such a way. Moreover, this is you chance to earn fame - the best of effects will be included into new program versions with credits.

My name is Victor Sazhin aka VicMan.

My e-mail:vicman@orc.ru

Page of my programs: www.vcw.ru(please put it in your Favorites).

I've been writing free software since August 1998.
My main language is C++ (of cource), sometimes Java, Pascal and Basic.

There are some projects I'm currently working on

VCW VicMan's DataBase
VCW VicMan's EmailEra
VCW VicMan's ReCoder
VCW CoDe StyleR for C++
VCW VicMan's Photo Editor
VCW Picture Searcher
Before that, I've been working on:
VCW Email Messages Eraser (last update was 1.81)
VCW C/L ReCode (last update was 1.1)

You can set constants to use in you formula as well as their names and initial values. All constants range from 0 to 100. It is recommended that the image stay unchanged when the constants have their initial values. For-instance, if you want the constant to be within a range from 7 to 11, enter the expression like (a/25+7). It is better to parenthesize all operations with constants - in this case the calculation will be made once, otherwise it will be calculated for each point. The constants will be presented in filter window as scroll bars. Now enter the equation either in polar or in orthogonal system. You can choose either the resulted coordinates are the reached target, or the origin. You can examine the filter on the image. You can create the distortion filter by the string. Soon these strings will be published at WWW.VCW.RU.

The ability of instant distortion effect creation is, alone with primitization, one of program flavours. Any man, having initial mathematics knowledge, can create his own effect in several minutes and become famous if he send it to WWW.VCW.RU Best distortion filters will be published in new program versions with credits. You can use existing functions as examples.

Constants

For orthogonal coordinates

w - image width
h - image height
cx = x - w/2 the distance from image center to current point along x axis cy = y - h/2 the distance from image center to current point along y axis

For polar coordinates system
r - the distance between the point and image center
t - angle

Operations and functions
+ addition
- subtraction
* multiplication
/ division
sin sine
cos cosine
sqrt square root
abs absolute
int integer part
random(x) random number within a range from 0 to x. It is calculated for each point even if it doesn't have cx, cy, x, y, r or t as argument