How to GoDot #12

By Arndt Dettke

What blurring is good for

Some issues ago I promised to talk about how to create headlines with “dissolved” or melted edges. Well, this time we will approach this topic, and use those modifiers which are handy for that task. But before affecting headlines, I’ll show you for what these modifiers were first and actually intended, for what they are good for.

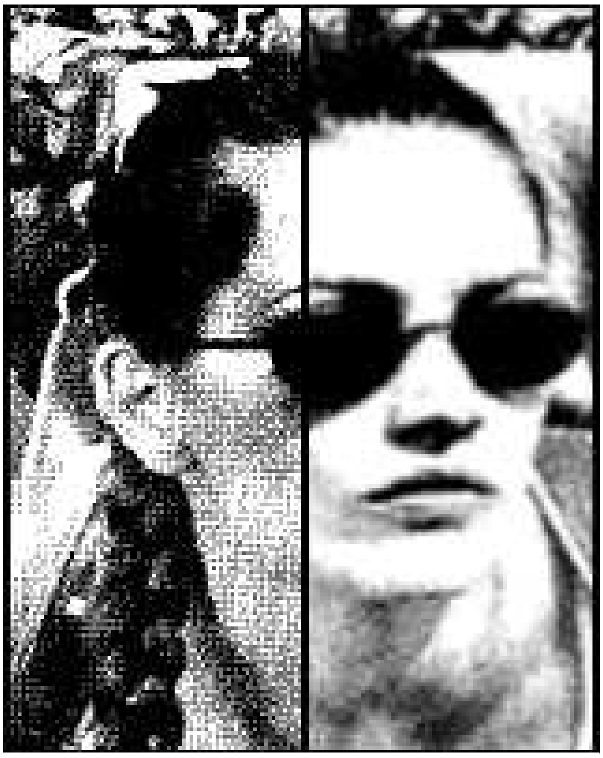
The modifiers in question are **mod.-Blur_Low**, **mod.Gaussian**, **mod.Blur_High**, **mod.CrossHatch**, and also **mod.ApplyDither**. Except for the latter, all of them serve for blurring an image. Now, what is that good for?

Yeah – just take a look at Pic #1. Do you see what I mean? Yes, (slight) blurring removes any kind of unwanted patterns! Prominently, such unwanted patterns arise when scanning (e.g. with a Handyscanner device) or digitizing video (GoDot is capable of several video digitizers). They are called “Moiré”, and you get rid of them by blurring. Blurring is when the contrast difference between two pixels gets lowered by adding pixels with graduated values in between. Where there was black and white is then black, gray, and white (slight blur) – or black, dark gray, mid gray, light gray, and white (strong blur). Or more generally: blurring is adding more intermediate grays between two existing grays. Ok.

But we have a problem here. The C64 doesn’t provide too many grays. Just five are available, and that’s all. What if I need gray values the machine doesn’t provide? For instance, between dark gray and middle gray? Locked up?

No. GoDot knows what to do. We created GoDot as a gray scale image processing software founded on 16 different grays – black and white, and 14 intermediate values. These are far more than those poor 5 hardware grays, but where do we get these additional grays from?

Hehe, there aren’t additional grays. We just declared colors for grays! :-) Not just randomly.



Pic # 1: The difference is remarkable! This is a C64 scan, the right half is processed with mod.Gaussian. Saved with svr.GIF.

The C64 colors all have distinct intensity values. We found out which, and ordered the color palette according to this fact – it’s the palette you see when you enter the Palette requester window under “Color Controls”. When processing an image with a blurring modifier, GoDot adds one or more of these palette colors to the image. Good.

The only drawback in all this is: you can’t see it on a native C64. What you see is colors, no grays. What to do now? Firstly, we created modules to output these gray scaled images to devices which are capable of many grays. Color printers for instance, but also PC computers and Laser printers. All GoDot’s printer drivers do gray scale printouts. For the PC (and the Amiga and Plus4) we have appropriate savers (svr.GIF, svr.BMP, svr.PCX, svr.IFF, svr.Plus4MC16). For Lasers and Wheels we have svr.Postscript.

People who emulate their C64 on a PC using VICE can easily watch a grayscaled 64 just by activating a grayscaled palette (which I offer for download at my site). Very impressive!

And for all the remaining people there’s mod.ApplyDither. ;-) You execute it, set the number of colors to 5, the dither mode to “Hires”, and apply these settings to the image. Impressive, too. It’s just this ApplyDither modifier which we will use for our processing of the headlines later on.

Before we reach there, let me tell you about the differences of the blurring modifiers listed above. I put them intendedly in the above order. The first three of them mark stronger effect from left to right.

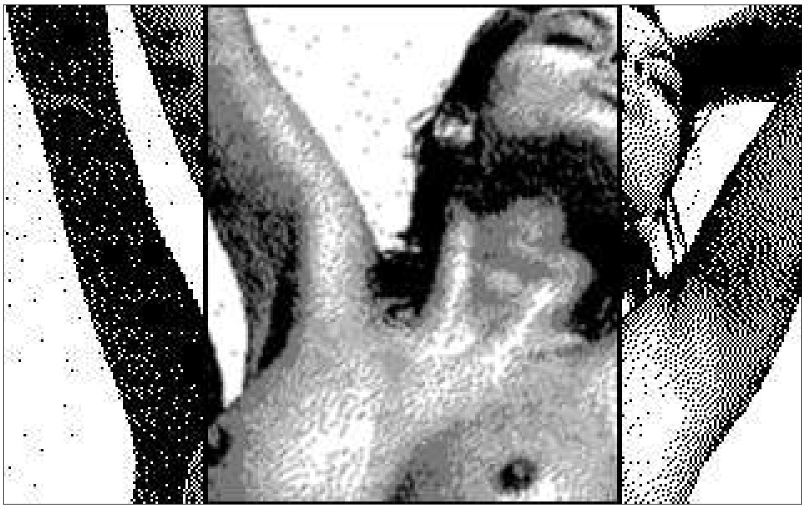
The least effect is mod.Blur_Low. Two neighboring areas of different gray get rimmed with a one pixel intermediate gray, which is close to the one of the area it is written in (Pic #3). Mod.Gaussian is quite similar, but with grays which halve the distance between two neighboring grays (Pic #4). And mod.Blur_High accentuates the vertical edges by adding more grays there (Pic #5). All of these modifiers tend to add more grays in the brighter areas of the image. By repeated use of them, the images get darker and grow more and more hazy.

Mod.CrossHatch is a special case of mod.Gaussian. Every even pass of it produces double pixelled results, good for multicolor screen output.

There’s one more useful feature of mod.-Gaussian which is very handy for processing headlines: If the two neighboring gray areas are also neighboring grays (with no more intermediate gray), then the brighter area gets thinner by one pixel! The opposite effect can be obtained with **mod.MosaikMaxi**. Applying it makes a brighter area thicker by one pixel. Very handy!

Remember that last Digest issue? We transformed monochrome images to multicolor by use of **mod.Smooth**. There was the disadvantage of reduced resolution after Smooth (160x100). Applying mod.Gaussian avoids that, see Pic #6.

Ok, next time we dissolve headline letters, which at the same time is the terminating article of this Writing-with-GoDot thread. See you then!



Pic # 6: A monochrome image processed with mod.Gaussian to convert it to grayscale. See either of them in this combination.

Mod.Blur Low – Modifier which brightens the edges of dark areas and darkens the edges of bright areas a little bit (one pixel depth). If two neighboring areas have neighboring colors, the brighter area gets slimmed by one pixel.

Mod.Gaussian – Like mod.Blur Low this modifier brightens the edges of dark areas and darkens the edges of bright ones. The difference is that the value for the new gray is just in between the two existing ones. The slimming effect also applies for this modifier.

Mod.Blur High – This modifier is similar to mod.Blur Low, but with much stronger effect.

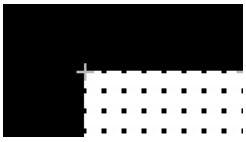
Mod.CrossHatch – A special effect modifier similar to mod.Gaussian, but producing double pixel output useful for multicolor display mode.

Mod.MosaikMaxi – This one brightens an image and blurs it at the same time. Repeated use produces a tessellated view, hence the name of the modifier.

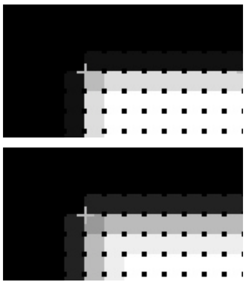
I have uploaded the GoDot scan to my site, as well as the grayscale palette for use with the VICE C64 emulator. Go to:

http://www.godot64.de/download/workshops/index.htm

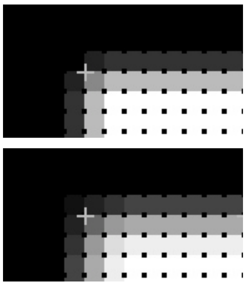
Have fun using GoDot!



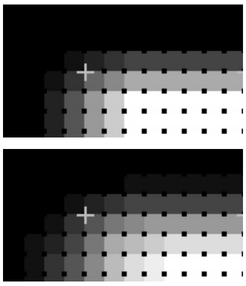
Pic #2: Two areas of black and white, unaffected.



Pic #3: **Blur Low** after first pass. Hardly to see: the dark gray rim along the black edge. Below is the result of the second pass.



Pic #4: **Gaussian** after first pass. The added grays provide smoother steps from black to white than Blur Low. Below again the result of the second pass of the modifier.



Pic #5: **Blur High** goes very smoothly from black to white. Even more in the second pass.