DA'S
VECTOR PRO

Contents

Introduction to the improved functionality of DA's VECTOR PRO

Introduction

For economic/ecological reasons we thought it sensible not to create a new manual for DA's VECTOR PRO, but to add an additional volume to the existing introductory manual for DA'S VECTOR. This was possible because the difference between the two programs is almost exclusively in the enhancements; only a few functions or interface elements needed to be changed. The beginner should therefore read the introduction to DA'S VECTOR first and then turn to this additional volume. The many DA'S VECTOR owners acquiring the PRO version as an upgrade will find in Part A a compact summary of all enhancements and changes.

Part B of this introduction serves a different purpose. DA'S VECTOR PRO is the core element in the first fully digital video production system for Atari computers. Since we as well as the users are breaking new ground with this, we thought it proper to give a more detailed explanation of the key concepts and of many detailed questions. A brief summary, however, will be given right now.

By fully digital video production we mean that the entire editing and creation of videos (in the widest sense) is done digitally inside the computer. This implies that the real film sequences are completely digitised and then edited in the computer, mixed with computer animations and then, finally, re-recorded on the target medium. This target medium can be a video tape, film or a digital medium such as the computer itself or its mass storage device. Of course this also includes sound. All editing is done at the highest quality, fully sufficient for studio productions. You have available all the graphics and image editing tools which have been developed in recent years for the DTP, graphics and image editing fields.

To avoid confusion: we are not talking about mini videos involving a loss of quality, intended for replaying on a PC (although these, too, can be created), but about digitally produced 'films' of maximum quality. The degree of quality achievable depends entirely on the peripherals used: for real film sequences, of course, the resolution of the source medium (film, various video systems) and of the digitising resolution (digitiser, slides scanner) are decisive. When recording the finished film sequences, too, only the quality of the equipment (colour resolution of computer or graphics card, video system and conversion, resolution of slides camera etc.) are relevant.

DA'S VECTOR PRO can, of course, scale all graphical elements (vector graphics, vector text) freely and therefore output them at almost unlimited resolutions. The mass storage device used will, of course, set certain limits. Without compression, one second of studio-quality video takes up about 30 Mb and many of the new options in the DA'S VECTOR PRO animation editor allow no further compression. Many semi-professional and amateur applications, however, require much less. Also, the processing speed of the available computer plays an important part. They are easily adequate for professional video resolutions - but the creation of cinema film formats requires faster systems.

We have described DA'S VECTOR PRO as the core element of such a digital video system because, in principle, you can do everything with this program and the peripherals available today, although often with considerable effort and time spent. This applies not too working with DA'S VECTOR PRO but to the control of the peripherals. Thus you can already today digitise picture sequences using a commercial colour digitiser and a recorder with good freeze frame and single frame stepping, but still need to record frame for frame 'manually'. The same applies to the output: even animations which can no longer be compressed can be copied to the target medium at high quality using single frame recording, but here too it is still necessary to operate manually and it also requires a rather expensive single frame recorder.

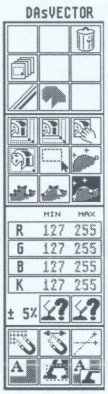
Here, too, we will come up with new products which can automate the work as well as enable high-quality recording on the more upmarket home recorders (S-VHS, Hi8). Our support line will answer all your questions concerning this topic, as well as new products and service bureau's which you may require to record your productions for you if you do not have the relevant equipment.

DIGITAL ARTS AG
CH-5503 Schafisheim, October 1993

A. Program enhancements and changes to DA'S VECTOR

1. Changes/Additions to the vectoriser

1.1. New option: Scanning monochrome images



The vectoriser can now also trace monochrome pictures in the TIM format. Because of the virtual tracing (unlimited image size!), it is not possible to vectorise pictures in the IMG format so that these need to be converted first into the TIM format. Users of GT-LOOK II can, of course, save the images in TIM format immediately. This also means that monochrome images can now be used as fill patterns.

1.2. Changes to the snapshot function for picture objects

The 'snapshot function' with which you can generate a picture object from an imported picture (rectangular frame in the picture proportions with the picture as a fill pattern) now returns to the vector graphics menu immediately after you have clicked on it. There you will find the picture object in the work area. You can, of course, generate picture objects from monochrome pictures, too.

2. Additions/changes to the vector graphics editor

2.1. Additions to the Graphics attributes sub menu

Most of the additional functions can be found in the "Graphics attributes" sub menu. Linear and radial (vector) colour graduations can now be used to fill vector areas, as can the so-called 'blendings' (which are metamorphoses between two parts of a vector path with colour blending). A lot has been done to the pictures used as fill patterns, too: in addition to the halftone, palette and true colour pictures you can now also use monochrome pictures in the TIM format as fill patterns; you can assign any drawing colour to them. For all picture types, the colour White can now be defined as transparent. A very important change may not at first be noticed but will come into effect automatically: when the object is rotated, the fill pattern picture will be rotated along with it at any angle. In other words, all pictures can now be freely ro-

tated. This angle can also be entered from the keyboard. Because of these new options, the user interface had to be slightly changed here (see below). In general, the user interface has become a little more complex due to the many new options. Depending on which fill icon you click on, the sub menu may have a different appearance in order to make available special functions for the corresponding basic function (colour graduations, blendings) which could not be fitted in elsewhere. All these functions will be described in detail below. You will also find a new parameter among the drop-down menu options (xx colour levels), which is connected with the new fill functions. Here you can set the number of colour graduation levels ON THE SCREEN; this only affects the display quality and speed on the monitor. Output (Print etc.) always takes the highest possible resolution.

2.1.1. Vector colour graduations

Beneath the two icons for the different fill algorithms you will find two new symbols for linear and radial colour graduations. You could, of course, create any kind of vector colour graduations before, using the multicopy function or the time-space converter (and these extremely versatile options are still present) , but these functions always create a large number of physical copies, requiring plenty of memory. This is no longer necessary now for 'simple' linear and radial graduations: such a graduated fill for objects (paths) requires virtually no additional memory. When you select one of these two icons, the sub menu changes appearance: the settings for the fill patterns disappear and instead there are special functions to set the colour graduations, to be described in more detail below. You can switch back to the normal sub menus by de-selecting the graduation icon. Set start and end colour: two rather than one colour fields appear. The first shows the start colour, the second the end colour of the graduation. Both can be clicked on as before, whereupon the colour setting dialogue appears and the colours can be redefined. Even simple graduation: when you select this icon, the graduation (linear or radial) will be drawn evenly from the start colour to the end colour. Soft simple graduation: when you select this icon, the graduation (linear or radial) will be drawn with a certain amount of 'bending' between the start and the end colour; the transition appears 'softer'; instead of long complicated explanations you should simply try this out. Even cyclical graduation: when you select this icon, the graduation (linear or radial) will be drawn evenly and cyclically from the start to the end colour and back again. Soft cyclical graduation: when you select this icon, the graduation (linear or radial) will be drawn with a certain amount of 'bending' and cyclically from the start to the end colour and back

again; the transitions are then 'softer'. Set graduation direction: This is an interactive tool that allows you to select the start and end points of the graduation. First select this icon. Then move the mouse to the desired starting point of the graduation (which can be outside the object!) and drag a 'track' with the mouse button pressed to the position of the end colour; the graduation 'track' will be indicated by a line. Just release the mouse button at the end position. The icon will be automatically deselected. If instead you wish to continue experimenting, hold down the SHIFT key when drawing the graduation direction: you can now repeat the operation as often as you like; you will only exit this mode when you release the SHIFT key. IMPORTANT HINT: if you are using colour graduations as fills, the line attribute will be automatically set to 'transparent'. It is therefore NOT possible to work simultaneously on 'outlines' and 'graduation fills'. If you need to do this, you must first create a copy of the object and place it on top of the first object (and preferably group them); set the area for this copy to transparent and select a line thickness. Hint for work in the animation editor: when you save two variants of an objects in the film clipboard, the following settings are interpolated in a linear fashion: start colour of both variants, end colour of both variants, start point of both variants, end point of both variants. Thus, within an animation, you can move as well as re-colour the colour graduation. The other settings (gradient type, gradient mode) cannot, of course, be interpolated. They are switched abruptly in the animation.

2.1.2. Blendings

Next to the two fill algorithm icons you now find a new icon for the so-called 'blendings'; the icon at this position for picture fills has been moved one down. When you select the blending icon, the sub menu changes: the fill pattern settings disappear and instead a special function for the setting of the blendings appears, which will now be described. You can switch back to the normal sub menu by deselecting the blending icon. 'Blendings' are calculated transitions between two (or more) sub-paths, with the colour changing with each transitional step. Blends are therefore only possible if a path object contains at least two sub-paths! Many different variations are possible, depending on which path segment you select as an end segment and which you select as a start segment, and also the rotation direction of the individual path segments. You can also define the start end points, that is, the points on the start and end segments which correlate which each other. Here, too, we recommend that you simply experiment with the different possibilities using simple objects, in order to gain some feeling for the almost unlimited possibilities.

Set start and end colour: Instead of one colour field you now get two. The first shows the start colour, the second the end colour of the blending. Both can be clicked upon with the mouse as before, whereupon the selection dialogue for the colours appears and the colour can be redefined. Number of intermediate steps (blends): Here you can enter the number of intermediate steps from the keyboard. The minimum value is 9, the maximum 255. How many of these intermediate steps are displayed ON THE SCREEN will depend on the setting for "Colour steps" in the Options drop-down menu. At the output stage, however, the number of blending steps selected in the Area attributes" sub menu is used. Select end segment/foreground: When you select this icon, you can select the end path segment for the blending using the mouse, by clicking on any point in the end path. The end path is assigned the right colour (at top of menu). Thus you are also defining the drawing sequence since the end segment will always be drawn last and is therefore in the foreground. This explains the icon (Foreground). Select start segment/background: when you select this icon, you can select the start path segment for the blending using the mouse, by clicking on any point of the start path. The start path is assigned the left colour (at top of menu). Thus you are also defining the drawing sequence since the start segment is always drawn first and is in the background. This also explains the icon (background). Change rotation direction: You already know this function from the vector path editor. When you select this icon, you can change the rotation direction of a path segment by clicking on the control point. Select start end point of blending: This tool allows you to define the transformation between path segments to a considerable extent. It is used to select corresponding curve points in the start and end segments. To do this, first select the icon. Then click on a control point in the start segment and then move the mouse with the button pressed to the desired control point on the end segment (or the other way round, makes no difference). Release the mouse button over this point. The icon will immediately be deselected again. The blending is redrawn, usually creating an entirely new type of transformation. Practice exercise: Enter the path editor and fetch a triangle and a square from the library. Place both sub-paths at a certain distance from each other. Now return to the vector path editor and from there enter the @Area attributes' sub menu. Select the blending icon. Now you can try out all the variants: change rotation direction of individual paths; select different start and end points on the triangle and the square; swap start and end path. For a second exercise, we recommend an object consisting of three path segments (e.g. a circle as a third element), giving more versatile combinations. IMPORTANT HINT: When working with blends, the line attribute is always set to 'transparent'. It is therefore NOT possible to work with 'outlines' AND blendings simultaneously. If

you need to do this, you must create a copy of the object and place it on top of the first object (and preferably group them); then set the area of this copy to transparent and select a line weight. Hint for work in the animation editor: When you save two variants of an object with blending to the film clipboard, the following settings are interpolated in a linear fashion: start colour of both variants, end colour of both variants, start point of both variants, end point of both variants. You can therefore use the blending in an animation for both a regular transformation of the shape as well as for re-colouring. The other settings (start segment, end segment, rotation direction) cannot, of course, be interpolated. They are switched abruptly in the animation.

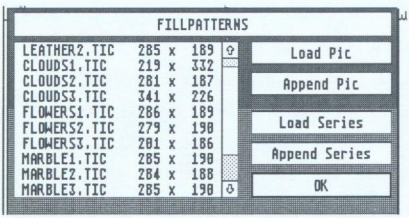
2.1.3. Changed and new fill pattern options

General changes/enhancements: First of all, you will now find the fill patterns icon in a new position. It has been moved one position down to make space for the blendings icon. You can now also use monochrome pictures in the TIM format (other formats can be converted to this using the picture converter) as fill patterns. The drawing colour can be freely redefined. The background is always white, but can optionally be set to transparent (as with all other image types now). This transparent setting is therefore the second enhancement. The third innovation is that images/pictures of all types can now be freely rotated by any degree. The many new possibilities necessitated slight changes in the user interface: you will no longer find the button marked 'rel' (relative). This setting is always selected when 'Abs' is disabled. The third button labelled 'TI' (Transparent Image) is used to switch transparency off or on. All the possible settings will now be described in detail. Absolute/relative fill pattern position: when this icon is selected, the fill pattern is firmly fixed in the work area. The co-ordinates of the top left corner are shown under 'X:' and 'Y:' and can, of course, be changed from the keyboard. When the icon is not selected, the fill pattern is positioned relative to the top left corner of the bounding box of the graphics object (as with the 'rel' button in DA'S VECTOR), and is therefore moved whenever the object is moved. You can add an additional offset under 'X:' and 'Y:' from the keyboard. Fixed/variable fill pattern size: when this icon is selected, the size of the fill pattern will remain constant when the relevant object is enlarged, reduced or otherwise transformed. The fill pattern is always rotated with the object, and the relationship with the object path maintained when 'fixed' is selected. This fixed size is indicated under 'W:' and 'H:' and can be changed there using the keyboard. When 'fixed' is disabled, the size of the fill pattern is adapted to that of the bounding box when the object is scaled;

during rotations this can lead to a change of the position of the picture in the path, since the rotation points of the picture and the path will only exceptionally be identical. Fill pattern transparent/opaque: when this icon is selected, all white areas of the picture are set to transparent, i.e. not drawn. When the icon is not selected, white areas are drawn opaque. Fill pattern position/offset: Here you can enter the fill pattern position (top left corner) from the keyboard. With 'Abs' selected, this is the absolute position in the work area. With 'Abs' not selected, this is the offset relative to the top left corner of the object's bounding box. Fill pattern size: Here you enter the size of the fill pattern from the keyboard if the size is defined as 'fixed'. If 'fixed' is not selected, you cannot enter anything here. The fill pattern size will then depend directly and entirely on the object size (bounding box!). Fill pattern angle: Here you can enter the rotation angle for the fill pattern. The fill pattern is rotated along with any rotations of the object. Should you not wish this to happen, you can always return the rotation angle to 0 (or any other desired value) here after the rotation.

2.1.4. Fill pattern editing (select/replace/load)

When you click on the fill pattern field, you will now no longer be presented with a file selector box directly but with the 'Fill pattern' dialogue. This gives you an overview of all fill patterns loaded, so you can select a fill pattern directly without a lot of clicking, replace patterns and load new ones. These functions will now be described in some detail. Select picture: when you have many fill patterns in your system a lot of clicking used to be necessary to find a particular one. Now all the fills are shown tidily in one small window with their names and sizes (important for picture optimisation and animation purposes). Upon selecting a picture field, the relevant fill pattern will be displayed in the display window so that you can now select any fill pattern directly. Load/replace picture: When you have selected a picture in the dialogue and click on 'Load picture', the usual file selector box appears and you can select a new picture as a fill pattern. This picture now replaces the previously selected one. To simply load a new picture without replacing an existing one, select the last (empty) entry or use the 'Append picture' function (see below). Append picture: When you click on this field the file selector box appears and you can select a picture to be appended to the list of fill pattern pictures. No existing pictures are replaced. Load series: When you click on this field, the file selector appears and you can select a picture file to be loaded as a fill pattern. If the filename ends with a number,



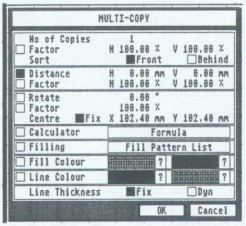
the program will automatically look for further pictures with subsequent numbers in the same path (e.g. Pict001.TIC, PICT002.TIC, PICT003.TOC etc.). All these pictures are loaded as fill patterns, beginning with the currently selected position in the selection window. The currently selected picture is therefore replaced with the new one as are any following ones. To avoid this, select the function 'Append series' instead. When you click on this field, a file selector appears and you can select a picture file to be loaded as a fill pattern. If the filename ends with a number, the programs looks automatically in the same path for further pictures ending with a number (e.g. PICT001.TIC, PICT002.TIC, PICT003.TIC etc.). All these pictures are loaded as fill patterns, in this case appended to the list of fill patterns. HINT on the use of the 'series' function with animations: The 'Load series' and 'Append series' functions were implemented especially for use in the animation editor. The offer a simple method of loading as a series of fill patterns entire animations or digitised film sequences existing as sequences of individual frames. Using the new multi-copier functions (fill pattern change), you can load complete film sequences with a few mouse clicks and include these in your own animations. You can find more on this in Part B of this manual.

2.2. Additions/changes in the 'Bezier net transformation' sub menu

Following many user requests we have now made the Bezier net library loadable so that you can now change between different libraries. This also necessitated (for reasons of space) some minor changes in the user interface. Above all, the 'Adapt Bezier net to object' icon has now slipped one row down where previously there was the 'Abort transformation' icon. This function no longer has its own icon; you now need to click on the trashcan icon in the bottom row of icons ('Delete library entry' function) with the CONTROL key pressed. Load new Bezier net library: When you click on the index file icon at the top, a file selector appears and you can select another Bezier net library to be loaded. These libraries have the filename extender 'DNT'.

2.3. Additions to the multi-copier function

Three very powerful options have been added to the multi-copier function: Rotation centre fixed/variable: When this option is selected, the rotation centre will not be changes during the transformation, otherwise it will also be affected by it, for example when moving etc. (depending on the setting of the other options), as in DA'S VECTOR. Fill pattern change: You can now change the fill pattern with every copy made. To do this, select the small square in front of the word 'fill pattern'. Now click on 'Fill pattern list'. The 'fill pattern' dialogue appears (which can also be called from the 'Area attributes' sub menu). Now select the



fill pattern for the first copy. All subsequent copies will now be assigned the following fill patterns in the list. If the number of fill patterns is insufficient, the last pattern in the list is used repeatedly. Calculator: The object can now be transformed using the graphics calculator with each copy. To do this, select the small square before the word 'Calculator'. You can now call up the calculator using the Formula' field and select a formula for the transformation. Each copy of the object will now be transformed (in addition to any other active options) using the selected formula.

3. Changes/additions to the animation editor

The animation editor has certainly received the most comprehensive enhancements. These consist not only of additional functions but very complex conceptual innovations. These new concepts (Camera and Filter) will now be described in detail. Some further details about the uses of the new functions can be found in Part B of this manual.

3.1. The camera concept and the camera functions

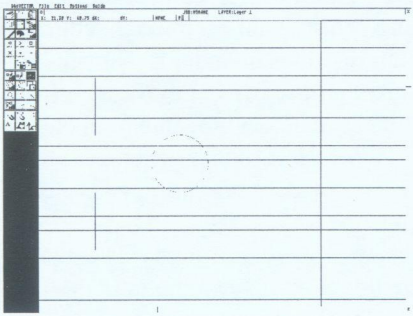
3.1.1. The camera concept

To begin with, we would like to introduce the camera concept using a simple example that will immediately show the advantages of the new approach. Let us assume that you want to make a simple figure run through the picture from left to right. A few motion frames are not enough because these have to re-appear at different positions in the picture. At best, this requires a number of copy functions, and you always have to move the object to the right position. We are now going to show you how much simpler this is with DA'S VECTOR PRO and the camera function implemented in it. First, create a few (about 4 to 8) motion phases 'on the spot', without any changes in position. Place these at appropriate distances as snapshots on the film clipboard. You can now use the Loop function (see below) to ensure that this motion sequence 'on the spot' is repeated as often as you like, without having to copy the individual motion phases by hand. Hence we now have a repeated 'on the spot' motion sequence. But how do we make the figure run through the picture from left to right? This is where the newly-implemented camera comes in: if required, you will now no longer work with a fixed page format at a fixed position, but you can now move the recording window itself just like a real camera can be moved. For example, you can position the

camera in the first frame so that the figure appears at the left edge of the picture. Now you move the camera so that the figure is just visible at the right edge and position this camera position in the last picture (loops included!). Now the figure will run through the picture! Please note that the camera movement (as in real life) is always in the opposite direction of the object movement; in our example, we have a camera movement from right to left so that the object moves from left to right. DA'S VECTOR PRO, however, does not just have one camera. There is a separate camera for each track (layer), and each can be positioned and moved independently. Let us clarify this with another example. The running figure is to remain at the centre of the picture now; instead the impression of movement is to be created by a 'passing' background which has to move in the opposite direction of the desired direction of the figure. This background could be a scanned colour photo placed on a lower track. Now you can simply move the camera for this lower track from left to right and achieve the movement past a background in this way. But the possibilities opened up by the DA'S VECTOR PRO camera go much further. Like a real camera, it also has a zoom function. Hence you can make the figure in our example shrink in size progressively by enlarging the camera field of vision; with an appropriate angle defined for the camera movement it will then run diagonally towards the background and become progressively smaller. Distortions, too, are possible, although these are not possible with a 'real' camera, or camera movements along a Bezier curve trajectory. And, to repeat this, there is a camera for every track, and these can be separately and independently 'animated'. Despite this wealth of possibilities, using the camera functions is very simple. Movement, zoom and distortion are defined with a single tool (cf. Chapter 3.1.2). A further tool exists for programming the camera movements (cf. Chapter 3.1.3). And the individual camera positions etc. are placed in the track using the same simple snapshot principle with which objects are placed in the track (cf. Chapter 3.3.1). Using the expanded animation settings you can in addition switch the camera and the camera movement off and on for each track separately (cf. Chapter 3.3.2). Each camera can also be assigned filters. This enormously versatile concept of filters is described in Chapter 3.2.

3.1.2. Movement, zoom and distortion using the camera

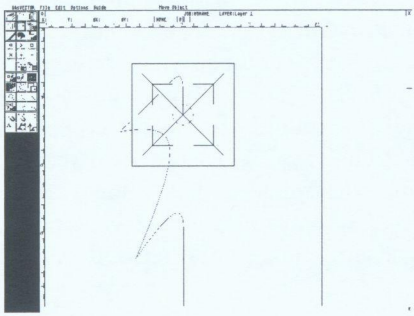
The camera appears as a kind of viewfinder as soon as you call up the animation editor. The basic camera setting will initially always correspond to the selected page format. When you select the new tool for camera operations, the camera tool is marked by handles at its corners. You can now move the camera by clicking in the camera field and moving the camera, with the left mouse button pressed, to the desired position. The size of the camera field can be changed by clicking on one of the corner points and, with the mouse button pressed, enlarging the



camera frame (Zoom to wide angle) or reducing it (Zoom to tele setting). If you also press the SHIFT key, the camera field can be enlarged or reduced disproportionately. This will create distortions: if you narrow the camera field the picture is distorted in width etc. The selected camera positions and sizes are then placed in the appropriate pictures of the film track using the 'snapshot' (Chapter 3.3.1 describes how you do this). As with objects placed on the track, the program will interpolate between the individual camera positions and sizes so that two or three snapshots will often be sufficient. To ensure that the camera is actually used during the recording or simulation of the film, the camera must be activated for each track in the 'Animation settings' dialogue (cf. Chapter 3.3.2).

3.1.3. Editing and using camera paths (camera movement)

A second tool for programming camera movement is the so-called camera path. This is a path consisting of four Bezier curves which can be edited with a special tool. Using the camera path you can generate smooth camera movements on curves of any curvature. It would be extremely tedious to create such a camera

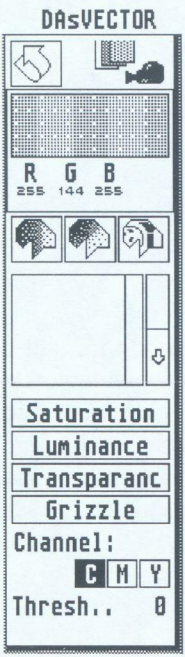


movement by hand if the movement is to be really smooth. There are two new icons in the animation editor for the camera path. One is a switch activating the camera path. The camera will now only move along the path, with the centre of the camera always on the path. You can therefore continue to manipulate the camera with the tools described in the previous chapter (move, zoom, distort) but with the restriction that the camera will not leave the path. There is a second tool for editing the camera path, which is used to bend and shape the path. This tool corresponds to the vector path shaping tool familiar from the path editor. If the appropriate button is active, this tool will also apply the 'Smooth Bezier curve' option. The movement along the camera path need not be linear from beginning to end. You can position the camera on any point on the path and save a snapshot of that position; hence the camera can run up and down the path arbitrarily. To ensure that the camera path is actually used during the recording or simulation of the film, the path must be

activated for each track in the 'Animation settings' dialogue (cf. Chapter 3.3.2).

3.2. The Filter concept and the filter functions

Just as you can place filters over a film or video camera lens, the cameras in DA'S VECTOR PRO can be fitted with filters - except that the filter options are much more powerful here as they could ever be with real cameras. Technically speaking, we are now leaving the field of vector graphics and entering that of image editing. The results of the filter settings can therefore not be viewed on the desktop but only in the film simulator. The new possibilities opened up by the use of filter techniques are so enormous that it takes much experimentation to work out what they are capable of. The following descriptions of the individual filter functions will therefore always include some simple examples and applications so that you get at least some impression of what they have to offer. In general, all the filters define how a track is drawn or combined with a lower (already drawn) track. Two types of filter are used to re-colour a track (saturation, luminance). A further filter offers transparent superimposition of tracks. The fourth filter (threshold value) allows pixel-based fade effects. The filters themselves are not, of course, static but can change smoothly or abruptly, making possible effects such as colour fade-outs and fade-ins, soft superimposition's and fade-overs and many others. A special new sub menu exists for the filter settings and this will be described in detail in the next chapter. This sub menu is strongly reminiscent of the 'Area attributes' sub menu as the settings are largely identical. Colour tone areas, vector colour graduations, monochrome pictures (fill patterns) with colour tints, halftone pictures with colour tints, palette and true-colour pictures can all be used as filters. The filter settings, too, are saved to the track using the snapshot (together with the camera settings). Here, too, the program will interpolate between the individual colour values, so that often two snapshots will be sufficient to program a fade effect. Since the filter can be used not only for colours but for all fill pattern types (pictures), not only colour but also spatial fade effects (or combinations of both) can be achieved.



3.2.1. The new 'Filter settings' sub menu

This new sub menu is very similar to the area attributes settings. This is not surprising as a filter can be visualised as an area fill of the camera area (the fill type will from now on be referred to as the filter type). The effects of the filter

depend on the filter function. In the following chapter, we will describe first the filter types and then the filter functions in detail.

3.2.2. Filter types

By filter type we mean the type of area fill for the filter. Except for the 'blending function', all fills used to fill vector areas can be used here too.

3.2.2.1. Monotone filters

The simplest filter type is a monotone area, that is a uniform colour value for the entire camera area. You can, of course, select a different colour value for each camera snapshot and the programs will interpolate between these colour values during recording or simulation. Monotone areas are therefore particularly well suited for re-colouring, fade-ins, fade-outs and fade-overs for the entire picture area. The settings are completely analogous to the settings for monotone fills of a vector area. The monotone fill is always active when no other options have been selected.

3.2.2.2. The graduation filter

The new vector colour graduations have already been introduced in Chapter 2.1.1. You can also use these graduations as filters. The program will interpolate the start and end colours of the graduation as well as the start and end point, so that the graduation filters can vary in colour as well as space. You can combine the graduation filters with time-space fades, shadow and lighting effects and much more. Changing the graduation type is also possible, but these changes will happen abruptly.

3.2.2.3. Monochrome filter (masks)

As already described, DA'S VECTOR PRO can now also use monochrome pictures as fill patterns, with the drawing colour determined by the selected tone value. These tinted monochrome fill patterns can also be used as filters, with the colour again being variable and fade effects being available. Monochrome fill patterns are ideally suited as masks for cutting out image sections (digital genlock), and the masks can be combined with fade effects. The settings are again completely analogous to those of the vector area fill patterns. You select the fill pattern icon and the required monochrome picture (TIM format) as the fill pattern.

3.2.2.4. Halftone filters

You are already familiar with the use of halftone pictures (greyscale pictures) as fill patterns which, using the tonal value setting, can also be tinted. Such tinted halftones can also be used as filters. Using continuous colour changes, halftone

filters are very nice for time-space fades, complex lighting and shadow effects, soft masks and much more. Their use is once again entirely analogous to that of the fill patterns for vector areas. You therefore select the fill pattern icon and the required halftone picture (TIH format) as a fill pattern.

3.2.2.5. Colour picture filters

As you may already have guessed, colour pictures (palette pictures in the TIP and true-colour pictures in the TIC format) can also be used as filters. Since you cannot program any colour changes here, they are used as static filters which do not change over time but can at best be switched abruptly. They are therefore not suitable for fade effects, but very suitable for complex static lightening and shadow effects, transparency effects etc. Their use is again analogous to the vector area fill patterns. You select the fill pattern icon and then the required colour picture as a fill pattern.

3.2.3 Filter functions

In the previous section we have discussed the way filters can be structured and what can be used as a filter. We have repeatedly indicated how these filters could be used. The actual effect of a filter will, of course, depend on the filter function selected. The four available filter functions will now be discussed in detail, giving simple examples for their use. The filters can, of course, also be set independently for each track.

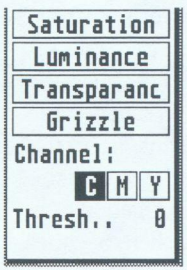
3.2.3.1. Saturation filter

This filter type is selected by selecting the 'Saturation' field. It is a colour filter which can smoothly adjust the colour saturation of the current track. Complete saturation means that the picture will be drawn unchanged. If the saturation is reduced, the picture will be gradually bleached towards white. Transparent picture parts (those not containing any graphics elements or white fill pattern areas with transparency activated) will remain transparent. The saturation filter is therefore used to fade out to white or to fade in from white; all intermediate settings are also possible of course, creating pastel shades according to the strength of the filter - in other words, the colour is more or less bleached. You can, of course, restrict the bleaching to a single colour channel and use the saturation filter in this way for re-colouring. We now have to explain how the filter type and the filter setting affect the saturation effect. The basic rule is very simple: where the filter is entirely black, the picture remains unchanged; where the filter is white, the picture is completely bleached to white.

Let us illustrate this with a simple example: Example: Fading in from a white background Place a colour picture (picture object) on Picture 0 and Picture 49, ideally filling the entire page format (camera format). Now select the saturation filter function and a white area as the filter type. Place this filter via the camera snapshot on Picture 0. Now change the tonal value of the filter to black and do a second camera snapshot for Picture 49. Activate the filter in the 'Animation settings' dialogue. That's all. If you now run the animation through the simulator, you can see how the colour picture is faded in from white. (You will also find this example on disk as an HTM file). This example can be varied in many ways: instead of a simple picture you could take an existing animation track, a digitised film sequence etc. Instead of fading in from all white you could also fade in from red or any other colour. Swap the two camera snapshots and you obtain a fade-out rather than a fade-in. Instead of the simple monotone area you could use a monochrome or halftone picture with additional colour setting. Do not hesitate to experiment in order to become acquainted with the effects and possibilities of the saturation filter.

3.2.2.2. Luminance filter

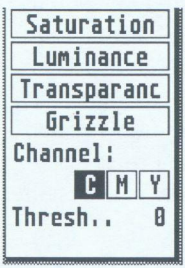
You select this filter type by selecting the 'Luminance' field. This is a colour filter which is used to smoothly adjust the colour intensity of the current track. Complete luminance means that the picture is drawn unchanged. Reduced luminance darkens the picture towards black. Transparent picture parts (those which do not contain any graphics elements or white fill pattern areas with transparency selected) will remain transparent. The luminance filter is therefore used to fade out to black or to fade in from black; all intermediate stages are also possible so that you obtained shaded colours depending on the strength of the filter, darkening the colours more or less. You can, of course, restrict the darkening to a single colour channel and thereby use the luminance filter for re-colouring. We must now describe the effect of the filter type and the filter setting on the luminance effect. The basic rule is very simple: where the filter is completely white, the picture remains unchanged; where the filter is completely black, a complete fade-out to black takes place. Let us illustrate this using a simple example. Example: Fading in from a black background Place a colour picture (picture object) on Picture 0 and Picture 49, ideally filling the entire page format (camera format). Now select the luminance filter function and a black area as the filter type. Place this filter via the camera snapshot on Picture 0. Now change the tonal value of the filter to white and do a second camera snapshot for Picture 49. Activate the filter in the 'Animation settings' dialogue. That's all. If you now run the animation through the



simulator, you can see how the colour picture is faded in from black. (You will also find this example on disk as an HTM file). This example can be varied in many ways: instead of a simple picture you could take an existing animation track, a digitised film sequence etc. Instead of fading in from all black you could also fade in from red or any other colour. Swap the two camera snapshots and you obtain a fade-out rather than a fade-in. Instead of the simple monotone area you could use a monochrome or halftone picture with additional colour setting. Do not hesitate to experiment in order to become acquainted with the effects and possibilities of the saturation filter.

3.2.3.3. Transparency filter

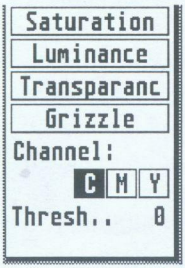
You select this filter type by selecting the 'Transparency' field. The filters introduced so far (saturation, luminance) were pure colour filters, which recoloured the current track before drawing it. The transparency filter is completely different because this filter controls the more or less transparent superimposition of tracks. In short, the transparency filter determines how the current track is superimposed on the already drawn tracks more or less transparently. The transparency filter therefore allows all settings ranging from complete transparency (nothing is drawn at all) to complete opacity (the new track is fully drawn - minus the transparent parts - over the existing tracks). There are many different possibilities because, in addition to the normal superimposition of two tracks you can also use it to construe special filters using suitable objects. You can even use it to "reconstruct" the saturation and luminance filters, as well as many other effects. We must now explain how the filter type and the filter setting affect the transparency effect. The basic rule is very simple: where the filter is completely white, the current track is completely transparent (and therefore invisible); where the filter is completely black, the current track is completely opaque. All intermediate settings effect a more or less transparent superimposition, and the three (or four) colour channels can, of course, be programmed entirely independently. Since the possibilities are almost endless, we shall illustrate them with a few simple examples: Example: Transparent superimposition of colour pictures Place any colour picture (picture object) on Picture 0 and Picture 49, ideally filling the entire page format (camera format). Create a second track (layer) and place another colour picture on positions 0 and 49. Select the transparency filter function and a white area as the filter type. Place this filter, using a camera snapshot, on Picture 0 (in the second track!). Now change the tonal value of the filter to black and place a second camera snapshot on Picture 49. Activate the filter in the 'Animation settings' dialogue (for the second track!). That's



all. If you now run the animation through the simulator you will see how the first colour picture is smoothly faded the second. (You will also find this example on disk as an HTM file). You can, of course, vary this example in many ways: instead of simple colour pictures you can use animation tracks, digitised film sequences etc. Instead of a simple monotone area you can use colour graduations, masks or halftone pictures. You need not, of course, create complete fades: you can, for example, create 'ghost' effects with partial transparency etc. Example: Blurred fade-out Proceed as in the last example. The second picture, however, is replaced by a blurred version of the first one. To do this, load the first picture into an image editor (such as DA's REPRO CD or DA'S PICTURE) and blur it as much as possible. This highly blurred picture now replaces the first picture in the above example. The picture is now faded into a blurred version of itself. You can continue this by adding a third picture consisting of only one colour (ideally the average colour of the first picture) and then fading out transparently into this picture. In all these examples the task was a more or less transparent superimposition or fading of tracks. As already mentioned, however, the transparency filter, in combination with the right objects, can also be used to construe more complex colour filters. The following example shows how you can use it to create the saturation and luminance filters and modify these in many ways. Example: Your own colour filter You proceed again as in the first example. Instead of the second picture, however, you now use a black (saturation) or white (luminance) monotone area. Now the (first) picture is faded out to black or white, exactly as with a fade using the saturation or luminance filters. If you change to a monotone area in a different colour, the filter will be much modified (fade-out to any colour). If you replace the simple area with colour graduations, tinted halftones etc. and combine these with the appropriate filter types, almost unlimited filter possibilities can be obtained, including different fade-overs or filters in different picture areas, light and shade effects and much more.

3.2.3.4. Threshold value filter(or grizzel filter)

All the filters described so far offer wonderful possibilities but they also have one serious drawback: the colour fade-overs usually change from picture to picture by changing ALL pixels. Such global changes are hardly compressible and therefore require a lot of storage space; for this reason, the animations created with them (especially with larger image formats) are impossible to replay in real time with the Player. The application of these colour and transparency filters is therefore restricted to video recordings with single frames or frame sequences (see Part B of this manual). For computer animations which are to be strongly compressed they cannot, as a rule, be used. This does not apply to the fourth filter, the Dither or Grizzle filter. This filter works in a

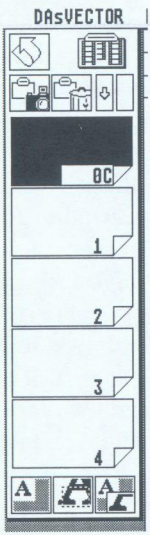


completely different way from those so far described and is also a little more difficult to understand. We shall therefore begin the explanation with an example. Example: Pixel-by-pixel random fade-over using the threshold value filter Construct the animation as described above for the first transparency filter example using two colour pictures. Now we must first construct a random filter. Call up the 'Graduation generator' under Services. Enter your page format as the format, select 'coarse' for resolution and enter the start value of 100 and the end value of 0 under Grey Graduation. For graduation type you select Noise (top left icon). Now generate a greyscale graduation under the name NOISE.TIH. Return to DA'S VECTOR PRO and select this halftone image as the filter. Set the colour to 254,254,254. Select the 'threshold value' filter function and enter the threshold 0. Do another camera snapshot to Picture 0. Now set the threshold to 255 and do a snapshot to Picture 49. Run the animation through the simulator. If you have studies computer animation before, you will recognise the effect. Picture after picture, through randomly arranged pixels, the second picture is increasingly becoming visible, until eventually the entire picture has been faded over. Rather than fading all pixels into one another step by step, a series of randomly selected pixels of the first picture are replaced by pixels from the second picture with each step of the animation. This is why such fade-overs are easy to compress, since only relatively small parts are changed between two pictures in the animation (about 2% in our example). Let us now try to explain the 'threshold' filter effect. The following rule applies to every single pixel of the track: if the threshold value is greater than the colour value of the corresponding filter pixel, the pixel is drawn, otherwise it is not drawn. In our example we have randomly distributed pixels in the filter picture, with all values from 0 to 255 occurring in roughly the same frequency. At the threshold of 0 (in the first picture) nothing is therefore drawn at all. In the second picture the (interpolated) threshold value is about 6; i.e. all pixels for which the filter pixel is smaller than 6 are drawn. In the next picture the threshold value is already 11, so that more pixels are drawn. In the last picture, the threshold value is 255, so that all pixels are drawn for which the filter value is less than 255. Since we wanted a complete fade-over, we have limited the maximum grey value (by setting the colour to 254,254,254) to 254, so that no picture value of 255 appears in the filter picture. This is admittedly not very easy to understand because many different factors are involved. A further example may help and will also demonstrate another possible application of threshold filters. Example: Geometric fade-overs using the threshold value filter We use the same animation again with the fade-over of two colour pictures, as in the last example. As the filter, however, we are now

using a linear colour graduation (e.g. from left to right) with the start colour 0,0,0 and the end colour 254,254,254. The graduation should extend over the entire area of the camera window. Now place a camera snapshot with a threshold value of 0 into Picture 0 and another with a threshold of 255 into Picture 49 and start the animation in the simulator. You can now see how the second pictures is faded in from left to right in strips. Using different colour graduations (or even halftone pictures) you can achieve a variety of geometric fade-overs very easily. They need not, of course, be static pictures like the two colour pictures in our example but could be animated objects in the different tracks. We have consciously restricted all our examples to static pictures because they show the principles involved in different filter procedures very clearly. The threshold value filter contains another particularity: it uses only one colour channel. You can therefore select one of the three colour channels (C, M or Y). This plays no role in the examples above, since the same data were present in all colour channels.

3.3. Additions to the Film clipboard sub menu

The appearance of the 'Film clipboard' sub menu has not changed but new functions have still been added. There are the snapshot functions for the camera and also the much more complex transformations and settings in the 'Animation settings' dialogue. Both will the described in the following chapters.



3.3.1. Snapshot functions for the cameras

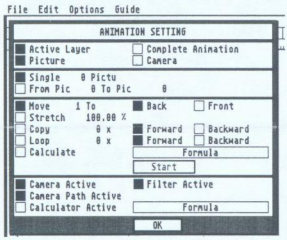
The camera settings require their own snapshot function since they are saved to the film track independent of the objects. The current filter settings is also part of the camera settings. The same icon as for the object snapshots is used. To enable the program to differentiate between the two types of snapshots, the ALTERNATE key is used with all camera operations. This also applies to the Delete function (trashcan icon). There are therefore the following functions:

ALTERNATE + click on snapshot icon: Place camera settings of selected picture into track.
SHIFT + ALTERNATE + click on the snapshot icon: Place camera settings of selected picture into ALL tracks.
CONTROL + ALTERNATE + click on snapshot icon: Bring camera setting of selected picture (incl. interpolated!) onto desktop.
ALTERNATE + click on trashcan: Delete camera setting of selected picture.

CONTROL + ALTERNATE + click on trashcan: Delete all camera settings of current track.
SHIFT + CONTROL + ALTERNATE + click on trashcan: Delete all camera settings in all tracks.
A camera setting placed on a track is indicated by the letter 'C' in the film clipboard entry.

3.3.2. Additions to the animation transformations and settings

The 'Animation settings' dialogue has become much more complicated than in DA'S VECTOR. It now consists of two parts. The upper part contains global functions to edit parts of tracks, whole tracks or the entire animation. Some of these functions were already present in DA'S VECTOR. They have been much expanded though and there are also some small differences in their use. We shall therefore once again describe all functions and their settings in this dialogue. In the top area of the dialogue the range of the current operation is defined. You can decide if only the current track is to be edited or the entire animation. You also choose between editing the 'pictures' (objects in the track) or the camera settings. Certain operations, however, are not permitted for the camera(s) (copy, loops, and calculator transformations). In the second part of the dialogue you select the region to be edited. You can edit single pictures or any region of a track or of the entire animation. In the third part of the dialogue you find the actual operations. To call one of them you need to select the small rectangle before the required operation and then click on Start. Move region: You can move the selected region forward or back (within the track!). Stretch/compress region: You can stretch (>100%) or compress (<100%) the selected region. Copy region: This function lets you make a physical copy of the selected animation region. The copy is always placed behind the last picture of the selected region. You can select the number of copies and, via 'forward' or 'backward', decide if the region is inverted when being copied, i.e. to be running backwards or not. This enables you to achieve the well-known Ping-pong effect: for example, an object is placed in Picture 0 and a transformation of the object in Picture 24. If you now copy this region once with the 'backwards' setting, the object is transformed once and then re-transformed back into its original state. You cannot copy camera settings! 'Looping': With this function you can make virtual copies of the selected animation region. Virtual copies have two important advantages: they need little storage space and the virtual copy is always changed along with any alterations to the original; this

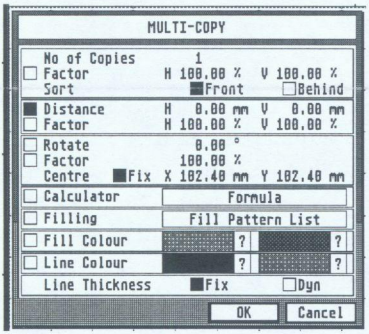


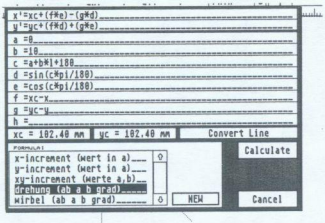
only applies, however, when 'keyframes' are changed. If you subsequently insert a new keyframe into the original region, this keyframe is NOT automatically part of the virtual copy. This can cause strange side effects which you may wish to use on purpose. At any rate, however, care is required when using 'loops'; if you want to play safe you should only add the 'loops' when the original region has definitely been finalised. Copies are always placed directly behind the last picture in the selected region. You can specify the number of copies and decide, with 'forward' and 'backward', if the region is to be inverted or not; this enables you to achieve the Ping-pong effect described above. Transform region with formula - 'calculate': This is an enormously powerful function for editing tracks and animations; you can transform any track regions using the graphics calculator. A new option is available, the running variable 'l' which is used for the relevant picture number. This means that you can design transformations which have different effects on different pictures (see also Chapter 3.5). When you click on the 'Formula' field, the calculator dialogue appears and you can select a formula or create a new one. All keyframes in the selected region are then transformed. The lower part of the dialogue has a different meaning from the other parts. Here no operations are carried out but certain properties of the CURRENT track defined: 'Camera active': When this is selected, the camera is used, otherwise the current page format is used. 'Camera path active': When this is selected, the camera path is used. 'Filter active': When this is selected, the filter functions selected for the camera are used. 'Calculator active': This is a powerful new function which requires detailed explanation. The first 10 tracks (Layer 1 to 10) can be combined using a calculator formula. This formula will be applied when the track is recorded. This makes it possible to use non-linear interpolations, smooth rotations (without many keyframes) etc. with little effort (provided you understand the necessary maths). You can take advantage of a new calculator function, the running variable 'l', which stands for the relevant picture number. This allows you to design transformations which have a different effect on every picture (see also Chapter 3.5). When you click on 'Formula', the calculator dialogue appears but with a small difference compared to the normal calculator: the top entry is 'Animation track'; this is a formula assigned to the current track. When you first call this up, there will of course be no formula yet. You can enter a formula or import one from the library. To do this, select the required formula and click once again on the entry with the CONTROL key pressed. You are asked if you wish to copy the formula to the animation track and enter 'Yes' in reply. Conversely, you can copy a formula from the animation track to the library by selecting an empty entry and then clicking on it again with the ALTERNATE key pressed. You are again asked to confirm your intention. Only

when you have entered a formula will you be able to activate the 'Calculator active' field. The formula is, of course, saved along with the animation. When you click on OK, the settings for the current track will be saved and the dialogue is closed.

3.4 Additions to the multicopy function in the animation editor.

The additions to the multi-copier function have already been discussed in Chapter 2.3. The same additions are, of course, also available in the animation editor, although they have a slightly different meaning here since it is not 'spatial' copies to the work area but 'chronological' copies in the current track (from the currently selected picture number) which are made. It is for this reason that the application possibilities of the extended multicopy function of the animation editor will be briefly explained here. One very important function is the change of fill pattern for every single copy. This allows the incorporation of digitised film sequences or of animations available in single picture sequences from other programs (or even from DA'S VECTOR PRO itself) with only a few mouse clicks. The picture sequences should be available as numbered sequences (filenames with numbers such as PICT0001.TIC, PICT0002.TIC etc.). They can then be loaded, or appended, as a series in the 'Fill pattern' dialogue. This makes them available within the DA'S VECTOR PRO system. If you now create a picture object (a rectangle with a picture as a fill pattern, in the simplest possible example) and then copy it x times using the multicopy function with fill pattern change enabled, the picture sequence will be placed in the track instantly. You can, of course, modify this in many ways: the animation loaded in can run in any shape (rather than a simple rectangle), can itself fly around, etc. Much of this can be taken care of by the multicopy function itself, for example a rotation. Individual pictures, you first enter 35 copies and the second picture of the animation as the first picture for the fill pattern change. Another innovation can be used very productively in the animation area: the inclusion of the graphics calculator in the multi-copier function allows the creation of very complex trajectory and shape transformations with many intermediate steps. The picture-specific running variable 'I' (picture number) can, of course, be used in this (cf. Chapter 3.5). This can claim a lot of memory and storage space, however, if the objects are very complex. In that case, it may be





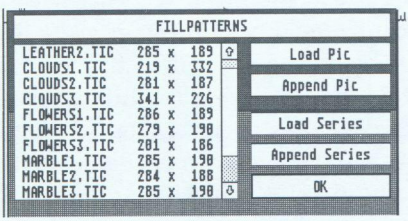
better to use the calculator in the animation track (cf. Chapter 3.3.1).

3.5 Enhanced calculator functions in the animation editor

The calculator itself appears unchanged but there is a small internal modification intended specifically for animations (and indeed only makes sense for this). An additional variable 'l' can be used which corresponds to the picture number in the track. This lets you design incremental formulae which have a different effect on every picture. For normal applications of the calculator in the vector graphics editor, 'l' is always set to the value 1.
Please note when using the multicopy function that the original is placed at the first position (current picture in the track), followed by the selected number of copies. If, for example, you wish to merge in an animation containing 36 i

3.6 Use of the extended fill pattern functions in the animation editor

The extended options for loading, replacing and appending fill patterns were already discussed in Chapter 2.1.4. Here we are only going to point out once again the relevance of these functions for the editing of animations. In particular, the option to load or replace sequences of numbered pictures was implemented with a view to using animations from other software or digitised real film sequences. Thus, for example, the program INSHAPE will in future be able to save picture sequences in the TIC format (currently these still have to be converted to TIC with the picture converter). You will then be able to load animations created with this 3D raytracing and shading package into DA'S VECTOR PRO without problems, combine them with other animation elements, record them at any resolution and replay them with DA'S PLAYER. The same applies, of course, for real film sequences imported into the calculator with the help of a true- colour digitiser (step by step using single frames/pictures, a tedious business so far if done manually). You can replace such sequences in their entirety very simply by clicking on the first picture in the fill pattern list and then replacing a whole series of pictures (and thus perhaps an entire animation) with 'Load series'.



B. Introduction to digital video production with DA'S VECTOR PRO

As already mentioned in the introduction, we consider DA'S VECTOR PRO (among other things) as the core element of a digital video production system. The following chapters will describe what me mean by that and how you will be able to work with it (now and in the future).

1. Basic concepts

What are we to understand by the term fully digital video production system? We have a very concrete idea of such a system: the combination of animated computer graphics (two-dimensional with DA'S VECTOR PRO, three-dimensional with programs such as INSHAPE) with real film sequences, which can also be digitised and post-edited in the computer, scanned photos (or Photo-CD images) etc. All these elements are managed digitally inside the computer, combined and blended together, with music, sounds and language added, and finally recorded to a target medium. Possible target media are film, video, the computer itself and also digital recording media such as CD-I. You will be able to take advantage of all the possibilities of graphics, text and image editing developed in recent years primarily for DTP applications. But what can you use all this for? Video amateurs can create titles and cartoon film sequences, musicians produce their own video clips, advertisers create their own advertising spots, show promotions and video product presentations for their customers and the new multimedia will generate a whole new range of possible applications. Of course you can also create computer films, slideshows, software presentations and similar works to be replayed on computers of all types. A particularly interesting aspect of all this is that it will be possible at a price that had hitherto been unthinkable, since in many cases you will be able to do without expensive studio equipment and still achieve professional quality. In the field of video it will be possible to achieve a quality that is sufficient for many purposes by using up-market consumer products (S-VHS or Hi-8 recorders). Where this is not sufficient, service bureau's can be used as has long been the practice in desktop publishing. We are currently working to build a network of service providers which will record videos at any quality for users of our software. Further details can be obtained from out telephone hotline.

2. The basic method of work

Apart from sound recording, the ingredients of a digital video originate from essentially three sources: firstly, DA'S VECTOR PRO can be used to create complete cartoon films combining all types of text, vector graphics and picture

elements. This provides a purely two-dimensional (despite many "3D" effects) animation studio, which models itself on the classical (but pixel-orientated) animation techniques (cartoons etc.) on one hand, but on the other offers a plethora of innovative possibilities through its new combination of vector graphics with pixel art. Another important advantage of the system is its resolution independence. The target resolution is only defined at the recording stage, including both the picture size and the colour depth (depending on the target hardware). For 3D animations you will also need further software, like the program INSHAPE. The newer versions of INSHAPE are capable of outputting animations as numbered picture sequences in the TIC format. These sequences can be loaded into DA'S VECTOR PRO at one go using the 'Load/append series' function. To incorporate such an animation in a DA'S VECTOR PRO track you must first generate a picture object for the first picture in the sequence; this is most easily done in the Tracer menu using the 'snapshot' function; alternatively you can draw any shape in the path editor and define the first picture in the animation sequence as a fill pattern; the INSHAPE animation can therefore run in any kind of area. This first object is then scaled and positioned as desired and multiplied using the multicopy function in the animation editor; the fill pattern change is enabled and the entire animation is entered into the required track. You can, of course, animate the 'animation object' itself: the 3D animation from INSHAPE can therefore, for example, fly around in space, grow and shrink, be masked and so on. Using the filter technique, it can be modified in its colours or be faded transparently. A possible third ingredient of such digital videos are real film sequences from a video recorder. Of course you need to get these sequences into the computer in the first place. To do this, you require a colour digitiser and a video recorder with first-class freeze-frame and single-frame forward capabilities, which are commonplace with many recorders these days. You now step through each single frame, digitise and save it. Repeat this as often as necessary. At the maximum PAL resolution of 768*576 pixels a single frame requires about 1.3 Mb hard disk space, a second of film therefore about 30Mb. A 2 Gigabytes hard disk can therefore record approximately 64 seconds of a real film sequence (in studio quality). Home users may take fright at such figures but mass storage is getting cheaper and more spacious all the time (take, for instance, the MiniDisc from Sony which will soon be available as a computer drive and can store 160Mb on a disc for around £7). and, in any case, a lower resolution (such as 384*288) is often quite sufficient for home recorders. It is then possible to store around 25 seconds on an affordable 200Mb disk. Once the animation has been recorded, the film sequences can be deleted from the hard disk again and longer sequences will be processed bit by bit anyway. The 'manual digitising' is, however, quite tedious; and for automation it is necessary to have remote

recorder control. We will soon introduce the appropriate hardware and software for this. Once the real film sequence has been saved as numbered picture sequences, it can be incorporated in a DA'S VECTOR PRO track in much the same way as described above for an INSHAPE animation. From now on there will be no loss of quality during mixing, superimposition, cutting, fade-overs etc., since the entire editing process is fully digital until the final re-recording to video. The decisive factors are therefore the quality (resolution, colour depth, purity) of the digitisation and then again the final recording quality. Both depend on the equipment used. All intermediate stages, however, take place inside the computer with virtually no loss of quality. It is even possible, in principle, to 'enhance' the picture material using the methods of modern image editing. Through the use of appropriate picture objects and the filter technique there are virtually unlimited possibilities of combining the elements from different sources: thus real film sequences can be combined with one another or with 3D animations and various animated picture, graphics and text objects, using fade effects, superimposition's, cut-outs etc. As long as the demands on real-time display are not too high, you can record re-playable computer animations from this. As a rule, however, a film will be made up of several part sequences; these can be combined in a film script and given a sound track. More about this can be found in the chapter about the movie compiler. To create a video film, the 'recorder' again has to record in the required target resolution. In addition to the compressed film format you can also generate numbered picture sequences in three different formats. Which method you use will depend on the subsequent use. More about this will be found in the following chapters.

3. Montage and sound track recording of computer films using the movie compiler

The movie player can replay animations, single pictures and sound tracks (in AVR format) in practically any combination. To do this, it needs a control file in the MOV format. For single sequences directly generated from the 'recorder', such a MOV file is also created (provided the 'Start picture' option is enabled). Such a computer film consists, at least, of this MOV file, the start picture in MPC format and the compressed picture sequence in the MAM format. The MOV file can, however, be much more complex and contain an (almost) unlimited number of individual pictures (in MPC format), animation sequences (in MAM format) and even sounds (in AVR format). How do you create such a complex control file? This is where the MOVIE compiler comes in. It reads a film script created by you in the form of a simple ASCII text file which you can create with any text editor. It will then generate from this film script an appropri-

ate control file (of the MOV type) which is required by the player to control the replaying of the film. Such a script must contain a number of commands which the MOVIE compiler recognises and interprets. In addition, you can enter into the script anything you like - as long as you stick to certain simple rules. It can therefore be a real film script describing the entire contents of your film. In between all this, certain commands are inserted which are translated by the compiler into the control file. We can only recommend this approach for the production of more complex films. For shorter and simpler sequences, create these in DA'S VECTOR PRO (and a suitable hard disk recorder for the sound) and then combine them afterwards into a film. You will then begin with a film script, create the individual animation sequences or single pictures one by one, and insert the appropriate commands. In this way, the first sketch gradually turns into a complete, annotated film, which the compiler finally (and in between for test purposes) processes into a control file for the player. Let us first describe the simple syntax of the compiler language: each control command starts with a '#' at the beginning of the line, variables with a '$'. All other lines are skipped by the compiler (interpreted as comments). A command line can also contain comments; these should be separated from the actual command by a semicolon. The commands can be divided into two groups: there are a number of commands which must always be used at the beginning of a film script. They give details about the size (in pixels) and the colour depth (and colour model) of the film, the timing and the sound resolution used (frequency, sample depth, channels etc.). This is the most complicated part, but we have done much of the work for you. Your disks include a file called MOVIEHDR.TXT, which is a text file with all the relevant details and an explanation. You should always take this text as your starting point for your film scripts and insert the required values in accordance with the instructions found there. All these commands begin with a capital letter and continue in small letters, followed by the necessary numerical values (or filenames) separated by spaces. The second group of commands is very simple to understand and to use: it consists of three commands with which you can start the individual elements (single pictures, animations, sound) as well as a PAUSE command. All commands consist of a single small letter after the '#' and a filename (without extender) or an identifier and a number which are separated from the command letter and from each other by at least one space. The number indicates a delay before the next picture or the next sound, and must always be given in 'frames'. The basic time unit is therefore the number of pictures output per second.#b NAME x ; PICTURE: Show picture with the filename NAME.MPC for a duration of x frames. Normally x=1 for the start picture in an animation, but you can also prescribe an intentionally longer pause to give the animation more time to start. For pure slide shows (static pictures) you can also enter here the time each picture is shown. #a NAME x ; ANIMATION: Star animation with

the filename NAME.NAM. 'x' specifies the time in frames. You should know this value - after all, you set it yourself when 'recording' in DA'S VECTOR PRO. You can also enter a longer time (in frames) than that required by the animation to include a subsequent pause in it. If the value is to small there will be output problems. #t NAME x ; SOUND: Start sound sequence with the filename SOUND.AVR. 'x' specifies the time in frames. You should know this value. #p TYP x ; PAUSE: x frames pause (between the end of the last event and the beginning of the next of the same type). Typ = t (for sound) or = b (for picture = event in the picture track). This command can be used for additional pauses. You will certainly need it whenever the first sound or picture event is to be started with a time offset. Variables: You can also use variables or globally defined constants for the time specifications. 26 variables are available in the MOVIE compiler for this purpose, all initialised with '1'. A variable is denoted by a leading '$' character, followed immediately by a small letter (a-z). Variables can be used in particular for globally parametrised pauses, with which the timing can be easily modified by simply redefining a single assignment. The assignments have the form: $a=x , 'x' being any numerical value. Finally a few rules applying to the design of more complex animations: An animation must consist of AT LEAST a start picture (MPC) and an animation file (MAM). Further animation files can be appended without (storage and time consuming) start pictures if you yourself ensure that the end picture of a sequence is identical with the start picture of the next sequence. There is a simple trick to ensure this (given appropriate parameters): Save the last picture of an animation (in the recorder) as a single picture in the TIC format and make this the start picture of the next sequence. This will rarely be necessary, however, with proper planning. You can, of course, also create simple slide shows with static pictures. Here you will only need single pictures in the MPC format. All pictures (MPC) and all animations within a film script must be exactly identical in picture size and colour resolution. The compiler itself does not check for this, because these data will often not yet be available in a test run or on a different computer. The same applies to the sound track: here all samples must have the same frequency, resolution and number of channels. All these parameters are defined globally in the 'Header' for the entire film. You must yourself ensure that the data produced by you (MPC, MAM, AVR) are present in the correct format! Most of these data (but not all) can be checked in the player under 'Film info'.

4. Recording to video

Part of the digital production of videos is, of course, the recording of the final result onto video. Depending on the complexity of the animation, your own quality expectations and the possibilities of the available equipment you need to proceed in different ways. A complete discussion of this topic cannot be

entered here, so we will only describe the basic procedures. To begin with, the signals must be output in video norm and the appropriate screen resolution and a good colour quality. Let us begin with the colour resolution of the computer. You should not use 'dithered' colours with animations because of the interlace effects. For this reason, only 15/16 bit High Colour or 24-bit True Colour are suitable for a satisfactory colour quality. Many affordable graphics cards for the ST/TT computers today offer 15-bit resolution, and True Colour cards are available in several price ranges. An important consideration when choosing a graphics card is that it should be freely programmable so that you can generate clean video frequencies to the PAL standard inclusive of interlace and suitable screen resolutions (4:3 aspect ratio). A graphics card is not needed for the FALCON 030 which offers 16-bit (64k colours) and a high screen resolution to video standard. You will, however, need some additional software to achieve the optimum resolutions of 768*576, 768*288, 384*288 or 384*576, because Atari have unfortunately not built these PAL-Overscan resolutions into the operating system. The output will still not conform to the video standard exactly in aspect ratio or width; at the Falcon's clock frequency a line should display 832 pixels. Additional external hardware should be able to obtain some improvement here. The next problem is the signal encoding. Graphics cards generally supply RGB signals only and these are at best found in studio recorder input. You need to connect an external FBAS or, better, a Y/C converter. Given the appropriate recorder you will generally wish to work with Y/C these days and the Falcon does not support this either but does at least offer FBAS output. The quality of the converter used will have a considerable impact on the quality of the recording. Now we come to a very important question concerning video recording: can we replay an animation in real time on the computer and thus record 'live'? As a general rule, this is only possible with relatively simple animations and lower resolutions (such as 384*288). You simply start the recorder and, at the right moment, the player. As soon as real film sequences, filters, camera movements and similar functions are involved, computer replay and video recording in real time are impossible. Almost unlimited quality can be achieved with single picture recording. For this, you need a suitable (expensive) video recorder or an (even more expensive) laser disk recorder. Using the single step player, you can now record picture by picture, with the control still being manual labour. New hardware and software will make this more comfortable and is now in development. Something else will then become possible: The Falcon 030 will be able to perform single picture recording with many home video recorders, thus allowing highly complex animations in remarkable quality with affordable equipment.

5. Recording to film

Since animations can be calculated to practically any resolution and saved as single picture sequences in the TIF format, there is no obstacle in principle to film recording with appropriately specified slide recorders. The processing times and data volumes, however, would be gigantic. A case of waiting for even more powerful computers...

6. MPEG recording

To be able to digitally store videos and replay them in (quasi) real time, these videos must be highly compressed, involving some loss of data. The MPEG method has established itself as the standard for this, and special hardware exists for this in the form of PC expansion cards. Such hardware is not on the horizon for Atari computers at the moment. However, you can achieve MPEG compression of films created with DA'S VECTOR PRO very simply by connecting the graphics output of your computer (ST/TT with graphics card, or Falcon) with the video input of an MPEG card. You can then generate MPEG-compressed films without much effort in suitable OC formats.