

Hold-And-Modify

From Wikipedia, the free encyclopedia

Hold-And-Modify, usually abbreviated as **HAM**, is a display mode of the Commodore Amiga computer. It uses a highly unusual technique to express the color of pixels, allowing many more colors to appear on screen than would otherwise be possible.

HAM mode has significant technical limitations and therefore cannot be used as a general purpose display mode. It could be used for the digitization of photographs and video frames, or for the display and manipulation of static images. It generally is not used for games or any other task requiring animation of arbitrary parts of the image.



A HAM picture that contains up to 4,096 colors in 352*480 resolution, made in 1989

Contents

- 1 Operation
- 2 Original Chip Set display modes
- 3 Original Chip Set HAM mode (HAM6)
- 4 Advanced Graphics Architecture HAM mode (HAM8)
- 5 Sliced HAM mode (SHAM)
- 6 Advantages and disadvantages
- 7 Third party hardware that implemented HAM techniques
- 8 References
- 9 External links

Operation

An Amiga display is output as scanlines that are displayed sequentially from top to bottom. Each scanline contains pixels that are displayed from left to right. All display modes available on native Amiga hardware, including HAM, use a color palette. Each pixel's color is identified by three components - red, green and blue.

HAM operates by remembering the color of the pixel that has just been displayed to the left of the current pixel, or in the case of the first pixel of a scanline, the last pixel on the previous scanline. Each succeeding pixel may be represented as a new selection from the color palette, or as a modification of the color of the previous pixel. In the latter case, only one of the color components may be modified for each succeeding pixel. Two of the color components are 'held', and one component is 'modified' - hence the name Hold-And-Modify.

HAM may be thought of as a lossy compression technique. The image is 'decompressed' by display hardware in realtime, while the graphics buffer is being displayed.

HAM was only originally put into the Amiga's custom chipset as an experimental mode. The chipset was originally designed to interpret bitplane data as the hue and luminance portions of an NTSC signal. To quote

Jay Miner:

“ Hold and Modify came from a trip to see flight simulators in action and I had a kind of idea about a primitive type of virtual reality. NTSC on the chip meant you could hold the hue and change the luminance by only altering four bits. When we changed to RGB I said that wasn't needed any more as it wasn't useful and I asked the chip layout guy to take it off. He came back and said that this would either leave a big hole in the middle of the chip or take a three-month redesign and we couldn't do that. I didn't think anyone would use it. I was wrong again as that has really given the Amiga its edge in terms of the color palette. ”

HAM is unique to the Amiga and its distinct chipsets. Pre-4.0 versions of AmigaOS can only use HAM mode in the presence of the native Amiga chipset. AmigaOS 4.0 and up, designed for radically different hardware, provides HAM emulation for use on modern chunky graphics hardware so that software can use this mode in the absence of the unavailable native Amiga chipset. Amiga emulators like the UAE family emulate the vintage Amiga chipset and so can emulate HAM.

Original Chip Set display modes

The original Amiga chipset and the Enhanced Chip Set support a possible 16 values for each of the red, green and blue color components, giving a total of 4,096 possible values for each color register ($16^3=4,096$). Each of the 3 components requires 4 bits to store it, so 12 bits are needed to describe any arbitrary color value.

However, the original and enhanced chipsets can only access up to 6 bits per pixel to describe its color. The bits are separated into bitplanes in a part of the memory known as 'chip RAM'. The original Amiga did not have enough memory bandwidth to access more than six bitplanes, and the memory available for the storage of bitplanes was limited.

A conventional planar screen mode would therefore only be able to have 64 distinct colors ($2^6=64$). The original and enhanced chipsets only have 32 color registers. Each entry in the palette could be set to one of the available 4,096 colors, but clearly without additional effort, the screen could still only contain 32 colors.

One display mode known as Extra Half-Brite uses 5 bits per pixel to create an index into the 32 color palette, and one extra bit to identify whether the color should be at full or half luminosity. With judicious selection of the color palette, 64 distinct colors may exist on the one screen, but this is still far fewer than the 4,096 that the Amiga can produce.

HAM allowed all 4,096 of the Amiga's colors to be displayed using just six bitplanes. HAM5 was not as flexible and was not used as widely.

Original Chip Set HAM mode (HAM6)

HAM mode was part of the original Amiga chipset, available in nearly all 680x0 based Amiga hardware. It allows up to 4,096 colors to be displayed at once, at up to 360*576 resolution. HAM6 uses 6 bitplanes, meaning that the buffer employs 6 bits per pixel to encode color. An unrestricted 4,096 color picture would normally be encoded in 12 bits per pixel and use twice as much memory and bandwidth. The horizontal resolution was 320 (up to 360 using overscan). The vertical resolution for NTSC was 200 or 400 lines (up to 238 or 476 using overscan), and for PAL, 256 or 512 lines (up to 288 or 576 using overscan). The frame rate for screen refresh was always 29.97 frames per second (NTSC) or 25 frames per second (PAL). An Amiga that had a flicker fixer could refresh the monitor at twice this frame rate, but each frame is being shown twice; the rate at which the internal buffer is being output does not increase.

In HAM mode, pixels are represented using 6 bits each; 2 bits are used as decision-making "control" bits, and 4 bits are used as "data" bits. The resulting color for a pixel would be the result of following the instruction of the control bits. The 2 control bits allow for four possibilities, which are as follows:

- **Set:** Use the 4 bits of data as an index into the first 16 colors in the color palette. Use that color for this pixel.
- **Modify Red:** Hold the green and blue color components of the previous pixel. Use the 4 bits of data as the new red color component of this pixel.
- **Modify Green:** Hold the red and blue color components of the previous pixel. Use the 4 bits of data as the new green color component of this pixel.
- **Modify Blue:** Hold the red and green color components of the previous pixel. Use the 4 bits of data as the new blue color component of this pixel.

PowerPC based Amigas, and Amiga emulators running on non-Amiga hardware, are only able to display HAM mode by emulation. Viewing a HAM image on any other architecture requires programmatic interpretation of the image file, since no other computer architecture displays images using the HAM technique. However, software-based interpretation does produce identical results, setting aside the normal differences in color fidelity between different video cards. Recent graphics cards which support shader languages such as OpenCL could do this within the video chips, but there is little need.

Advanced Graphics Architecture HAM mode (HAM8)

With the release of Amigas with the Advanced Graphics Architecture, the original HAM was renamed "HAM6", and a new "HAM8" mode was introduced. HAM8 can display many more colors on the one screen, from the 16,777,216 that the AGA chipset can produce. The vertical display resolutions for HAM mode are the same as for HAM6. The horizontal resolution could be 320 (360 with overscan) as before, doubled to 640 (720 with overscan) or even quadrupled to 1280 pixels (1440 with overscan). The AGA chipset also introduced even higher resolutions for the traditional planar display modes.

The theoretical number of colors that can be displayed simultaneously using HAM8 was widely reported to be 262,144 colors, but this number could be higher. A HAM8 image cannot display any arbitrary color at any arbitrary point, since every pixel relies on either a limited palette or relies on at least two components of the previous pixel.

The total number of pixels in a HAM8 image cannot exceed 829,440 (1440*576) using PAL modes but can exceed 1310720 (1280x1024) using Indivision AGA flickerfixer. If the values of the colors in the color palette were chosen wisely, a HAM8 image could be produced that used a different colour per pixel. To make the use of HAM8 practical, the design of the image will take into account the limitations of the format. Designers may also choose to 'slice' the image (see below) in order to remove some of these restrictions.

With the Advanced Graphics Architecture, the Amiga's graphics quality was improved. Instead of 4 bits per color component, the Amiga now had up to 8 bits per color component, which offered 16,777,216 possible colors. The 6 bits per pixel (implemented as six bitplanes) was raised to 8 bits per pixel (eight bitplanes), allowing for a 256 color palette in conventional operation. A HAM7 mode is also available but not widely used.

HAM8 worked in the same way as HAM6, using 2 "control" bits per pixel, but now there were 6 bits of "data" per pixel instead of 4. The *set* operation could now select from a palette of 64 colors instead of 16. The *modify* operation could modify the 6 most significant bits of either the red, green or blue color component - the 2 least significant bits of the color could not be altered by this operation.

Sliced HAM mode (SHAM)

Sliced HAM mode, also known as **dynamic HAM** mode uses the Amiga's standard HAM6 color mode, 6bitplanes (4 bitplanes for the base colour palette of 16 colours and 2 bitplanes for modifying the base colours). It requires the use of the 'copper' coprocessor to increase the number of colors available on screen to the full 4096 palette by reprogramming the color registers every scanline. Slicing was common when using copper graphics in demos, image manipulation, and certain games. It is also possible to reprogram color registers at arbitrary points along individual scanlines, meaning that individual scanlines can be sliced in the same way.

SHAM expands the colour palette of the overall screen, each scan line can have a unique 16 colour base palette from 4096 possible colours. The advantage of SHAM mode is in having a more precise representation of the image's intended colors, since it would not be limited by the palette choices made for the previous scanline, thus helping to reduce HAM6 colour fringing that occurs more often when an image uses just a single 16 colour base palette for all scan lines.

Dynamic HiRes uses a similar palette changing technique. Where SHAM is limited to lores HAM modes, Dynamic HiRes uses the 4bitplane (16 colour) high resolution modes. Each scan line can have a unique 16 colour palette. Unlike the lores HAM6 modes, the hires screen modes of the original chipset have no ability to modify the base colours to get more than 16 colours per scan line.

The SHAM idea was deprecated when HAM8 was introduced,^[1] since even an unsliced HAM8 image has far more color resolution than a sliced HAM6 image. However, SHAM remains the best available HAM mode on those Amigas with the original chipset.

Modern graphics hardware does not have an Amiga-style copper, so in order to correctly display sliced HAM images, the Amiga hardware must be emulated. The programming required to do this can be computationally expensive, since to emulate a screen that is arbitrarily sliced, the only way to make certain that the emulation is correct is to emulate every bus cycle. However, if the goal is merely to display a SHAM image on a non-Amiga platform, the required color values may be pre-calculated based on the palette entries that are programmed via the copperlist, regardless of whether the palette is modified in the middle of a scanline.

Slicing an image no longer has any practical value on modern graphics hardware, since the color fidelity and bandwidth of modern hardware do not impose the limitations that slicing is designed to circumvent.

Advantages and disadvantages

When the Amiga was launched in 1985, HAM mode gave it a significant graphical advantage over competing systems. It allowed the system to display all of its 4096 potential colors, though with the aforementioned limitations. This far exceeded the then PC state of the art, which was EGA with 16 colors from a palette of 64. Even the newer VGA released in 1987 with its flagship games mode, Mode 13h, had 256 colors from 262,144. The Amiga therefore had a way to display digitized photographs and rendered 3D images. HAM mode was also frequently used to show off the Amiga's ability in store displays and trade presentations, since competing hardware could not come close to the graphical quality.

However, HAM mode does not allow two arbitrary colors to be adjacent to each other. Given two arbitrary colors, it can take up to three pixels to change from one to the other, which potentially reduces the horizontal chroma resolution, from 704 pixels down to as low as 240 (for comparison, contemporary technologies LaserDisc and VHS had chroma resolutions of 40^[*citation needed*]). When a horizontal scanline undergoes a sharp change in color from one color to another that is not in the palette, a noticeable multi-hued "fringe" will appear. The quality may therefore not be acceptable to all users and for all applications.

Furthermore, the HAM screenmode is difficult to use in games or any other application where the contents of the screen need to be modified or animated. If an arbitrary image is moved to any arbitrary place on the screen, the colors will probably not appear correctly. If the left-most portion of the image that has been

inserted contains any 'modify' pixels, or if the image immediately to the right of the moved image contains any 'modify' pixels, those pixels will probably not be set to the appropriate value before being expressed. Accordingly, an attempt to move an image around the screen (such as with the use of a blitter object) will create noticeable fringing at the left and right borders of that image, unless the graphics are specially designed to avoid this.

Therefore, any application of HAM mode requires careful planning of the palette and image. Generally, images that were created for display in HAM mode were designed to incorporate subtle horizontal colour gradients, avoiding vertical lines and contrasts. A game or demo that used HAM mode would have to have its graphics specially designed to avoid fringing, and there would be limitations on the design of the game or demo. The left-most pixel of every line of a scrolling playfield, and the left-most pixel of every blitter object, would have to be a 'set' pixel, and the palette would have to be designed so that it incorporates every such left-most pixel.

An additional problem with using HAM for games is that games that were developed for the Amiga were often also developed for similar systems, such as the Atari ST. The ST used the same CPU as the Amiga, had a comparable amount of RAM, and had similar display options. Although the ST compared unfavorably in some areas (particularly sound generation, and the fact that the original ST had no blitter), a game that was developed for the Amiga's planar modes could be ported without having to rebuild the game from the ground up. However, any graphics engine designed to use HAM would have to be completely refactored for the ST and other systems.

The introduction of eight bitplanes and increased color resolution with the Advanced Graphics Architecture meant that a conventional planar image would have a palette of 256 colours with significantly higher color fidelity. Arguably, the original HAM mode became far less attractive to users of an AGA machine, with its limited palette and limited color resolution. However, it was still included for backward compatibility.

The new HAM8 mode was arguably far less useful to the AGA chipset than the HAM mode was to the original chipset, since a planar 256-color palette greatly increased the options to the artist, and a planar mode does not suffer from the fringing of HAM mode, that in turn limits arbitrary animation and so on. A well-programmed sliced planar mode could prove to be more useful than HAM8. The original purpose of HAM, which was to allow more colour resolution despite limited video buffer size and limited memory bandwidth, was no longer as relevant.

Most modern home computers are easily capable of very high resolution truecolor displays without the need for HAM techniques, and so HAM is not used at all in any modern hardware. Planar (paletted) graphics are also obsolete for general purpose computing, since modern graphics hardware has enough memory bandwidth to used packed pixels.

Third party hardware that implemented HAM techniques

A device produced by Black Belt known as HAM-E was able to produce images with HAM8 color depth at low horizontal resolution from an Amiga with an Original Chipset.^[2]

The Amiga would be set up to produce high resolution images (640 pixels wide, 720 with overscan). This required the use of four bitplanes at 70ns per pixel. The first few lines of the image encoded information to configure the HAM-E unit. Then each pair of pixels was encoded with information for the HAM-E unit, which converted the information into one 140ns pixel (generating an image 320 pixels wide, or 360 with overscan, at a color depth of eight bitplanes). The quality of HAM-E was thus comparable to a low-resolution HAM8 image. The HAM-E technique exploited the fact that a high resolution image with four bitplanes delivers a third more memory bandwidth, and therefore a third more data, than a low resolution image with six bitplanes.

References

- ↑ Standards and specs: The Interchange File Format (IFF) (See HAM and Cheese)[1] (<http://www.ibm.com/developerworks/power/library/pa-spec16>)
 - ↑ Black Belt HAM-E (<http://www.amiga-hardware.com/showhardware.cgi?HARDID=481>)
- *Amiga Hardware Reference Manual*, Commodore-Amiga
 - *Specification for the Advanced Amiga (AA) Chip Set*, Commodore-Amiga

External links

- An animated demo rendered in HAM mode: HAM-6 (http://www.randelshofer.ch/animations/animseric_w_schwartz/AmigaHopAnim.HAM.html) version, and HAM-8 (http://www.randelshofer.ch/animations/animseric_w_schwartz/AmigaHopAnim.HAM8.html) version (Java required).

Retrieved from "<http://en.wikipedia.org/wiki/Hold-And-Modify>"

Categories: Amiga | Color depths

- This page was last modified on 24 March 2011 at 16:33.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.

Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.