



**AMERICAN
INTERACTIVE
MEDIA**

A PhilipsPolyGram Corporation

AIM Technical Note #65

AIM High Resolution Image Tools

Frank Vitz

May 2, 1991

The CD-I community needs techniques for capturing, converting, and displaying still images of the highest possible quality for inclusion in a variety of CD-I applications. The tools discussed here were developed to provide a production pathway to QHYB images; however, they may also be used wherever producers need to produce high resolution images.

Copyright © 1991 American Interactive Media.

All rights reserved.

This document is not to be duplicated or distributed without written permission from
American Interactive Media.

AIM High Resolution Image Tools

OVERVIEW

The CD-I community needs techniques for capturing, converting, and displaying still images of the highest possible quality for inclusion in a variety of CD-I applications. Such techniques involve processing images of better quality than that provided by standard video frame-grabbing techniques and of greater resolution than that which can be handled by existing CD-I utilities. We have written a suite of programs, detailed below, for conversion of electronic pre-press image data into CD-I IFF files. Included are utility programs for converting the images to CD-I format and scaling, rotating, filtering and finally encoding them into QHYB format. QHYB has double the vertical and horizontal resolution of normal DYUV. There are also programs for generating QHYB displays on a base-case CD-I system. These programs are now available as production utilities. While these programs were developed to provide a production pathway to QHYB images, they may also be used wherever producers need to process high resolution images. This paper outlines the process of acquiring high resolution images and provides a summary of the capabilities of these new tools.

INTRODUCTION: WHY USE HIGH RESOLUTION?

A good place to start is with the basic question: Why bother capturing images at high resolution? Obviously, if you plan to use an image format like QHYB, you need to start with high resolution source images. But there are also several other reasons that are independent of the desired target resolution.

One reason results from a cardinal rule of basic image processing: most transfer and conversion processes that are applied to an image result in some unavoidable degradation of the image. This is especially true in the realm of film and analog video in which the image quality worsens with each copy. It is also true in the digital domain when we apply filters or encoding algorithms. Only by starting with images of greater resolution than that of the delivery format can you be sure of a final image that realizes the best possible quality the format can offer. This applies equally to spatial resolution, color resolution, and color accuracy.

A second reason is the fact that a base case CD-I player is capable of displaying better images than those normally obtained by video frame grabbers. Video cameras are inherently quite noisy devices. You can prove this to yourself by

attempting to grab a frame of a flat, single color background. If you examine the actual pixel values obtained, you will see variations in the numbers from one pixel to the next. This random noise reduces the potential sharpness and dynamic range of the image. Even if you provide consistent lighting, the image exhibits variations in brightness across the frame. And there are geometric distortions introduced by the camera as well. Using a very expensive camera and digitizer can reduce these effects, but it does not eliminate them. Because the CD-I player is a digital device, it faithfully reproduces these artifacts.

A third reason is to provide flexibility in the utilization of the images; a high resolution image leaves room for cropping, panning, zooming, or conversion into a variety of formats. Source images archived in a high resolution format can be reused later for a higher resolution version of the same application.

Fortunately, there are ways to capture still images that are much higher resolution than can be displayed on a CD-I player. When the extra resolution is needed, a desktop scanner or high-end laser drum scanner can capture images at very high resolution. These devices have high spatial and color resolution and avoid most, if not all, of the artifacts described above.

Of course, many applications may not need such high resolution. For example, if an application requires only normal resolution DYUV, a video frame grabber may provide adequate spatial resolution, because the typical video resolution is double that of normal DYUV. Even so, the color range is not as good as that obtained with a good scanner. If the original source material is on video tape to begin with, then there isn't much choice.

HIGH RESOLUTION IMAGE PROCESSING

For the sake of convenience, the task of acquiring, processing, and displaying high resolution still images can be broken into eight basic steps:

- Conversion of high resolution scanned images to CD-I IFF format
- Color correction
- Image rotation and mirroring
- Image scaling
- Vertical filtering to reduce interlace flicker
- Histogram equalization (contrast enhancement)

- RGB to DYUV conversion
- QHYB display and cross dissolve

Conversion of High Resolution Scanned Images to CD-I IFF Format

The publishing industry uses high resolution color scanners to process images for print applications. These systems offer the best possible image quality. Many electronic pre-press systems support the ANSI standard IT8.1-1988 for the exchange of color picture data. This standard provides the DDES tape and UEF file formats for this purpose. AIM and other CD-I production houses can have transparencies scanned by a service bureau and the resulting image files delivered to them on 9-track magnetic tape in DDES/UEF format. These images can be extremely high resolution (we have tested up to 2000 line). The UEF files must be converted into CD-I format to be useful in our applications.

The program `ueftorgb` has been written to address this need. It reads a high resolution UEF format file set and outputs a CD-I IFF, v.99 RGB file of the same resolution. (See `ueftorgb` manual pages.)

The program `rawtorgb` is a second option for this purpose. It reads a raw RGB image file and converts it to a CD-I IFF, v.99 RGB file of the same resolution. (See `rawtorgb` manual pages.)

Color Correction

When an image is displayed on a CD-I system, it is desirable for the colors to appear as much like the original as possible. This is a very difficult task to handle objectively, because the image looks different on different monitors. The digitization process has a big influence, as well. There are calibration devices available for adjusting the color balance of video monitors, but they are relatively expensive. Some desktop publishing and paint systems offer algorithms for performing color adjustments or corrections. We have not yet written any programs to address this area. We try to preserve as much of the original dynamic color range of the image as possible and try not to introduce any numeric shifts or offsets. While this is not a rigorous approach, we have obtained subjectively reasonable results so far. The whole science of the accurate reproduction of color for CD-I is still an open field for development and standardization.

Image Rotation and Mirroring

When images are scanned by a service bureau, a common mistake is to insert a slide into the scanner upside down or to flop the transparency taped to the scanning drum one way or the other. To add to the confusion, there are several different ways to define the orientation of the data in the file. The net result is that your image comes out upside down or backwards.

The program `rgbrotate` addresses these problems. It reads any resolution CD-I IFF RGB file and provides the means to rotate it 90, 180, or 270 degrees or to mirror it vertically or horizontally. (See `rgbrotate` manual pages.)

Image Scaling

The main reason for scanning images at high resolution is to avoid sampling artifacts and to obtain the greatest possible detail and color range. With a high resolution image, it is also possible to provide pan and zoom features in an application to explore an image that is larger than full screen. Regardless of the target resolution, a means must be provided to arbitrarily resize high resolution images into the appropriate size for an application with a minimal loss of quality. There are several algorithms available for this. In each case, it is useful to imagine a grid of pixels at the target resolution that overlay the denser grid of pixels that form the original source image. One measure of a scaling algorithm is the accuracy with which it considers the contribution of the source image pixels within the boundaries of each output image pixel.

Pixel replication is a quick approximation method that yields very poor results. Each output image pixel is computed by finding the one source image pixel that is closest to the center of the new pixel and simply using it. The resulting image shows severe sampling artifacts.

The so called "Nearest Neighbor" algorithm provides better results. For each output image pixel, the algorithm finds all the nearest neighbor pixels surrounding it in the original image and computes a weighted average based on their distance from the center of the new pixel. It yields decent results when the target resolution is close to the source resolution because, in this case, the final image includes a computed contribution from most of the source pixels. When the change in resolution is great, sampling artifacts again show up because the algorithm skips all but those pixels which are right next to the output image pixel centers.

Pixel integration yields the best results. Each target pixel covers some number of pixels in the source image. This algorithm computes the exact area of each source pixel covered by a particular target pixel and sums their contributions to create the output image pixel. This minimizes the effect of sampling errors,

because the algorithm accurately computes and sums the contribution of all pixels in the original image, regardless of the change in scale.

The program `rgbscale` addresses this need. It reads a high resolution CD-I IFF RGB image file, rescales it and writes out the result as a new RGB file. It uses either pixel replication or pixel integration. `Rgbscale` also handles pixel aspect ratio differences and windowing into the source image.

Vertical Filtering to Reduce Interlace Flicker

When an image with high vertical detail at high contrast is displayed on an interlaced video device, pronounced flicker effects are often observed. This is the so-called inter-line flicker problem, and its occurrence is not limited to CD-I players. Filtering in the vertical axis of an image can reduce the flicker effect substantially by reducing the contrast between adjacent scan lines. The benefit of interlace mode is that it provides double the vertical resolution.

The program `rgbfilter` addresses this need. It implements Norman Richard's vertical filtering algorithm with an adjustable filter factor. (See AIM Technical Note #42 and `rgbfilter` manual pages.)

Histogram Equalization (Contrast Enhancement)

Regardless of the technology used to acquire them, scanned images often exhibit inefficient use of the range of values available. It is sometimes desirable to reassign pixel values so that the dynamic range of the image increases or is shifted. In many cases, histogram equalization algorithms can achieve this effect for continuous tone images. The result is improved contrast in the image. However, histogram equalization does not always work well; results are dependent on the image content. An additional use for histogram equalization is to constrain the image data to fall within the specified CCIR limits for white and black levels.

The program `rgbequal` helps deal with these problems. It analyzes a source RGB image to determine its histogram and then scales or clips the values of the pixels to fall within a specified range. (See `rgbequal` manual pages.)

QHYB Encoding

Normal resolution DYUV provides high quality color images. However, it suffers from several limitations, including a lack of full horizontal and vertical resolution and artifacts from the delta encoding process. If normal DYUV were the best that CD-I could do in terms of resolution, our concern

with high resolution would be pointless, because DYUV is half the resolution of most video sources. The Green Book* defines an extended case image format that overcomes these limitations. The extended case is called DYUV+QHY. In this format, a Quantized High Y image is combined with a normal DYUV image to produce an image with double the horizontal and vertical resolution of normal DYUV.

Unfortunately, DYUV+QHY is defined *only* for extended case players, because it must be decoded by special video hardware. However, there is a way to achieve the same effect on a base case player via software. This technique is called QHYB. It uses the trick of combining image data from two planes of memory rather than special hardware. QHYB provides a means to increase the luminance resolution of an image by mixing a DYUV image in one plane with a double resolution CLUT4 luminance correction image in the other plane at display time. This combination essentially doubles the horizontal resolution and corrects the most visible errors introduced by the DYUV encoding. When coupled with field interlace to double the vertical resolution, QHYB has the potential for greatly improving DYUV image quality. Properly encoded QHYB images rival 24-bit RGB images in color quality. At 768x480 resolution, the crispness of detail is excellent. The main drawback is that image file size is basically quadrupled for the best resolution format and load times are increased proportionally. (We are working on techniques to reduce the file size via compression. See below.)

The program `rgb_qhy2` implements a test case QHYB image encoder based on work done by Norman Richards at Philips Research Labs. As input, it takes an RGB image at any resolution up to 768x560 and encodes it into a QHYB file set. The current version achieves the maximum possible resolution by creating an interlaced DYUV file and an interlaced CLUT4 file. Future enhancements may include techniques for compressing the data to reduce storage requirements. (See manual pages for details.)

QHYB Display and Cross Dissolve

The final step in the process is displaying the QHYB images on a base case player.

The programs `qhyb_dsp` and `qhyb_cross` demonstrate this functionality. These programs use the standard AIM libraries for support and to read IFF files. `Qhyb_dsp` displays a single image and allows the viewer to examine the DYUV and QHY components separately. `Qhyb_cross` demonstrates a technique for cross dissolving between two QHYB images (kind of tricky since

* Compact Disc Interactive Media Full Functional Specification, N. V. Phillips and Sony Corporation.

one QHYB image uses both planes!). These programs provide a starting point for producers wishing to use QHYB in their titles.

FUTURE ENHANCEMENTS

The DYUV+QHY defined in the Green Book is for extended case players. QHYB is not yet officially defined, and Norman Richards has pointed out several possible variations. These are discussed in AIM technical Note 52 "Notes on QHYB: Base Case QHY." We plan to investigate the benefits and trade offs of these formats and incorporate our findings into the encoding and display programs described above. If QHYB proves to be useful for real CD-I applications, it will need to be standardized and incorporated into the CD-I pantheon. Hopefully, the tools described herein will provide a basis for evaluation by the CD-I community. Ideas or feedback from those of you who try working in this new format will be incorporated into the standardization process and the results included in the latest CD-I IFF specification.