Welcome to BLACKART Help

www.terrainmap.com

BLACKART is an interpolation tool for converting elevation contour files to digital elevation models. The program accepts as input either an ASCII row echelon elevation file (.txt suffix); an Arc View shape file triplet (consisting of a .shp, .shx and .dbf files produced by the raster to vector conversion application Wintopo Pro or Arc View); an ASCII index file of ordered x y z triplets (.ind suffix), or a .bmp or .jpg topographic contour map. (Note: If using shapefiles, the shape file triplets can be in any directory but they must all be in the same directory.) The integral BLACKART graphics package can assist in the preparation of input arrays from .bmp or .jpg maps.

BLACKART is not a generalized raster-to-vector conversion utility. It does not produce a vector image at all. Since the interpolation input files required by the BLACKART solvers are sparse matrix representations of equations derived from raster
elevation grids, no vector image files are needed or produced by this utility.

What BLACKART will do is convert graphic image files of unencoded elevation contours (or other non-graphical representations of such contours) into raster encoded elevation files by use of contour line tagging and other custom algorithms designed expressly for this purpose. It will then interpolate the encoded contour line files into raster elevation grids (by computing the elevations between the contours) using the Franklin algorithm. This produces a terrain representation free from the unwanted artifacts common to many other interpolation algorithms.

BLACKART will also fill in holes (null data areas) in SRTM and ASTER DEMs using one of several user-specified substitution and interpolative techniques. Currently, BLACKART can read SRTM-3, SRTM-1, SRTM30, ASTER and DTED0 DEM file formats.

Description of methods for preparing input files for interpolation:

1) The program has its own native utility for importing contour line .bmp or .jpg graphics files and tagging the contour lines of the input files with elevation values. This way, if your map is a .bmp or .jpg contour map, BLACKART can process the map and prepare the interpolation input grid from the elevation contours. The graphics preprocessor is loaded automatically whenever a .bmp or .jpg file is opened.

2) The program can also use a Bresenham line drawing algorithm to convert the polylines

produced by Wintopo Pro (in the form of an input shape file) to an elevation contour file. BLACKART should also be able to handle shape files produced by Arc View. Just make sure to use only polyline (data type 3) objects. You will be able to work in UTM coordinates. BLACKART will perform a coordinate transformation to graphics coordinates. The graphic coordinate system origin is established at the corner of the image and the units are scaled so that the maximum graphic coordinate is MAXSPAN. You will be prompted to set this parameter when you open your shape file. The default value is 1200, meaning your image will be 1200 pixels, maximum. Using larger values will yield a larger image but may also cause an out-of-memory error. The original UTM system is not preserved, but may be recovered if you save your file as a USGS ASCII DEM as the program will prompt you for the corner tie point and grid spacing.

3) Alternatively, program will accept a space separated ASCII (z-value only) elevation contour grid as an input file.

(Note: This is simply an array of elevation values in row and column echelon format. In this case you must enter the row and column numbers manually when prompted by the program. If you make a mistake in specifying the correct row and column values, the program will at best produce a corrupt output or at worst overflow an array boundary and cause a segment violation exception. If you are lucky, this will only cause the program to terminate.)

4) The program can also accept input in the form of an index file. This format must have the suffix '.ind' to be interpreted by BLACKART and must have one line for each elevation posting of the form:

```
x0 y0 z0
x1 y1 z1
x2 y2 z2...etc.
```

Index format can accept coordinates in the x-y plane regardless of the magnitude of the coordinates because the user-supplied x and y coordinates are automatically transformed into graphic coordinate space. If necessary, the user coordinates may be scaled.

The scaling may result in some data loss if the transformation effects a compression of the data. In addition, the original coordinates are not maintained within BLACKART for subsequent processing, only the transformed coordinates. However, the x and y translational values (minRows and minColumns) and the x and y scale factors (scaleX and scaleY, if used) are reported to the diagnostics screen so that the original coordinates can be recovered if necessary. After processing, the graphics coordinates can be transformed to UTM coordinates by supplying a southwest corner tie point and a grid spacing.

Regardless of the technique for creating the input file, the program then uses the Franklin over-determined Laplacian algorithm to compute a surface approximation (i.e. a computation of the missing elevations between the contour lines.)

Detailed Description of the BLACKART Graphics Preprocessor.

The best way to prepare an input file is probably the .bmp or .jpg graphics file input option. The idea is to input a .bmp/.jpg raster contour map into BLACKART, tag the contour lines within the application, and then convert the tagged raster image to an input array. Users are able to input a graphics file and manipulate it by thinning, zooming, clustering, filtering, tagging, and other image processing utilities. These tools are used to create a BLACKART input array for the interpolative solvers.

The program has the capability to import 24 bit .bmp topo maps. The user can for example apply a sixteen color bandpass filter that will eliminate all pixels that do not have one of up to sixteen specified colors. This feature is useful in separating the contour lines from other raster data on scanned contour maps.

The user can then employ the native Pixel Query feature to determine the decimal integer RGB color values of the pixels that make up the contour line. Then BLACKART can be used to filter out all colors except those specified. The pixel values are input to the filter dialog box in integer or hexadecimal format in the order RGB, i.e. 0Xffffff is white, 0Xff0000 is red, etc.

An article on the use of the input utility is available at http://www.terrainmap.com/rm35.html. A description of the graphics utilities is listed below:

1) Load the .bmp or .jpg file into BLACKART using the dialog box provided.

2) If your input file does not require contour line bandpass filtering, binarize your input contour .bmp image so that the contour lines are black (0x00) and the spaces between are white (0x255). This can be done using the utility provided in the BANDPASS graphics processor module. The binarization tool will prompt you for the background color value. It will convert any pixel with a value less than this value to 0x000000.

3) If your input image requires processing, you may wish to load a reference image. This will load the specified file into a new window. In this way you can maintain an unchanged copy of your input file for inspection as you strip away the non-contour information from your process image.

4) If you wish to subset your image, select Graphics|Subset|Subset Process Image. Click the mouse on one corner of the desired subset and drag the rubber

band box to the other corner. When the mouse button is released, the subset will be extracted.

5) If you wish to separate the contour layer by bandpass filtration, first select Image Processing|Query Pixel Value|Enabled. This turns on the pixel query utility. Select a contour line pixel and click. The 24-bit RGB color value of that pixel will be returned. BLACKART will store this value and up to fifteen other pixel color values for later use. Repeat for up to sixteen different color values corresponding to just the contour line pixels. (If more than sixteen pixels are selected, the oldest value is discarded and the newest one is substituted.)

6) Select Image Processing|Filter|Value and enter the pixel values that you noted down in the previous step, or else click the Load Values button. If the Load Values button is clicked, the sixteen stored values will be automatically loaded into the text boxes. Note that there is a Value Toggle check box. If the box is not checked, the pixel values in the text box will be kept and all others discarded. If the box is checked, only the values in the edit boxes will be discarded and all others will be kept.

7) Alternatively, you can filter pixel values by range. In this case, select Image Processing|Filter|Range and enter the numerical range of pixel RGB values that you wish to keep. For example, entering 0xff0000 for the low value and 0xffffff for the high value will pass all the values between 0xff0000 and 0xffffff and remove all the rest.

If you check the Range Toggle box, then the filter will work in the opposite sense, that is in the above example it would discard all of the pixel values in the specified range and keep the rest. (Often it is convenient to use one mode or the other in successive filter iterations in order to pass the optimal pixel values.)

If you want to filter only one pixel value, first enable the pixel query utility by clicking Image Processing|Query Pixel|Enabled and then click on that pixel. Select Image Processing|Filter|Range and then select Load Value. When you click Submit, the specified pixel value will be removed.

8) The Red, Green and Blue byte filtering tools let you screen pixel values by their individual RGB byte characteristics. The Red Byte Filter for example prompts you for three byte values. The filter screens all pixels with Red bytes > Red Byte AND < Green Byte AND < Blue Byte. This tool allows screening large groups of pixels that are either reddish, greenish or bluish. The selected pixels will either be converted to the background color (created by enabling the query tool and clicking on a pixel) or to the target color (selected by entering an integer value in the edit box and checking the toggle box.)

9) High pass and low pass filters are available under the Image Processing|Filter menu selection.

10) An unsharp mask tool is available under the Image Processing|Filter menu selection. The unsharp mask is an interesting name for an algorithm that does just the opposite of what it sounds like. It is thus named because it “sharpens” an image (or rather increases its contrast) by first creating a blurred image (actually Gaussian smoothed image) and then sharpens the base image by subtracting a portion of the blurred image from it. (Sounds weird but it works).

In order to use the tool, select Image Processing|Filter|Unsharp Mask. A dialog box will appear asking you for a sharpening constant. Choose a decimal value between zero and one. The lower the number, the more extreme he contrast. Start with something like 0.80. You can apply the mask multiple times.

11) It is possible to convert the image to a grayscale image at any time. Grayscale images have the same byte value in each of the Red, Green and Blue color bytes.

12) A powerful feature for extracting contour lines from raster topos is pixel classification or clustering. Classification covers a variety of geometric and statistical techniques for reducing the number of discrete pixel colors in an image to a more manageable number. For example, it is possible to reduce a 24-bit image with 16 million possible colors in its color space to an image with only say, six discrete colors. This may make filtration of the non-contour line data from the contour lines easier.

To use the classification tool, select Image Processing|Pixel Classification. A dialog box will appear. You can choose from (at the moment) two computational methods for computing the cluster centroids (k-means Mean and Median) and four methods for computing the distance vector between two pixel vectors in RGB color space.

In addition, you must select the number of clusters (i.e. the number of colors that the image will be reduced to) and the number of computational passes. It will take a few seconds or longer to process larger images.

Important: The number of passes is NOT the same as the number of iterations that that algorithm uses to place pixels into clusters by least squares minimization of distance vectors. In order to get the optimal match between the initial colors and the cluster assignments, BLACKART will (if requested) automatically run the computation for multiple passes, each time storing the distance sum and returning the solution with the lowest sum.

The algorithm employed is designed for biological applications whereby attempts are made to group genes with similar characteristics and therefore (hopefully) similar functionality. This level of processing is usually unnecessary in image processing applications. Since running multiple passes may take a lot of time, try a single pass first and then try more passes if you are not satisfied with the color assignment. Usually three passes is the most that is required to get a good segmentation.

13) BLACKART also contains several pattern recognition tools. Rather than filter the image according to pixel color, the pattern recognition tool attempts to take advantage of morphological attributes of groups of pixels to effect separations.

The block pattern algorithm removes pixels with a "blocky" i.e. non-linear structure.

The line pattern algorithm looks for pixel triplets with signatures characteristic of line structures.

The Nibble tool eliminates any pixel touching at least one white (0xffffff) pixel.

The Neighbor pattern searches for pixels with the specified number of neighbors with the same color. Setting the Filter Sense toggle will either set those pixels with more than the specified neighbors or less than the specified neighbors to the background color (0xffffff).

The Web Remover tool is used to remove unwanted pixels between contour lines. It is best used after the pixel classification tool (see above). Clicking on a pixel after enabling this tool removes all pixels of the same color (converts them to the 0xffffff) using a chaining process that is similar to that used to tag contour lines.

Select Image Processing|Pattern Recognition to use these utilities.

Note: The line recognition tool does not work very well at the present time. It will successfully recognize almost all pixels that are part of a single pixel width line of a given color. However, practical line structures are more complex, either being wider than one pixel or comprised of pixels of multiple pixel values. I am in the process of adding to the pattern library that the line recognizer uses and hopefully this will become a more useful tool soon.

14) Three edge detection tools have been provided: Sobel trailing edge, Sobel leading and trailing edge, and a Laplacian leading and trailing edge. An edge detector will seek areas of high pixel color gradient by repetitive application of a convolution mask. This has the effect of mapping the areas of highest gradient (edges) to 0xffffff and areas of low gradient (solid areas) to 0x000000.

The Sobel filters apply two 3X3 convolution masks, while the Laplacian applies a single 5X5 mask. As a result, the Sobels are better suited to images with sharp edge delineations while the Laplacian, which can span more pixels in each direction is more sensitive and can detect more subtle edges. However, because it is more sensitive, it is more susceptible to noise.

Select Image Processing|Edge Detect to apply edge filters.

15) After filtering the image, binarize so that it consists of black contour lines on a white background. Use the binarization menu selection for this purpose.

16) A tool has been provided to create a negative image. This may be useful in creating image masks. Select Image Processing|Negative Image in order to use this tool.

17) An image stretch tool has been provided. Image stretching means mapping the RGB color values of all of the pixels according to some rule to a new set of values. The purpose is generally to highlight certain features of an image that is otherwise more difficult to see. In the case of contour separation, the idea is to make hard to separate features more easily separable by shifting pixel values appropriately. In order to use the tool, select Image Processing|Stretch Array|Scale Factor. The algorithm will multiply the integer corresponding to the three byte RGB value by a scale factor that you will be prompted for.

Contrast stretch is also available. Basically, this adjusts the color distribution histogram so that its extents span the entire RGB color space.

18) Another tool allows the user to perform arithmetic operations on the image. In particular, the NAND operation compares two images, the process image and the reference image on a pixel by pixel basis. If two pixels are numerically equal, the pixel is set to the background color (for which you will be prompted by a dialog box.) This feature is useful for removing features by an image mask. In order to use the tool you must have the main (process) image and a reference image open. Select Image Processing|Array Operations|NAND Arrays. Note: both images must have the same dimensions in order to use this feature.

19) A despeckling tool has been provided to help clean the image after filtering. Select Image Processing|Despeckle in order to remove isolated pixels from the image. (Note: the despeckling tool will only work for binarized images. Select Image Processing|Binarize in order to enable the despeckling tool.

20) A digitizing tool has been provided for use in manually digitizing contours. In order to use the tool, first set the line color and marker colors using the Graphics|Tool Palette selection. You will want to choose both colors for easy filtering later, i.e. choose a marker color that you can filter out and a pen color that you do not want to filter out. Then select Image Processing|Digitize|Enable.

Each time you click on a contour, a marker will be dropped. When you are done digitizing the contour line, double click and a line will be drawn connecting all the markers. In order to close a loop, drop the final marker within ten pixels of the first marker. The digitizer tool will search the marker list and connect the last marker to the closest marker it finds. Markers have been designed so that they will not cut an adjacent contour.

21) It is important that the contour lines be thinned in order to produce an accurate interpolation. Select Image Processing|Thin from the graphics window menu. It may be necessary to apply the thinning algorithm twice in order to remove redundant pixels. Do not thin a zoomed image as this will produce gaps in the contour lines.

22) After thinning, it will probably be necessary to zoom in at least once. Remember, do not thin the image after zooming as this will probably produce gaps in the contour lines.

23) Speaking of gaps, BLACKART provides a tool for automatically seeking and closing gaps for *binarized* thinned images. Select Image Processing|Clean Image Processing|Connect Lines. A dialog box will prompt you for the pixel search radius. For example, if you select three pixels, the algorithm will attempt to close line ends with any line end within a three pixel radius from the end.

Two modes are available. The connector tool will either attempt to close gaps by searching for unconnected endpoints at radius=1 and then working outward to the specified search radius or else it will start at radius=search radius and work inward to radius=1. Both modes may be useful, depending on the nature of the gaps you are trying to heal.

Note: At present, the healing algorithm cannot distinguish between endpoints on the same line and endpoints on different lines. The algorithm is very primitive and does not take advantage of the directionality of the lines when seeking matches. This may cause the connection to be made coincidently with a short line segment. The type of break may have to be closed manually.

Note: The “seek nearest first” line closing algorithm is not the most efficient in the world and takes a little time to execute, especially for larger images with a lot of breaks. The computational time will increase geometrically as a function of the search radius. A progress bar on the main form will track the status of the computation.

24) Another image clean up feature is a tool that will remove short line segments from your image. Often a multitude of these short segments remain after extracting the contour lines from a raster image. Select Image Processing|Clean Image Processing|Delete Short Segments. You will be presented with a dialog box that will prompt you for the maximum allowable line length. The algorithm will remove all artifacts that are composed of less than this many pixels.

25) Use the Graphics|Draw and Graphics|Erase tools to correct any errors in the contour map. Fill in gaps with the drawing tool and erase spurs with the eraser tool. Note: it is not necessary that the lines be continuous in order to produce an accurate interpolation. However, it may be easier and less confusing to tag the lines this way.

26) Use the Graphics|Tool Palette menu selection to adjust the pen width for drawing and erasing.

IMPORTANT after editing the image, you must select Graphics|Copy Bitmap in order for the screen contents to be copied to the logical arrays. If this is not done, your editing will be LOST.

27) When you are ready to tag the lines, select Image Processing|Binarize to binarize your image.

IMPORTANT: you will not be able to tag your image successfully if you do not BINARIZE the image first!

After binarization, select Contour Interpolation|Tag|Set Elevation. This will load the Elevation Control Panel tool. This tool is not modal and will stay visible until manually closed. It has several features to make tagging lines easier.

You must first set the initial contour line elevation. IMPORTANT you must click on “Submit” in order to set the elevation. If you want the elevations to increment or decrement, enter the appropriate value (positive integer to increment, negative integer to decrement). Then check the auto increment box.

Select the “Tag Enabled” radio button to enable tagging. (This can alternatively be done from the main menu.).

Once enabled, tagging will stay enabled until a graphics tool or the pointer is selected or until the Contour Interpolation |Tag|Disabled menu selection is clicked or until the “Tag Disabled”.

Note: the tagging tool has a one pixel search radius to make it easier to select contour lines without zooming in, which slows down the tagging process because the image must be refreshed after each line is tagged.

28) Select a line. The line will turn color, a 'beep' will sound and the elevation value will appear on the diagnostic screen if the line is tagged successfully.

29) When you are done tagging all the lines with a given elevation value, select a new value by selecting Image Processing|Tag|Tag Lines. It is not necessary to re-select Contour Interpolation |Tag|Tag Lines each time.

30) If you make a mistake, you can re-tag a line by changing the elevation value and then re selecting the line. Or else you can select Edit|Undo.

31) If you wish to query a tagged line to determine its elevation value, select Contour Interpolation |Show Elevation|Enabled. The elevation value will appear on the diagnostic screen when you click on the line.

32) Often when tagging contour lines with the auto increment feature enabled it is necessary to go back to the previous elevation. This may be necessary if a line is broken and needs to be tagged with the same value as its neighbor, or if two lines are touching and need to be edited on the fly so that different elevations may be assigned. It is possible to reload the previous elevation by selecting "Reload Last" button on the Elevation Control Panel. The previous elevation will be displayed in the elevation text box. If you want to tag several contours with the same previous elevation, uncheck the "Auto Increment" check box.

Alternatively, if you see that a contour line is broken and you want to freeze the elevation at the current value while you tag all the pieces, move the cursor over the target and press the right mouse button. Repeated depressions of the right mouse button will tag additional contour lines at the current value. Pressing the left mouse button will resume tagging at the next (that is decremented) value.

When you are done correcting the errors, you may resume tagging at the elevation that you left off at by clicking on the "Resume" button. The last elevation value before pressing the "Reload Last" button will be loaded into the elevation text box. This will be the next elevation assigned.

33) It is also possible to enable the elevation query tool by selecting the appropriate radio button on the Elevation Control Panel.

34) If you are working on a large file and wish to discontinue work temporarily and then do more tagging later, you may save your work as an elevation file. Select File|Save Elevation Array. To load the file, select File|Open|Elevation from the main menu.

35) A single level of Undo is supported in this version.

36) A LineScan tool is available by clicking Image Processing|LineScan|Enabled. This tool queries the bitmap object for the pixel color. The Query Pixel Value tool queries the underlying (logical) 24-bit bitmap array.

Correcting SRTM and ASTER DEM Files

While 90m SRTM and ASTER DEM data have been welcome additions to the global DEM dataset, they both have certain limitations. Problems such as negative elevations and null data areas are often present in these DEMs. These are caused by clouds, snow, water, shadowing, phase unwrapping anomalies and other optical and radar-specific causes. The data can be significantly improved using BLACKART image processing and interpolation tools.

1) To remove noise (small negative and positive deviations from zero) from ocean areas select DEM Interpolation|Fix Sea-Level Elevations. The negative elevations greater than the lower sea level elevation and upper sea level elevation limits and positive will be replaced with zeros. The default values are –15 and +3. The values can be changed by selecting DEM Interpolation |Set Sea-Level Clipping Parameters.

2) If it is desired to interpolate the holes and if any single elevations within the hole are known, these postings can be manually added to zero elevation regions (if desired) prior to interpolation. Select Graphics|Set Elevation to set the elevation value. Then select Graphics|Draw Elevation|Enabled. Place the cursor over the location where you wish to insert an elevation and click the mouse. A "beep" will indicate successful placement of the elevation value. (Note: only a single elevation posting, i.e. a single pixel will be added to the image. The pixel will be small and may not be easy to see if the elevation value is low and matches the background zero elevation color closely.)

3) You may then interpolate the entire image. BLACKART will fill in the null elevations by interpolating. This is done without affective valid elevations. BLACKART first stores the target array in a temporary array. It then replaces the null elevations with zero elevations and interpolates. After the interpolation is finished, it scans the target array and replaces any null values it finds (for example –32,767) with interpolated values.

This will fill in the holes from the edges inward. (Unless of course other elevation values have been added manually as described n (2) above.) Although the holes will usually fill in evenly with a well-behaved surface, keep in mind that the nearest valid data to any interpolated point within the hole will be the nearest edge. The farther the point is from the edge, the greater the degree of approximation. Note: if you do not wish to enter elevation values manually, you can skip steps (1) and (2). In this case, BLACKART will clip the negative elevations automatically prior to interpolating.

Important Note: The default definition of null data is any elevation <-32000. This definition may be changed, and in fact must be changed in order to successfully interpolate some files for example ASTER files where null data may be defined as –150 elevation units. Select Image Processing|Array Operations|Set Null Data Definition to change this value.

4) Alternatively, you can use NIMA DTED0 data to fill in null data areas in SRTM files. DTED0 DEM files will always coincide exactly with a corresponding SRTM DEM file. Each will cover an area equal to 1 degree of arc of latitude and

longitude. Download a DTED0 file that has the same southwest coordinate as your SRTM file. Then load the image by selecting File|Open Merge DEM. When both files are open, select Image Processing|Array Operations|Get Merge Data. BLACKART will insert the elevation value from the DTED0 file that is the simple nearest neighbor to the null data posting location of your SRTM file.

5) Alternatively, you can use NIMA DTED0 data to fill in null data areas in SRTM files by first expanding the DTED0 file up to a sparse 1201 by 1201 elevation posting array; then interpolating to a 1201 by 1201 elevation surface, and finally substituting the interpolated DTED0 elevations for any null data in the SRTM file. This can be done by first opening the SRTM file, and then opening the matching DTED0 file. When BOTH files are open, select File|Input Data from the graphics menu. You will be presented with a dialog screen. Enter your desired Laplacian and LSQR iterations as before. Now select DEM Interpolation |Expand DTED0 DEM. BLACKART will expand the 121 by 121 DTED0 file to a sparse 1201 by 1201 file. A graphics window will show you the result of the expansion.

To interpolate, Select Run|Run Blackart as before. BLACKART will interpolate your expanded DTED0 file to a smooth elevation surface. You may inspect the result of the interpolation by selecting View|Display Image from the main menu. You can check the interpolated elevations by dragging the mouse cursor over the file and looking at the elevations reported at the bottom of the main window. You may also save this interpolated image in SRTM or any of the formats supported by BLACKART.

If the interpolated DEM is satisfactory, you can merge the DEMs by selecting Image Processing|Array Operations|Assign Interpolated Elevations BLACKART will inspect the SRTM file, replacing any null elevations with the corresponding elevation from the iterated DEM file.

6) Alternatively, you can interpolate just a portion of the target file instead of the whole file. This is done by selecting DEM Interpolation |Interpolate Subset Array. Draw a rubber band box around the area you wish to interpolate. Enter the desired iterations when prompted. Then select Run from the main menu. An interpolation as described in (3) above will be computed except that the computation will be done only on the specified subset. At the completion of the interpolation, only the subsetted area will be scanned and interpolated elevations substituted. This option has the advantage of running a lot faster than an interpolation of the entire file.

7) To remove non-zero elevation noise over water, you can try the low pass filter by selecting Image Processing|Filter|Low Pass Filter. However, this will probably not work as well as simply clipping the negative elevations.

8) BLACKART has the ability to interpolate files in batch mode. This allows you to interpolate a list of files overnight, for example. Currently, SRTM .hgt files,

index .ind files and South African index DEM .ort format can be handled in this mode. (Please contact me if you need other formats added to this option.) To use this option, select File|Batch Mode Interpolat|SRTM to SRTM. You will be prompted to enter the iteration parameters and the null data definition. Click the 'Continue' button on the dialog box to add the iterations and null data definition. Then select 'Run|Run Blackart' from the main menu to start the interpolations. Note: the same interpolation parameters and null data definition will be used for all files. Each file will be saved with the input file name except with the last character (the 't' in the .hgt extension) replaced with a zero. To use the file, rename it replacing the zero with a 't'. This is obviously to prevent over writing your data file.

9) BLACKART version 3.81 and higher features greatly improved ability to handle DTED0 and SRTM30 data. DTED0 data is available from the NIMA website and offers near global legacy DEM coverage. SRTM30 is a near-global digital elevation model (DEM) comprising a combination of data from the Shuttle Radar Topography Mission and the U.S. Geological Survey's GTOPO30 data set. It can be considered to be either an SRTM data set enhanced with GTOPO30, or as an upgrade to GTOPO30. SRTM30 data represents the best 30 Arc-second posting data currently available, surpassing DTED0 and GTOPO30 in data quality. This is the best of the three to use for patching. First, a comment about file naming conventions.

The SRTM-3 file name includes the latitude and longitude of the Southwest corner of the tile. DTED0 tiles are stored in folders that are named by their longitude coordinate. The folders then contain files that are named and listed sequentially by the latitude coordinate. DTED0 tiles are the same size (1 degree by 1 degree) as SRTM-3 tiles. This makes selecting and matching SRTM-3 and DTED0 pretty easy. Just choose the single tile in the longitude folder of interest with the same lower left coordinate as your SRTM-3 tile.

The SRTM30 tiles, on the other hand use a different naming convention. They are formatted and organized in a fashion that mimics the GTOPO30 convention. Unfortunately, this makes the file naming different from the SRTM standard and care must be taken in its use. The most salient differences are that the latitude and longitude coordinates are reversed as compared to SRTM-3 files, i.e. with longitude first and latitude second. The file name coordinate identifies the upper left (Northwest) corner of the tile, NOT the lower left (Southwest) as in SRTM tiles. The 27 tiles that individually cover 50 degrees of latitude and 40 degrees of longitude each have 6,000 rows and 4,800 columns.

Selecting the proper SRTM30 coordinate is a little trickier. In order to find the minimum latitude coordinate subtract (or add, if the NW coordinate is in the southern hemisphere and is considered negative) 50 degrees from the latitude. To get the other longitude coordinate, add (or subtract if east of the prime meridian) 40 degrees. It is necessary to map out the SRTM30 tile corners in order to make sure that your SRTM-3 tile lies within its boundaries.

Load BLACKART and open the SRTM-3 file first by selecting File|Open from the main menu. This tile will display in the graphics window. Let's patch with a DTED0 file first, because it is a little easier. Select File|Open DTED0 Merge DEM. Open the .dt0 file that has been previously down loaded from the NIMA site. This file will display in a small (121 by 121 pixel) graphic window when it loads.

BLACKART will do the same thing for both DTED0 and SRTM30 files. First, it expands the file to match the SRTM-3 file size. Then it interpolate it to compute the missing data. Finally, it substitutes the interpolated data for the null data in the target file. (Note: the target SRTM-3 file is not interpolated at all using this procedure.)

When both files are open, select File|Input Data from the graphics menu. You will be presented with a dialog screen. Enter your desired Laplacian and LSQR iterations as before. Now select DEM Interpolation|Expand DTED0 DEM. BLACKART will expand the 121 by 121 DTED0 file to a sparse 1201 by 1201 file. A graphics window will show you the result of the expansion. It will appear as a grid of dots that form the general outline of the source DTED0 file.

What has happened is that BLACKART has just spread out the data from the DTED0 file evenly across a 1201 by 1201 array. It does this by starting at the upper left corner of the array, and puts the first data point from the DTED0 file at coordinate [0,0] of the larger array. Instead of putting the next point at [0,1], it skips nine spaces and puts the next point at [0,10]. It proceeds in this fashion, skipping rows and columns until it creates a sparse 1201 by 1201 array using the data from the DTED0 file. In between it places zeros.

To interpolate the file, select Run|Run Blackart as before. BLACKART will interpolate your expanded DTED0 file (the "dot" file) to a smooth elevation surface. You may inspect the result of the interpolation by selecting View|Display Image from the main menu. You can check the interpolated elevations by dragging the mouse cursor over the file and looking at the elevations reported at the bottom of the main window. You may also save this interpolated image in SRTM or any of the formats supported by BLACKART.

If the interpolated DEM is satisfactory, you can merge the DEMs by selecting Image|Array Operations|Assign Interpolated Elevations BLACKART will inspect the SRTM file, replacing any null elevations with the corresponding elevation from the iterated DEM file. When this occurs, any null data values, which appear bright orange in the target file will generally turn blue or black. Of course the rest of the data values in the target file are left untouched by the substitution. You can check the results visually, or by querying using the mouse cursor.

To interpolate an SRTM30 file as the source map, first open the SRTM-3 file as before. Now open the SRTM-30 file. This is a large, 58MB file and it will take a moment to open. BLACKART will automatically find and extract the small 120 by 120 subset of data that it needs for the interpolation and then it will close the file and deallocate the internal array. When both files are open, select File|Input Data from the graphics menu. You will be presented with a dialog screen. Enter your desired Laplacian and LSQR iterations as before. Now select DEM Interpolation |Expand SRTM30 DEM. BLACKART will expand the 120 by 120 SRTM30 file to a sparse 1201 by 1201 file. A graphics window will show you the result of the expansion.

Now interpolate the file as before. Select Run|Run Blackart as before. BLACKART will interpolate your expanded SRTM30 file to a smooth elevation surface. You may inspect the result of the interpolation by selecting View|Display Image from the main menu. You can check the interpolated elevations by dragging the mouse cursor over the file and looking at the elevations reported at the bottom of the main window. You may also save this interpolated image in SRTM or any of the formats supported by BLACKART.

If the interpolated DEM is satisfactory, you can merge the DEMs by selecting Image|Array Operations|Assign Interpolated Elevations BLACKART will inspect the SRTM file, replacing any null elevations with the corresponding elevation from the interpolated SRTM30 file.

Second Technique - Coupled Interpolation and Substitution

The procedure outlined in the proceeding section has a problem related to the decoupled nature of the interpolative patch and the target image. In the procedure outlined above, the DTED0 data was interpolated and then patches were cut out of the interpolated surface and inserted into the target image wherever there were null data areas. There was no opportunity for information to flow from the target map to the interpolated map, and the result was abrupt and unsightly inconsistencies between the two.

A better approach might be as follows: expand the DTED0 image as before. Rather than interpolate this image, let's just cut the expanded grid and insert the sparse grid into the target map. Now make a copy of the target with the grid points inserted and interpolate it, subject to certain constraints on the known data points. This allows information from the DTED0 map to combine with information from the target map to produce a better interpolated surface. Now, cut sections from this improved interpolated surface and insert them back into the target map wherever null data areas exist. We would expect a much better edge match using this technique and in fact this appears to be the case.

In order to execute this method, load the SRTM-3 .hgt file as before. Then load the DTED0 file also as before. Next, expand the DTED0 file by selecting DEM Interpolation

| Expand DTED0 DEM. Now, insert the DTED0 data points into the target file by selecting DEM Interpolation | Insert Expanded DTED0 or SRTM30 into SRTM. If you look closely at the image, you should see an array of dots across the null data areas wherever the DTED0 file was not zero. Now select File|Input Data from the graphics window menu as before. Let's try 100 LSQR and 1000 Laplacian iterations. Select Run|Run Blackart from the main menu. After processing, BLACKART will update the null data areas with the computed patches.

General Notes on Solving the Interpolation Problem.

If you have prepared your file using the graphics preprocessor, you may either leave the graphics window open or close it. For large input files it is best to close this window as this will free up any storage resources not yet released so that the solver algorithms can have as much of these resources as possible. Large files run at a high enough number of iterations can use all available system resources.

Regardless of the method of input, you will be prompted to enter the number of iterations for the Saunders-Paige solver as well as the number of iterations for the central difference (Laplacian) solver that computes the initial estimate. This is done by selecting the File|Input Data from the graphics module menu.

BLACKART uses the Saunders-Paige iterative Conjugate Gradient solver as its principal computation kernel with a fast Laplacian central difference approximation solver to prepare the initial estimate for the CG solver. The central difference solver is about ten times faster than the Conjugate Gradient solver. This means that computation of a good initial estimate is relatively cheap. As a consequence, it is generally better to start with a larger number of Laplacian iterations and a smaller number of LSQR iterations. Try a 10:1 ratio for starters.

It is possible to run only LSQR iterations (although this is not recommended) or only Laplacian iterations. In the past, running only Laplacian iterations tended to produce inferior interpolated surfaces and was recommended only for zeroing in on the correct final interpolation parameters. However, recent improvements to the BLACKART Laplacian algorithm have made it almost as good as LSQR in many cases, so it pure Laplacian runs are now very much recommended because of their markedly reduced run time requirements and storage requirements.

(For more information on iterative solvers, see my Master's Project report on www.terrainmap.com. There is also an article describing the use of WinTopo with BLACKART
on the website.)

When you are done telling the solver how many iterations of each of the two solvers to run, simply select Run|Run Blackart from the main menu. The solvers will begin running. Since these computations can be quite lengthy, a progress bar and feedback to the diagnostics screen will keep you informed of the state of progress of the two solvers. Three audible beeps will sound at the conclusion of the computation.

Note: the grayscale image menu selection now works for SRTM files. You can also save your grayscale image by selecting File|Save SRTM Grayscale from the graphics menu.

Viewing the Result of the Interpolation.

When the computation is complete, you may either view your DEM by selecting the 'view' menu item or save your file or both. Output format is at present limited to USGS ASCII DEM, ASCII xyz, 16-bit flat binary (BSQ) or Terragen format. If you choose to save in USGS ASCII DEM format you will be prompted for georeferencing information. (Enter a negative latitude for the southwest corner coordinate of southern hemisphere locations.) You can also save the image file as a .bmp.

Note: 256MB minimum RAM is recommended for this application as it is very compute
intensive.

Modifying the Interpolated File.

BLACKART allows you to edit your interpolated file by re-importing the file in elevation (.elv) format after processing. To do this, first save your file as an elevation file after interpolation. The load the elevation file back into the program. Your interpolated image will be reloaded into BLACKART. You may then use any of the appropriate image processing tools to modify your image. At the end of the editing process, you may choose to run another interpolation.

Known Issues:

1) The output DEM file will have the correct latitude and longitude at the lower left corner only. No provision exists for precisely georeferencing the other three corners but they will be fine as long as your input map is oriented toward true north

2) When saving a file, do not attempt to over write an existing file as this produces unpredictable results.

3) Be advised that some applications (3DEM) will not accept USGS ASCII DEM unit codes other than meters(2). This is not a problem with the BLACKART DEM writer. Some applications may not comply fully with the USGS DEM file specification for a variety of reasons.

4) The shape file reader in BLACKART is not a general reader, i.e. do not expect it to handle all shape files. Shape files are capable of handling a wide variety of data types, most of which are not relevant to BLACKART. If you use ArcView or another application to generate a shape file for input to BLACKART, use only the type 3 polyline data type. Files of this type have been tested and appear to run OK.

5) There have been several complaints of BLACKART file read errors that have been traced to corrupt WinTopo shape files. WinTopo v2.5 is recommended to generate the shape files. It may also be necessary to become familiar enough with the shape file format so that some cursory file checking can be done, for example checking that all three files indicate the same number of data records. Because BLACKART has an internal front-end graphics processor, I am devoting my limited developmental efforts toward this end rather than incorporating extensive shape file error checking and diagnostics.

6) When assigning elevation values, avoid elevation 255 as this is a magic number for the image processing routines. Use 254 or 256 instead.

7) BLACKART v3.xx is still beta at this point. A fully stable release is not yet available as I am continuing development and have not completed extensive testing. However, I have run the program through several examples and have worked out the major bugs. I have also received a lot of helpful input from users that has both improved the program and allowed me to fix errors. Please let me know if you are experiencing difficulties and I will try to solve the problem and thus improve the program.

8) Iterative linear least squares solvers can have a variety of issues that can produce significant elevation errors. These are not a result of program bugs but rather are artifacts of the computational approach to the interpolation problem (i.e. the iterative approach as opposed to the direct solution approach). In general, large files require many thousands of iterations in order to allow the results of previous iterations to propagate across the entire file. This can take a while because information “flows” from one grid point to the next one iteration at a time. In addition, terrain features like cliffs falling into lakes will cause problems for any interpolative solver. The Franklin algorithm handles these better than most, but still requires many iterations to produce the expected result. See my Master's Project report for more information on the characteristics of iterative solvers, their advantages and disadvantages and the Paige-Saunders LSQR solver in particular.

9) BLACKART can currently (as of release 3.77) interpolate negative elevations accurately ONLY IF ZERO LSQR ITERATIONS ARE SPECIFIED. Interpolating a file with negative elevations using LSQR iterations will result in errors if there

are null data areas within the legitimate negative elevation areas.

This is due to a weakness in the multiplication algorithm that defines the LSQR computation and/or the supporting data structures. (This is my fault, not that of the basic algorithm since the multiplication algorithms are application-specific.) Fixing the multiplication algorithm so that the program works as intended will be time consuming because I do not fully understand the problem yet. I hope to have this accomplished by Summer, 2004.

In order to keep the program from crashing when processing a file with legitimate negative elevations when LSQR iterations are specified, ALL of the negative elevations, including the null data (default setting less than -32000) are clipped to zero before processing. The legitimate elevations are ultimately restored to their original values while the interpolated null values are patched into the source file. These areas thought that they were surrounded by zero rather than negative elevations during the interpolation, so that patches appear as hills or mesas that top out at zero elevation within a "sea" of negative elevations.

Negative elevations can also be handled by incrementing the elevations by the most negative elevation, interpolating, and then decrementing the elevations by the same value after the interpolation.

10) BLACKART can open both 3 arc-second global SRTM files and 1 arc-second USA SRTM files. However, options for processing 1 arc-second files are limited at this point. The DTED0 merge option will not work. In addition, the extremely large internal arrays created by the 3601 by 3601 integer input file size will probably cause an out of memory error if you try to interpolate the entire file using LSQR iterations. You may be able to complete an interpolation if your computer has 1GB RAM but mine (with 256MB) crashed during an attempted interpolation.

There are two possible work-arounds. The best is to use subset interpolation. Since this only processes a subset of the main array, even large input files will process relatively quickly. In the case of sea-level anomalies, you can use the clipping options without memory issues.

In other cases, however, the missing data regions are two prevalent and too widely scattered for subset interpolation to be feasible. In this case it is possible to effect an interpolation using zero LSQR interpolations. Since the memory requirements for Laplacian interpolation are so compact, it is unlikely that memory will be a problem even for large 1-arc second SRTM files.

11) SRTM30 tiles cover an area of 50 degrees latitude and 40 degrees longitude with a 6000 by 4800 elevation posting file. This means that each 1 degree by 1 degree subset spans exactly 120 by 120 elevation postings, unlike the DTED0

tile that cover 121 by 121 postings for the same area. Unfortunately, 120 rows by 120 columns do not map cleanly to 1201 rows by 1201 columns when expanding the array to match the SRTM-3 file size.

BLACKART handles this by expanding to 1200 rows by 1200 columns and then adding an extra row and column of all zeros so that everything matches. The zeros are subsequently interpolated so the zero row and column are blended out. However, there may be slight edge issues under certain (rare) conditions.

11) Several of the BLACKART graphics features do not work for DEM images. This is a result of the special way that some of the contour tagging images must be handled. I am in the process of rationalizing these algorithms so that they work in all cases.