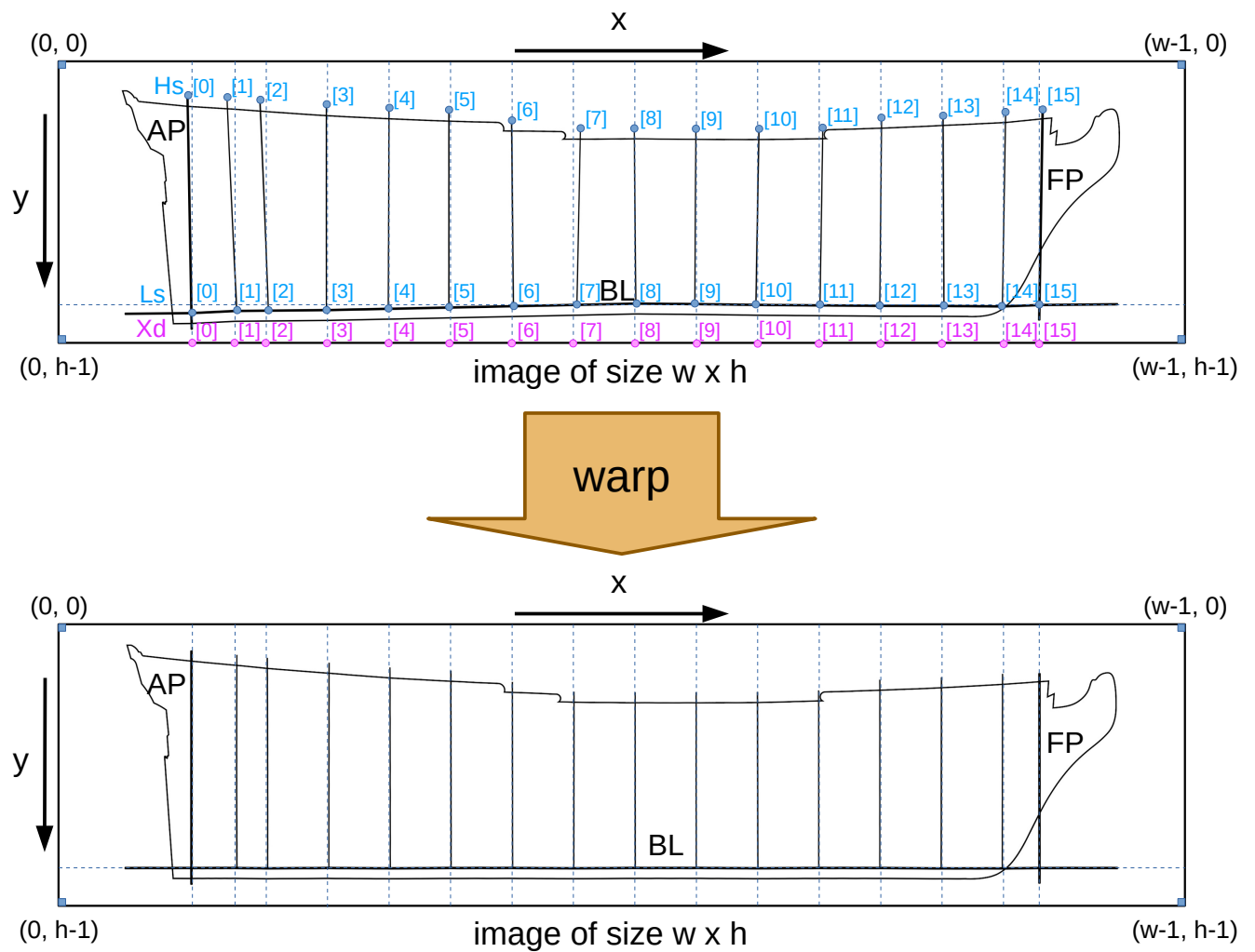


# Ship plan adjuster

## User Manual



# 1 Introduction

This manual is written to accompany the *ship-plan-adjuster.exe* program. It will give guidance on what is required by the program to adjust a ship plan, or more precisely a specific view of a ship plan, and make it suitable as a background image for use in 3D CAD programs.

With adjusting we actually mean in the first place aligning so that near horizontal lines become exact horizontal lines (e.g. base line at top of keel), and near vertical lines become exact vertical lines (e.g. perpendiculars and stations).

Apart from alignment the program can also make small corrections to distortion by stretching or compressing area's that do not comply with known rules like 'room and space'.

My main focus is related to ship plans of British ships of the 18<sup>th</sup> century. The RMG (Royal Museums Greenwich) has an enormous collection of such plans.

A small amount, 3678 at the time of this writing, is available on the Wikimedia Commons website: [https://commons.wikimedia.org/wiki/Category:Ship\\_plans\\_of\\_the\\_Royal\\_Museums\\_Greenwich](https://commons.wikimedia.org/wiki/Category:Ship_plans_of_the_Royal_Museums_Greenwich)

It must be stressed that this program has only limited capabilities and will not remedy all distortion, flaws or artifacts that can be present in a ship plan.

Always be critical on the transformed output.

## 2 Why did I make this program

I started years ago to use sheer, body and half-breadth plan as background images in FreeCAD, a 3D parametric modeler. Assuming that the advanced technology concepts used in such a modeler would make it easy to build a 3D model out of the different views was a deception. On a paper plan with the 3 characteristic projections of the ship's lines, each point of your model will be represented and positioned in 2D in each projection. That gives you a total of 6 coordinate values (2x X, 2x Y and 2x Z) while you only need 3 values for your model. It is almost impossible to predict which of the 6 possible values gives the best result.

So I started using GIMP, **GNU Image Manipulation Program**, to do some basic transformations (rotation, shear, scaling...) on the background images to get them more aligned in the hope that the 6 possible values finally combine into 3 values. Well, GIMP is a fantastic tool, but I had trouble to get the images pixel perfect.

To do the transformations with the ability to specify the exact mapping of points in the source image to points in the destination image was a challenge that I would explore with a Python program. The first test was very hopeful and so I continued programming until I could make all the wanted transformations.

## 3 Using the program

I wrote the program in the Python language because there exists a reliable and stable library (named Pillow) for image processing and one for numerical computation (named NumPy). To avoid that you, as a user, have to install the Python language and the required packages, I converted the Python program *ship-plan-adjuster.py* into an executable *ship-plan-adjuster.exe*.

That way you can just put it in a folder of your choice and by a double click on it, make it active and start a transformation. There is no GUI (Graphical User Interface) for the program, so you need to provide all required info for the transformation you want it to make in a text file that must be present in the same folder and have the same name (e.g. *ship-plan-adjuster.exe* will search for *ship-plan-adjuster.txt* at its start).

The image file to transform can be in the same folder as the program, but also in a subfolder to avoid cluttering up the program folder.

An example *ship-plan-adjuster.txt* file with the image file in a subfolder:

```
HMS Essex\J2656-sheer.png
As=(1373, 5135)
Bs=(17094, 5225)
Cs=(17137, 1456)
Ds=(1357, 197)
```

Points with pixel coordinates are identified with a capital letter and a suffix: s for 'source' and d for 'destination'. Above are shown 4 source points that will instruct the program to perform a squaring action on the image file.

A more in depth description of the text file contents will be presented later in this manual after you have some knowledge of the different transformations offered by this program.

### 3.1 Supported transformations

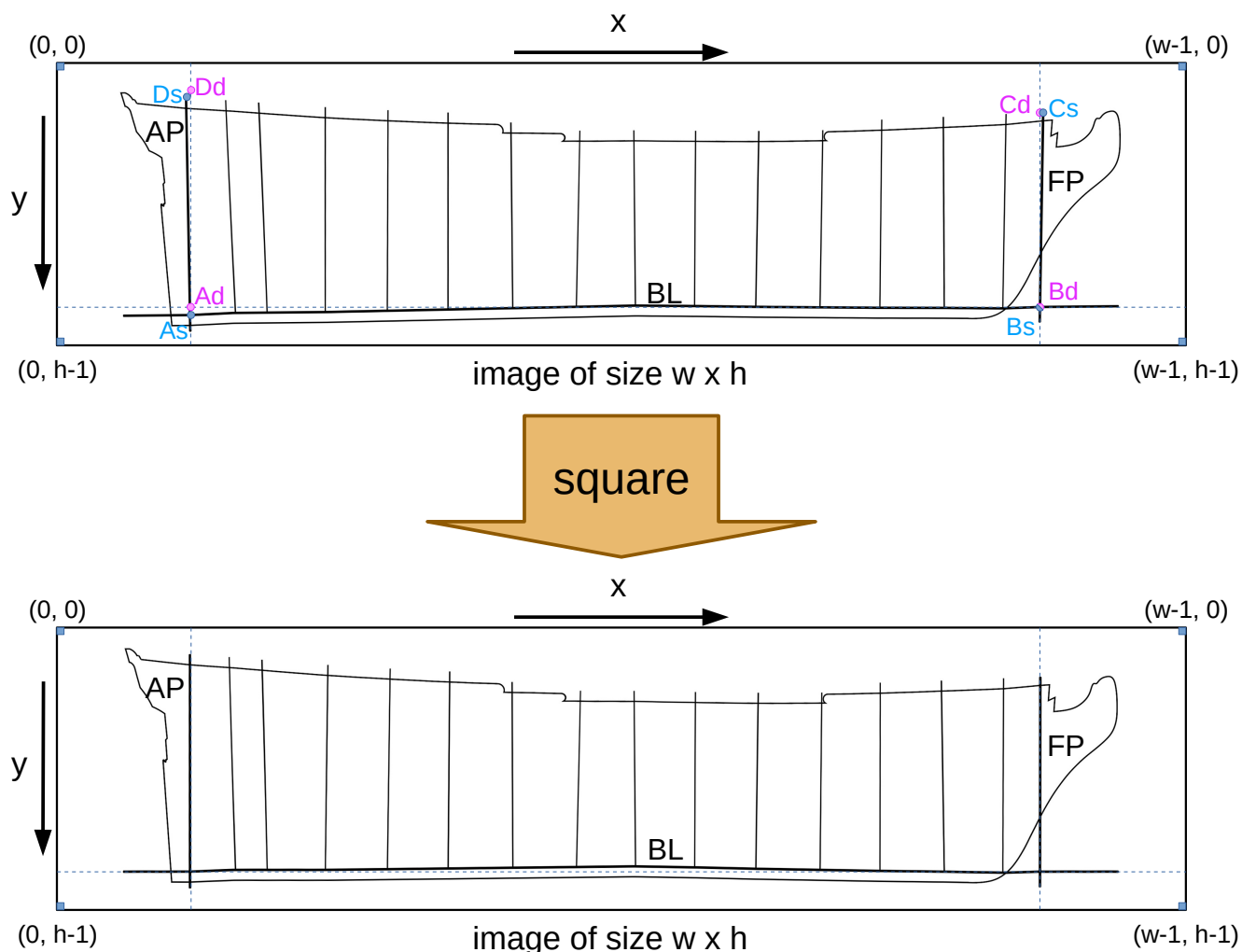
There are 4 different transformations you can apply to an image:

- ***squaring*** to make the base line horizontal and perpendiculars vertical
- ***flattening*** to level a curved or jagged base line into a horizontal line
- ***distribute*** to make stations equally divided (conform 'space and room' rule)
- ***warp*** to do most of the above in one combined action

Let's go over each type and describe what to expect based on an illustration with exaggerated misalignment and/or distortion.

### 3.1.1 Squaring the image

This is a perspective transformation aimed at setting the base line (BL) horizontal with reference to its intersection points with fore perpendicular (FP) and aft perpendicular (AP) and also making these perpendiculars vertical. The following (exaggerated) illustration makes it more clear.



To make the illustration shown above, I made a 3200 x 800 image file with the outline of the sheer view of the HMS Winchelsea 1764 but halved the number of stations shown to keep things clear. They are transformed with my program and then scaled into the illustration to show as realistic as possible what you can expect. The Winchelsea ship plan (ZAZ3093) is actually too good and was therefore severely deformed as you can see in the upper half of the illustration where I marked the 4 blue source points  $A_s$ ,  $B_s$ ,  $C_s$  and  $D_s$ . The other 4 fuchsia points  $A_d$ ,  $B_d$ ,  $C_d$  and  $D_d$  are optional and need only be given when you want to differ from the standard situation.

The standard situation is where the intersection point  $B_s$  of FP with BL will stay in place (so  $B_d$  will be identical to  $B_s$ ), all other points will be shifted but the distance between points  $A_s$  and  $B_s$  will be kept intact. For this standard situation you only need to provide the x and y coordinates of the 4

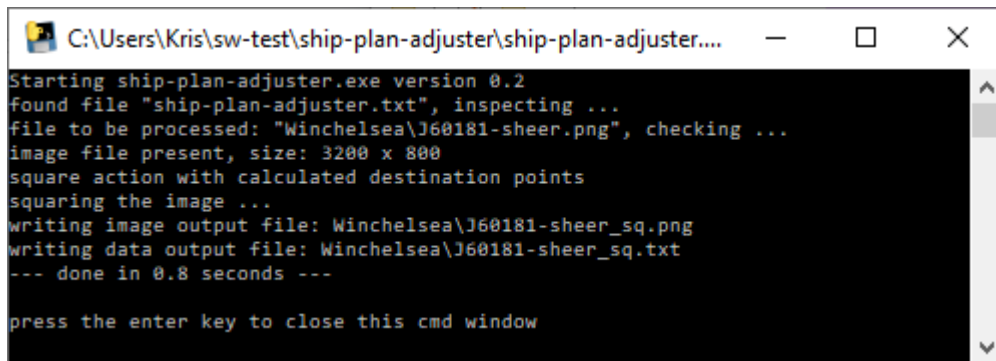
blue points. The program can calculate the position of Ad, Cd and Dd.

The transformed output image will have the same size as the input image and at the edges, where there are pixels that will not receive a transformed pixel value, a fill color is used that is extracted from the input image as the most dominant color.

The input file *ship-plan-adjuster.txt* that is used for the squaring shown in the illustration is:

```
Winchelsea\J60181-sheer.png
As=(379, 714)
Bs=(2787, 691)
Cs=(2795, 141)
Ds=(368, 100)
```

To start the transformation, double click on *ship-plan-adjuster.exe*. A cmd window will pop up showing some actions taking place. At the end it will write an image file with the transformed image and a text file with some details about the points used for the transformation for eventual later reference. They are both placed in the same folder where the input image file was found.



```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer.png", checking ...
image file present, size: 3200 x 800
square action with calculated destination points
squaring the image ...
writing image output file: Winchelsea\J60181-sheer_sq.png
writing data output file: Winchelsea\J60181-sheer_sq.txt
--- done in 0.8 seconds ---
press the enter key to close this cmd window
```

The transformed image now has A and B level but the lifting of A has caused more deviation from the base line in the midships section.

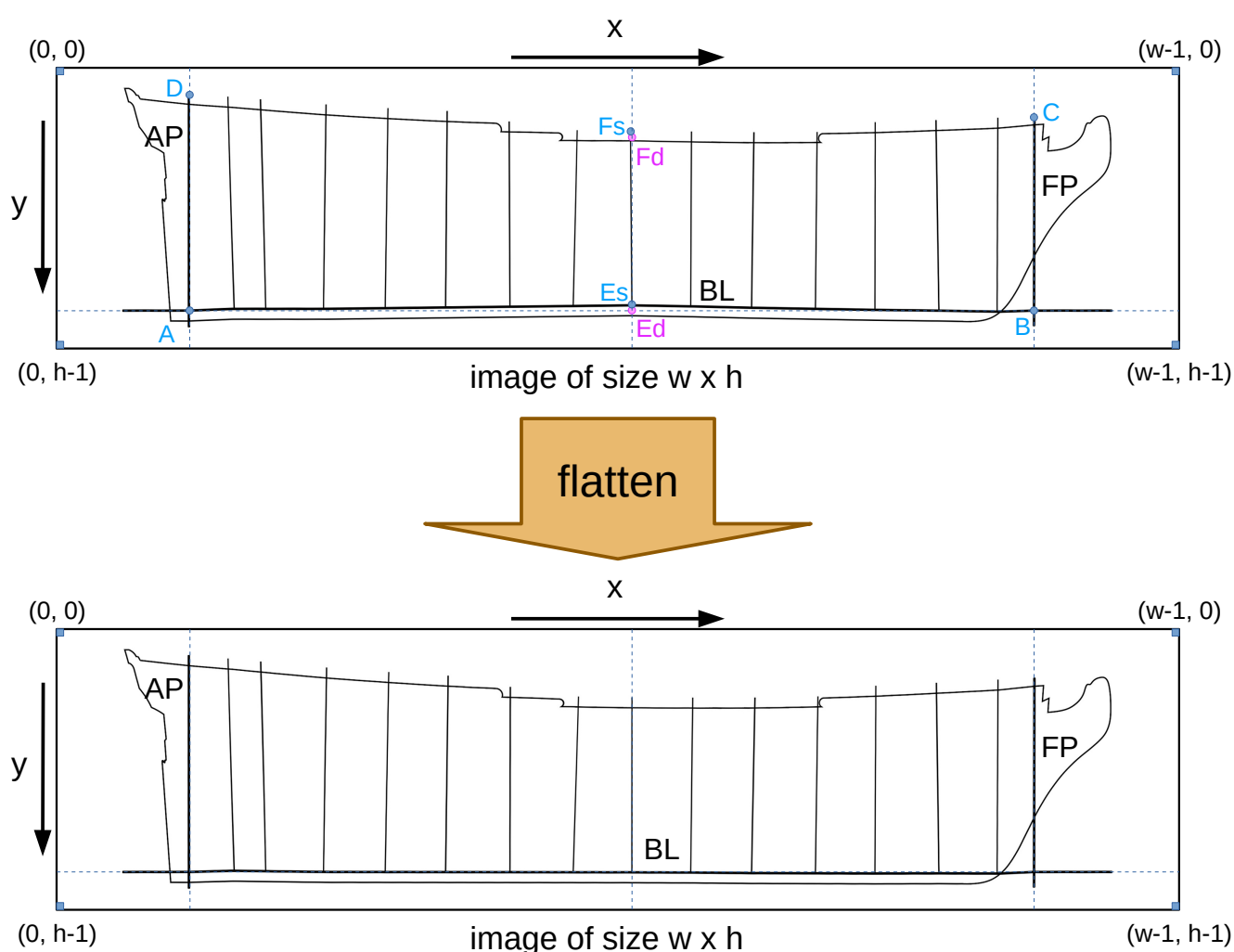
The content of the output data file *J60181-sheer\_sq.txt*:

```
Processed by ship-plan-adjuster.exe version 0.2
Winchelsea\J60181-sheer.png
As=(379, 714)
Bs=(2787, 691)
Cs=(2795, 141)
Ds=(368, 100)
calculated:
Ad=(379, 691)
Bd=(2787, 691)
Cd=(2787, 141)
Dd=(379, 77)
Winchelsea\J60181-sheer_sq.png
```

This file also shows the destination points Ad, Bd, Cd and Dd as calculated by the program. These can be consulted for later reference or as input for further transformations. Also keep in mind that you can specify these points yourself by including them in the *ship-plan-adjuster.txt* file. That way you overrule the build-in calculation that may occasionally result in transformed images that do not match your expectations. In that case the line 'calculated:' will not be present in the output data file.

### 3.1.2 Flattening the image

When the base line in the original image is not flat but has some curvature or even shows sections going up and others going down, then flattening can come to the rescue. The most simple case is shown in the following illustration.



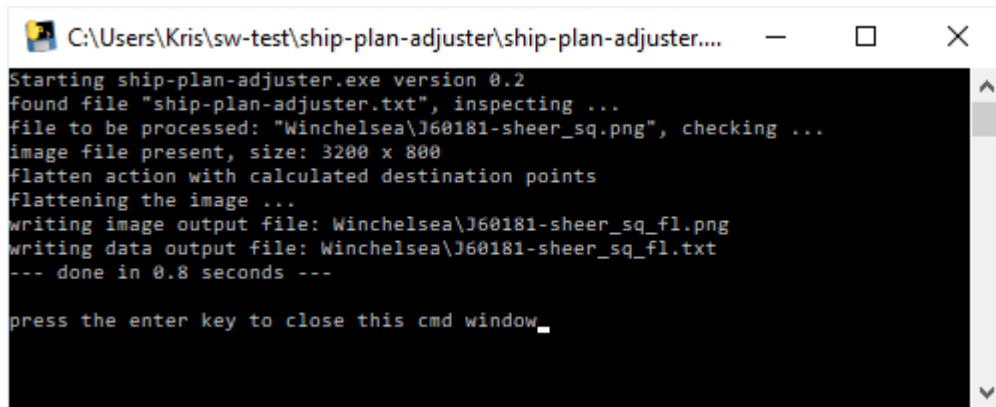
A flattening action is done when source points A, B, C, D, Es and Fs are present in the *ship-plan-adjuster.txt* file. Again it is possible to also provide destination points Ed and Fd, but, if omitted, the program will calculate these points and uses the y coordinate of Es for both Ed and Fd.

The contents of the *ship-plan-adjuster.txt* file for flattening of the Winchelsea illustration above:

```
Winchelsea\J60181-sheer_sq.png
```

```
A=(379, 691)
B=(2787, 691)
C=(2787, 141)
D=(379, 77)
Es=(1641, 675)
Fs=(1637, 180)
```

The CMD window output:



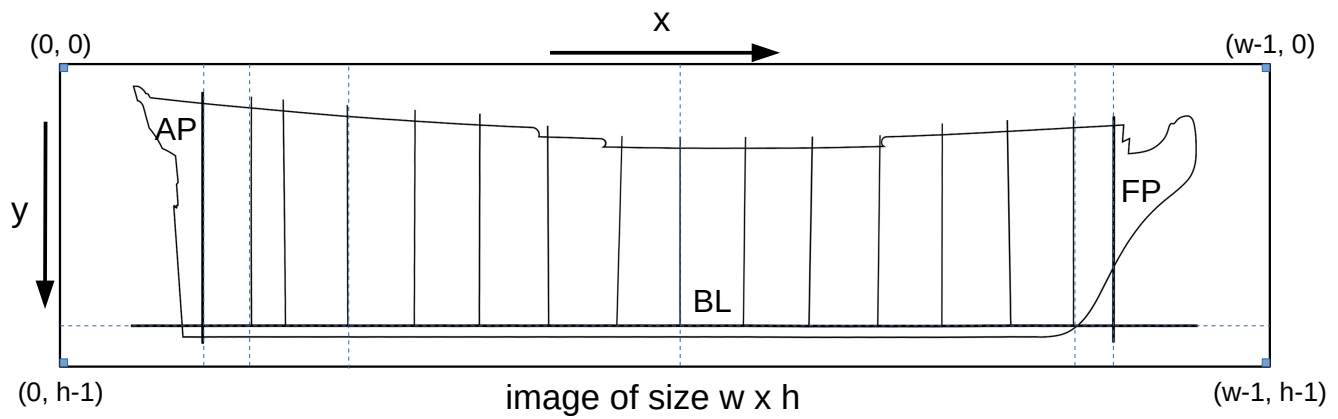
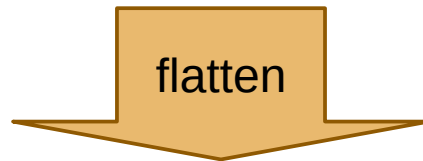
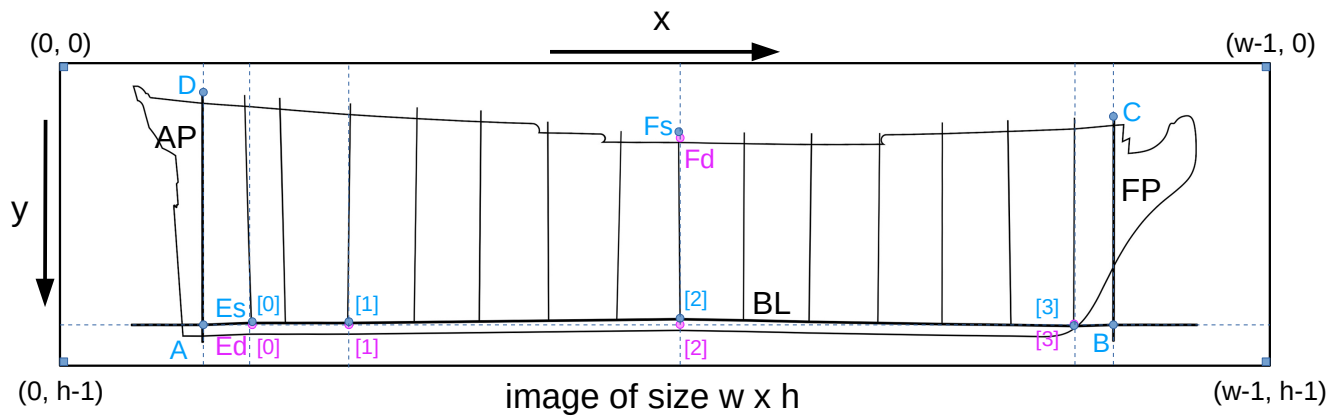
```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer_sq.png", checking ...
image file present, size: 3200 x 800
flatten action with calculated destination points
flattening the image ...
writing image output file: Winchelsea\J60181-sheer_sq_fl.png
writing data output file: Winchelsea\J60181-sheer_sq_fl.txt
--- done in 0.8 seconds ---

press the enter key to close this cmd window_
```

The output data file for this flattening action:

```
Processed by ship-plan-adjuster.exe version 0.2
Winchelsea\J60181-sheer_sq.png
A=(379, 691)
B=(2787, 691)
C=(2787, 141)
D=(379, 77)
Es=(1641, 675)
Fs=(1637, 180)
calculated:
Ed=(1641, 691)
Fd=(1641, 196)
Winchelsea\J60181-sheer_sq_fl.png
```

When the base line has more variation and you can identify rising, falling and horizontal sections, then it is possible to do a more accurate flattening by giving several points as Es and Fs. The next illustration shows this case with 4 points for Es and Fs. The points are represented by their index between square brackets. Index counting starts at 0.

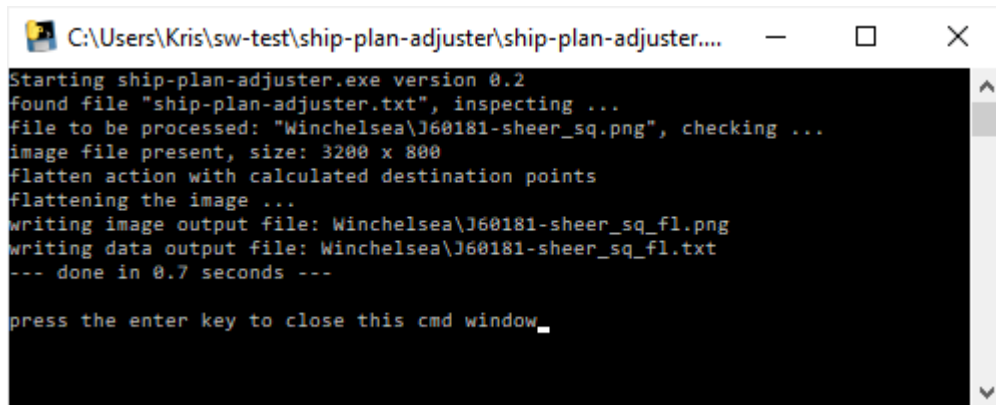


You have to enter the 4 points for Es and Fs between round brackets as you can see below.

```
Winchelsea\J60181-sheer_mt_pt_sq.png
A=(379, 691)
B=(2787, 691)
C=(2787, 141)
D=(379, 77)
Es=((508, 686), (762, 686), (1641, 676), (2681, 694))
Fs=((490, 85), (770, 107), (1637, 181), (2683, 145))
```



The CMD window output:



```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer_sq.png", checking ...
image file present, size: 3200 x 800
flatten action with calculated destination points
flattening the image ...
writing image output file: Winchelsea\J60181-sheer_sq_fl.png
writing data output file: Winchelsea\J60181-sheer_sq_fl.txt
--- done in 0.7 seconds ---

press the enter key to close this cmd window_
```

The data file output:

```
Processed by ship-plan-adjuster.exe version 0.2
Winchelsea\J60181-sheer_sq.png
A=(379, 691)
B=(2787, 691)
C=(2787, 141)
D=(379, 77)
Es=((508, 686), (762, 686), (1641, 676), (2681, 694))
Fs=((490, 85), (770, 107), (1637, 181), (2683, 145))
calculated:
Ed=((508, 691), (762, 691), (1641, 691), (2681, 691))
Fd=((508, 90), (762, 112), (1641, 196), (2681, 142))
Winchelsea\J60181-sheer_sq_fl.png
```

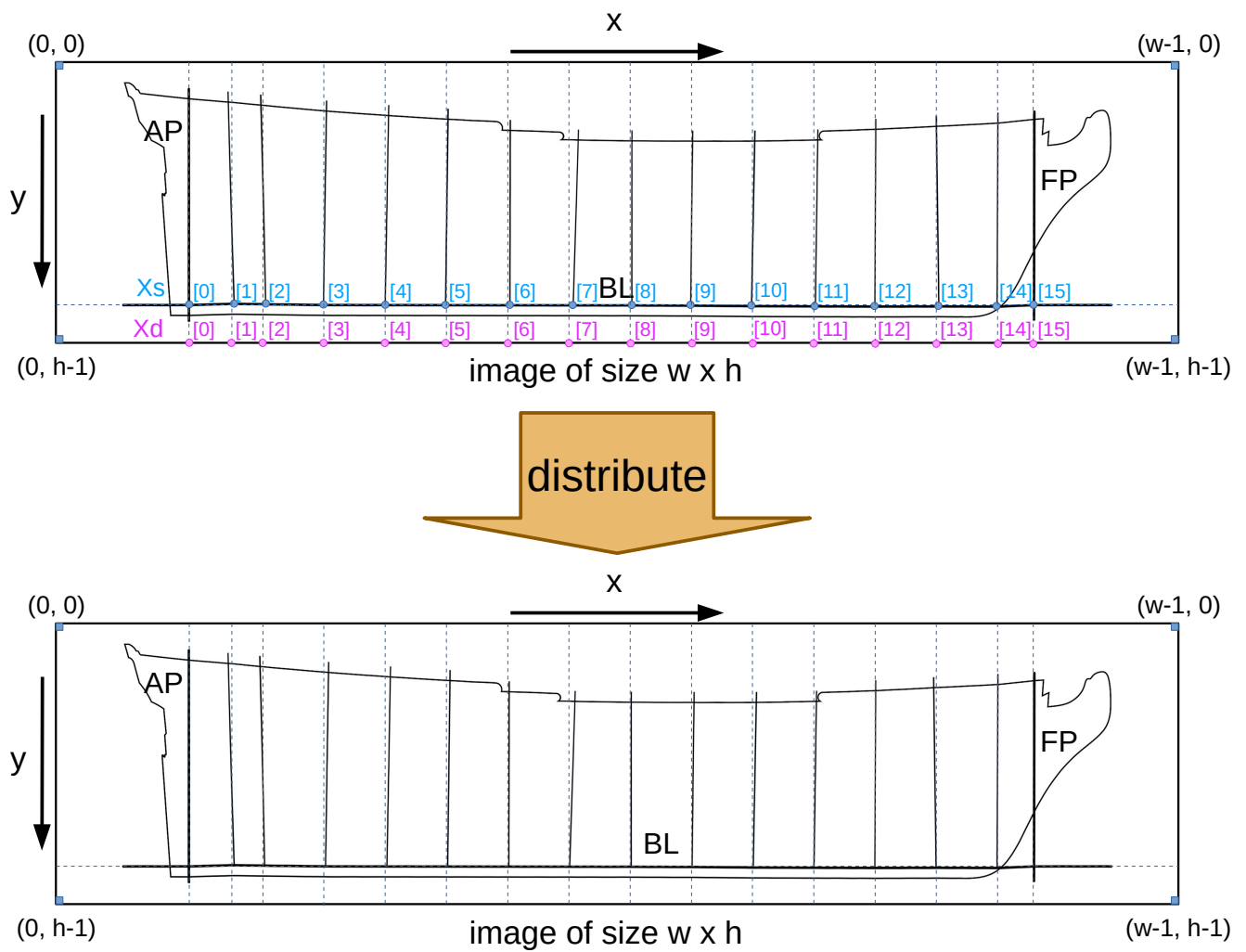
### 3.1.3 Distributing the stations

Once squaring and flattening are done there can still be minor variations in the distance between the stations. For Royal Navy ships of the 18<sup>th</sup> century there is the 'room and space' rule that can be used to do some small corrections to the station positions.

Between stations that represent full frames 2x 'room and space' should be respected. These should make the majority of the distances between station lines. In many cases there are 1 or 2 stations fore and/or aft that will have just 1x 'space and room'. The program will try to identify both spacings and calculate the average value in pixels for 'room and space'. Then it uses that value to shift station lines fore or aft to have them evenly distributed. The 2 stations closest to the perpendiculars will stay in place and keep their x coordinate.

This kind of transformation is called 'distribute', because it just stretches or compresses the distance between stations.

Its time to show this with an illustration.

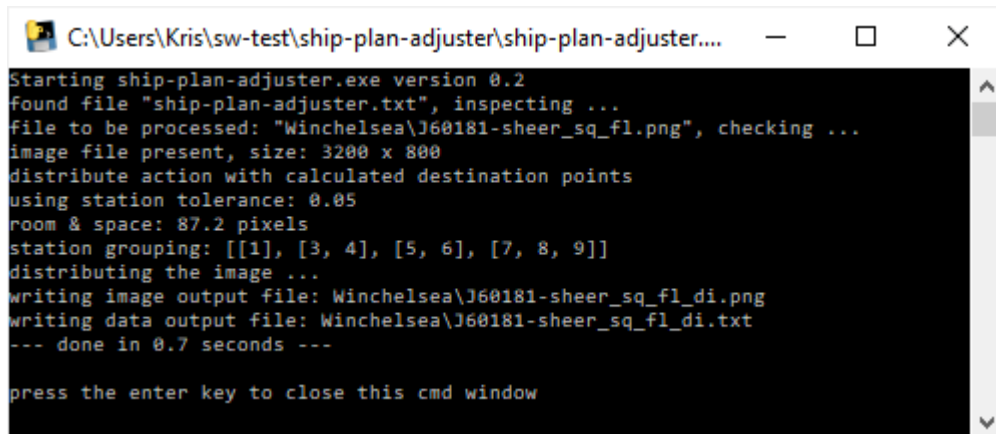


You have to edit the *ship-plan-adjuster.txt* file and add a series of x coordinates for all intersections of AP, stations and FP with the base line. Xs is used to identify this series of x coordinates, an example is shown below.

Winchelsea\J60181-sheer\_sq\_fl.png

Xs=(379, 508, 598, 763, 938, 1110, 1293, 1474, 1640, 1808, 1981, 2163, 2333, 2515, 2681, 2786)

The CMD window:



```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer_sq_fl.png", checking ...
image file present, size: 3200 x 800
distribute action with calculated destination points
using station tolerance: 0.05
room & space: 87.2 pixels
station grouping: [[1], [3, 4], [5, 6], [7, 8, 9]]
distributing the image ...
writing image output file: Winchelsea\J60181-sheer_sq_fl_di.png
writing data output file: Winchelsea\J60181-sheer_sq_fl_di.txt
--- done in 0.7 seconds ---

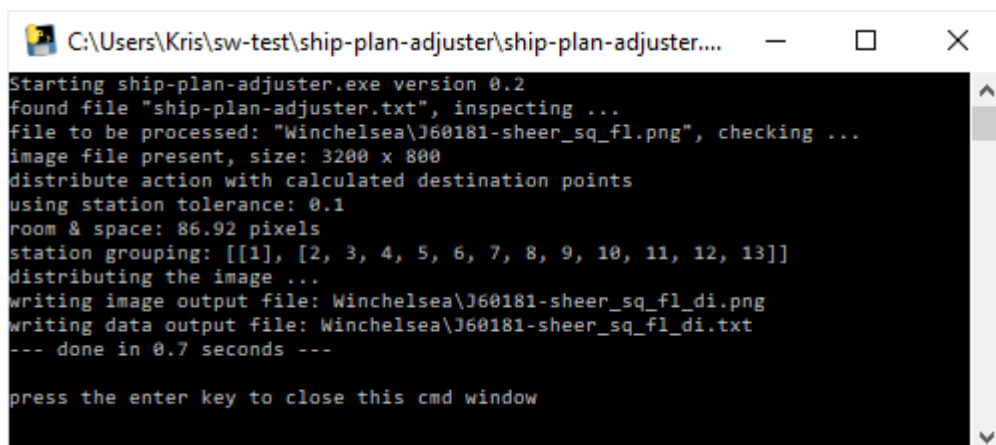
press the enter key to close this cmd window
```

Now here we face a problem, see the line: 'station grouping: [[1], [3, 4], [5, 6], [7, 8, 9]]'. There should only be 2 groups mentioned: the first one should be small with only the indices of distances according 1x 'space and room' and then a larger one with all indices conform 2x spacing. Because the program can not know the magnitude of deviation from the 'room and space' on your ship plan, it uses a tolerance to search for inter-station distances that belong to the 2x or 1x 'room and space' group. In the case shown above we let the program search with its default tolerance of 5% and that resulted in 4 groups. I clearly messed up the ship plan too much before starting the transformations.

To solve such issues, you can enter an extra line to the *ship-plan-adjuster.txt* file to set a specific tolerance to use: 'tol=0.07' what corresponds to a 7% tolerance. For the Winchelsea I had to increase to 'tol=0.1' (10%) before all station distances are correctly found and grouped. So the adapted *ship-plan-adjuster.txt* file becomes:

```
Winchelsea\J60181-sheer_sq_fl.png
tol=0.1
Xs=(379, 508, 598, 763, 938, 1110, 1293, 1474, 1640, 1808, 1981, 2163, 2333,
2515, 2681, 2786)
```

The new CMD window:



```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer_sq_fl.png", checking ...
image file present, size: 3200 x 800
distribute action with calculated destination points
using station tolerance: 0.1
room & space: 86.92 pixels
station grouping: [[1], [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]]
distributing the image ...
writing image output file: Winchelsea\J60181-sheer_sq_fl_di.png
writing data output file: Winchelsea\J60181-sheer_sq_fl_di.txt
--- done in 0.7 seconds ---

press the enter key to close this cmd window
```

Now we have just 2 groups. The first group has only 1 index: 1. That is the distance between the 2 leftmost stations. Index counting starts at 0 and runs from left to right, so index 0 corresponds to the distance between AP and most aft station. In our example case index 14 is for the distance between the rightmost station and FP. All other indices from 2 to 13 are now present in the second group. This is conform with what can be seen in the illustration.

The output data file:

```
Processed by ship-plan-adjuster.exe version 0.2
Winchelsea\J60181-sheer_sq_fl.png
Xs=(379, 508, 598, 763, 938, 1110, 1293, 1474, 1640, 1808, 1981, 2163, 2333,
2515, 2681, 2786)
using station tolerance: 0.1
room & space: 86.92 pixels
station grouping: [[1], [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]]
calculated:
Xd=(379, 508, 595, 769, 943, 1116, 1290, 1464, 1638, 1812, 1986, 2159, 2333,
2507, 2681, 2786)
Winchelsea\J60181-sheer_sq_fl_di.png
```

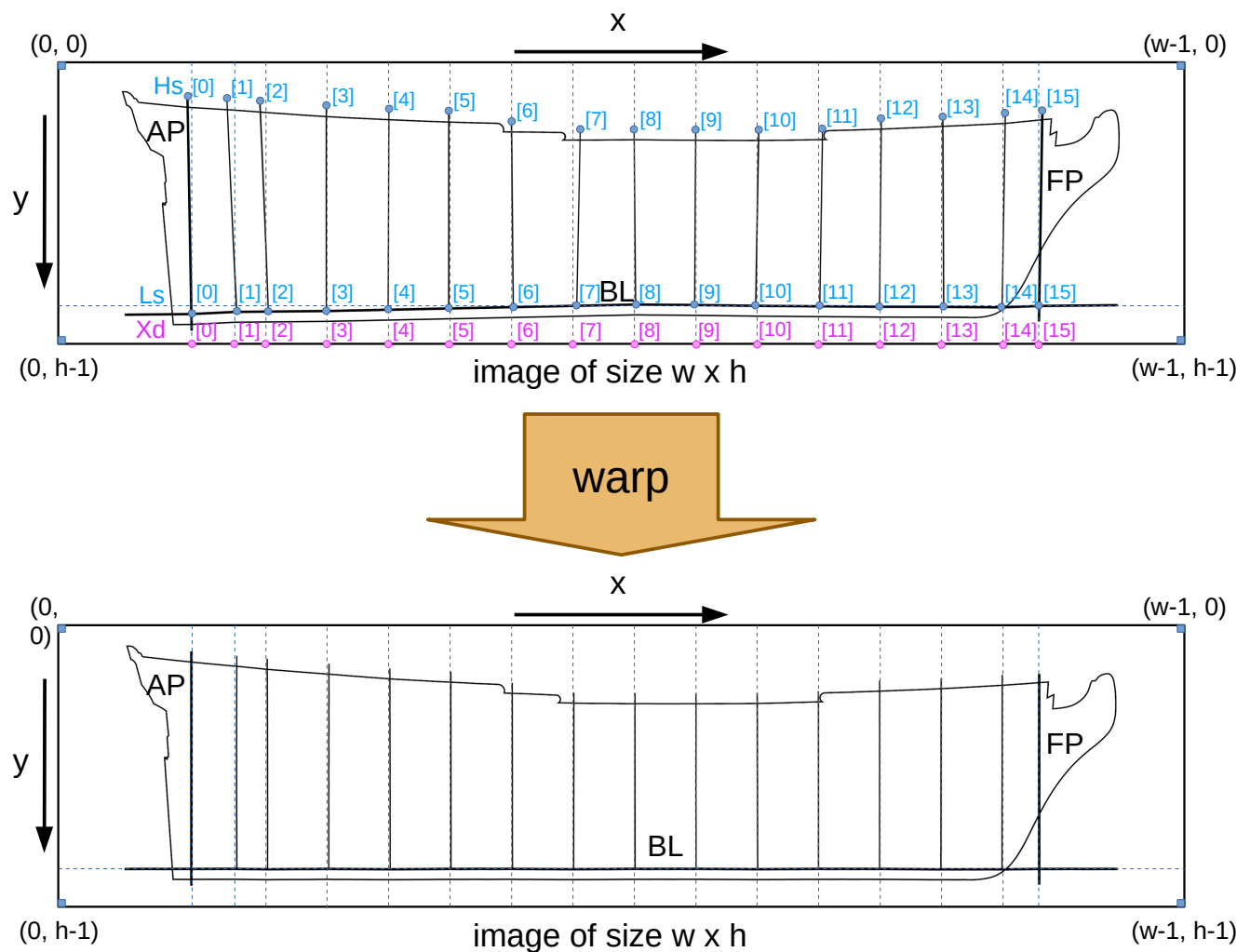
You can see that I added info for the used tolerance, the found pixel value for 'room and space' and the station grouping. So when in doubt about the resulting image you can check these values.

Please note that distributing will no correct for slanting of the stations, that should be handled by flattening or warping (see below).

### 3.1.4 Warping the image

This is a combined transformation that does squaring, flattening and distributing in one action. It is actually a sequence of a perspective transformation followed by a mesh transformation. The downside is that you have to give 2 series of pixel coordinates: one for the 'low' values, Ls, and one for the 'high' values, Hs. Ls are all the intersections of AP, stations and FP with the base line. Hs are the highest points of AP, stations and FP that you can observe on the plan. As always it is possible to add a series of x coordinates, Xd, to avoid the station distribution calculation by the program. In the case the program does a distribution calculation, check and eventually adapt the tolerance parameter.

Here is the illustration of an image warping.



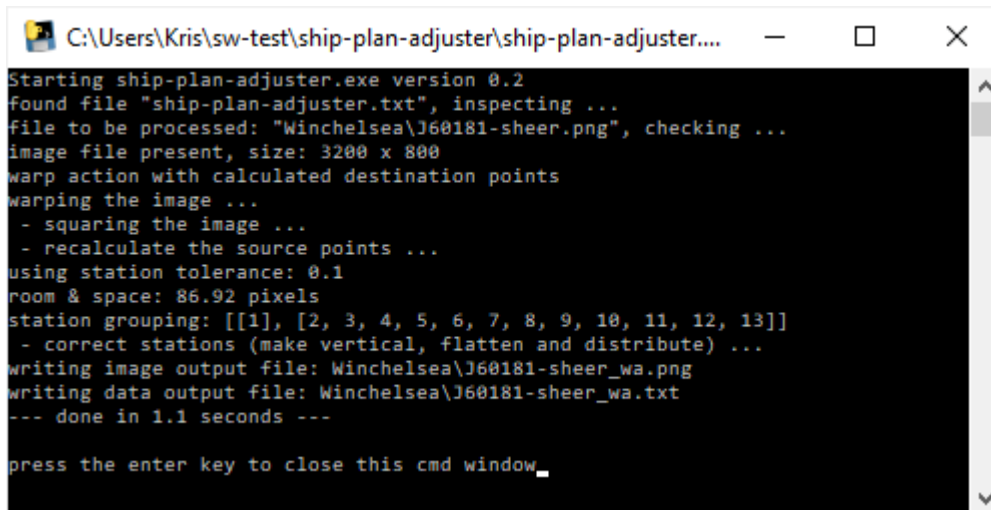
As you can see all stations are now vertical and we get a perfect horizontal base line.

This is the input provided for the warping:

```
Winchelsea\J60181-sheer.png
tol=0.1
Ls=((379, 714), (508, 708), (597, 706), (762, 705), (938, 701),
(1110, 698), (1293, 694), (1473, 690), (1640, 687), (1808, 688),
(1981, 690), (2163, 692), (2333, 693), (2515, 693), (2681, 694),
(2786, 691))
Hs=((368, 100), (480, 107), (574, 113), (762, 126), (940, 135),
(1110, 142), (1288, 170), (1483, 193), (1636, 192), (1811, 194),
(1990, 194), (2171, 192), (2338, 162), (2512, 154), (2690, 146),
(2795, 141))
```

Note that I already anticipated the station distribution issue and entered a 10% tolerance.

And this is the CMD window output:



```
C:\Users\Kris\sw-test\ship-plan-adjuster\ship-plan-adjuster....
Starting ship-plan-adjuster.exe version 0.2
found file "ship-plan-adjuster.txt", inspecting ...
file to be processed: "Winchelsea\J60181-sheer.png", checking ...
image file present, size: 3200 x 800
warp action with calculated destination points
warping the image ...
- squaring the image ...
- recalculate the source points ...
using station tolerance: 0.1
room & space: 86.92 pixels
station grouping: [[1], [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]]
- correct stations (make vertical, flatten and distribute) ...
writing image output file: Winchelsea\J60181-sheer_wa.png
writing data output file: Winchelsea\J60181-sheer_wa.txt
--- done in 1.1 seconds ---
press the enter key to close this cmd window_
```

The data output file gives all details about the transformation:

```
Processed by ship-plan-adjuster.exe version 0.2
Winchelsea\J60181-sheer.png
Ls=((379, 714), (508, 708), (597, 706), (762, 705), (938, 701),
(1110, 698), (1293, 694), (1473, 690), (1640, 687), (1808, 688),
(1981, 690), (2163, 692), (2333, 693), (2515, 693), (2681, 694),
(2786, 691))
Hs=((368, 100), (480, 107), (574, 113), (762, 126), (940, 135),
(1110, 142), (1288, 170), (1483, 193), (1636, 192), (1811, 194),
(1990, 194), (2171, 192), (2338, 162), (2512, 154), (2690, 146),
(2795, 141))
using station tolerance: 0.1
room & space: 86.92 pixels
station grouping: [[1], [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]]
calculated:
Xd=(379, 508, 595, 769, 943, 1116, 1290, 1464, 1638, 1812, 1986, 2159, 2333,
2507, 2681, 2786)
Winchelsea\J60181-sheer_wa.png
```

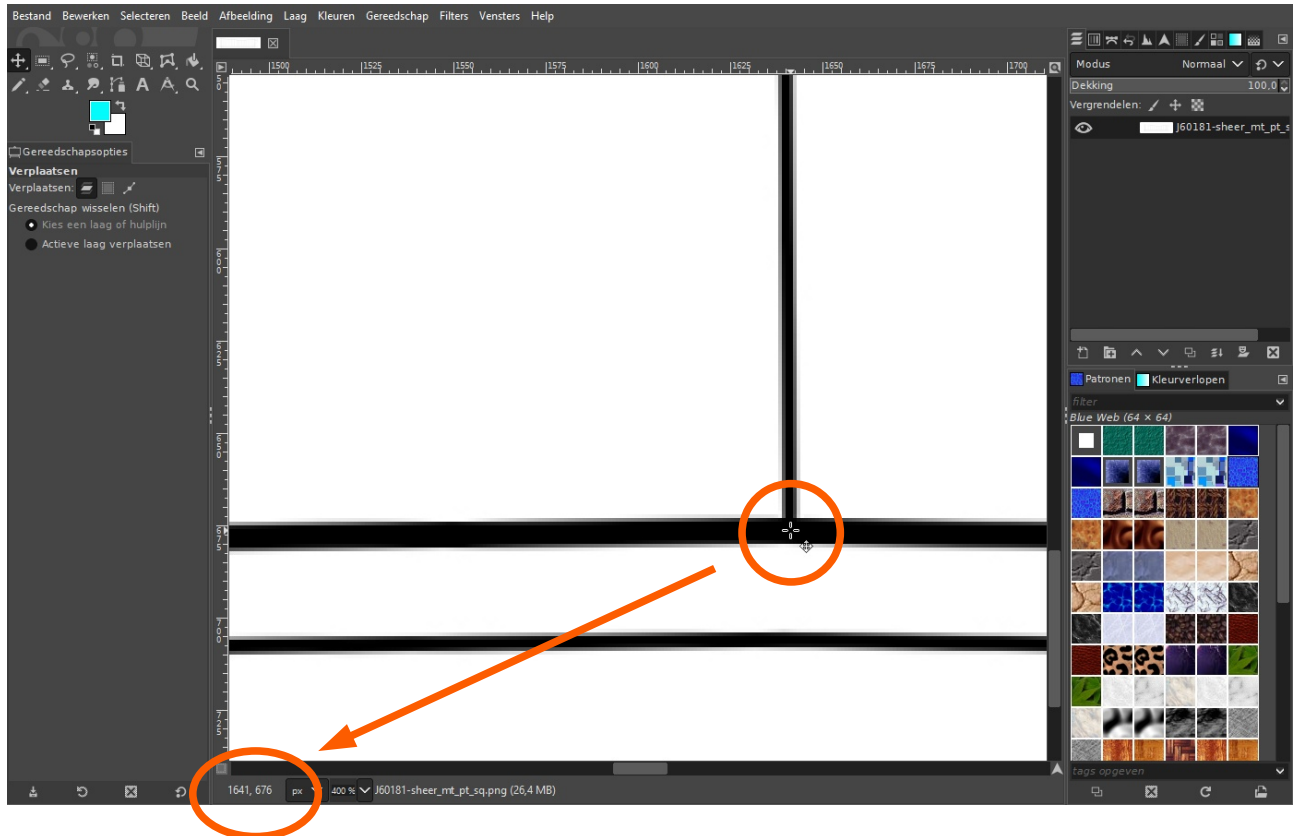
So far the explanation regarding the 4 different transformations offered by this program.

Always carefully check the output image if it matches with your expected modifications. I make use of GIMP to add horizontal and vertical guides that make it easy to verify the transformed image.

## 3.2 Providing the required coordinates

The program relies on the given coordinates to do the transformations. So you should try to be accurate and consistent when determining x and y of points in the source image. Again GIMP can come in handy. When you zoom in on the area of interest, e.g. an intersection of a station with the base line, and position the cursor at the point you need, the pixel coordinates are shown in the

status bar below the image. See next image to know the exact location.



Unfortunately there is no tool to extract these coordinate values and offer them in an easy way for copying. So I note the values on a piece of paper and then type them into the *ship-plan-adjuster.txt* file. That is time consuming but I know that the transform will be fast and pixel perfect with correct input.

I included several tests on the input values that you provide:

- every source point is checked to have coordinate values in the range of the input image
- a check is done to know that all required mandatory points are present before a transform is started
- when a point has multiple coordinates (cfr. Xs, Es, Fs, Ls and Hs) then all x coordinates should be strictly increasing
- when Es and Fs have multiple coordinates, then the number of coordinates given should be equal for Es and Fs
- Ls and Hs should always have an equal number of coordinates

In all cases where these checks show an inconsistency, a warning is reported in the CMD window. So watch this output before pressing the Enter key to close the window.

## 4 The *ship-plan-adjuster.txt* file

This file is crucial in telling the program what image file to open and instructing it what transform to make.

The next table gives an overview of point names to use and resulting transforms.

<i>Mandatory source points</i>	<i>Optional destination points</i>	<i>Resulting transform</i>	<i>Optional parameter</i>
As, Bs, Cs, Ds	Ad, Bd, Cd, Dd	squaring	
A, B, C, D, Es, Fs	Ed, Fd	flattening	
Xs	Xd	distributing	tol
Ls, Hs	Xd	warping	tol

A typical workflow is starting with a squaring and then use the AD, Bd, Cd and Dd coordinates from the output text file as input values for A, B, C and D for a flattening. If desired a distribution can follow.

When you want to set all stations perfectly vertical, then warping is your best option.

## 5 Conclusion

For me the development of this program was challenging but also fun when seeing good results appear. I have not yet used transformed images in 3D CAD programs so I can not confirm on the expected benefit.

During development and testing I have seen strange output images. In most cases that was caused by typo's in the input coordinates. That should now be minimized by the added checking on input values.

It would be nice if you have good results when using this program, to hear that on the Model Ship World forum.

– end of document --