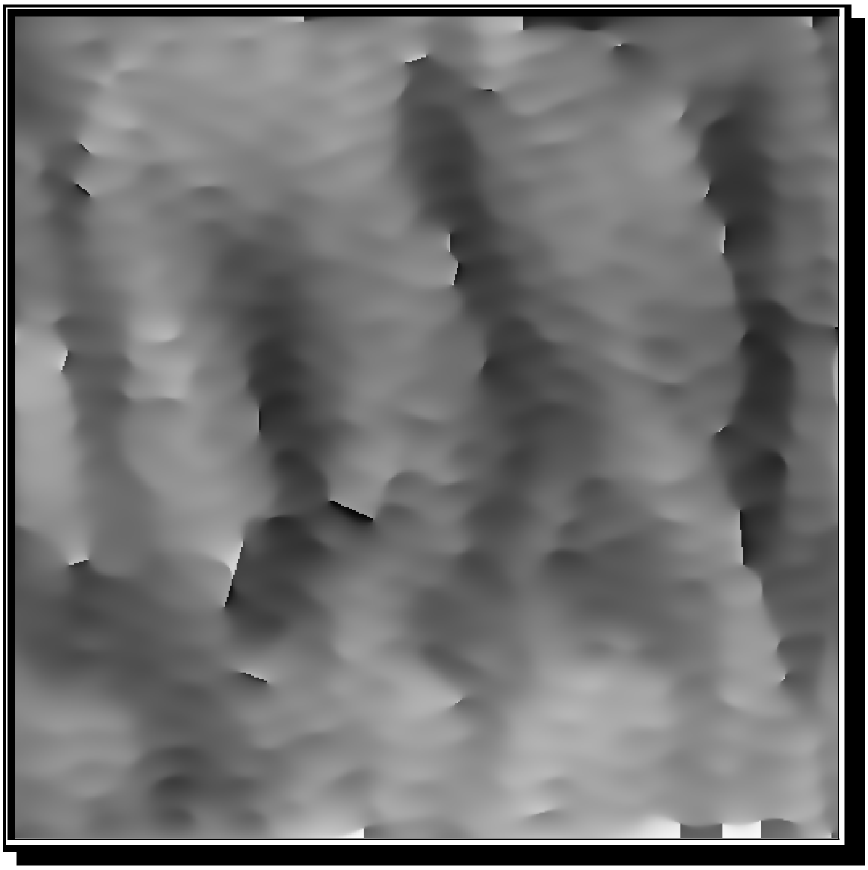
29 July 1998

By:
Erik Elfgren, elf@ludd.luth.se
Gnejsstigen 83
SE-977 53 Lule
Sweden

At:
Cavendish Laboratory, PCS group
University of Cambridge
Cambridge CB3 0HE
U.K.

MOIRÉ PROFILOMETRY

SUPERVISOR: TIM GOLDREIN



"Except where specific reference is made to the works of others, this work is original and has not already been submitted either wholly or in part to satisfy any degree requirement at this or any other university."

Signed

Erik Elfgren

Summary

Surfaces can be measured with moiré profilometry and the information can be stored digitally. This report presents experimentally and analytical improvements to an earlier work of Lang (1998).

The main specimen of investigation has been a cast of an ancient Greek carved tablet, but a darker, two-coloured piece of plasticine has also been studied.

To achieve a good result, the moiré fringes have to be easy to see. There should be a good contrast between the moiré fringes and the surrounding area, and they must stay clear of each other. If this is not fulfilled, the result will contain discontinuities and be of bad quality.

A coarser grating generally produces a better contrast between the moiré fringes and the surrounding area. For a dark specimen, the contrast naturally is worse than for a light specimen and therefore a coarser grating is needed. However, a coarser grating has the disadvantage of giving worse resolution.

When the specimen is multicoloured, this needs to be compensated for. Taking a complementary image, i.e. an image without the grating, and dividing the moiré fringe image does this. However, more light hits the surface when the grating is removed so the brightness of the complementary image must be lower than for the moiré fringe image. How much lower can be determined by comparing the colour in the area between the fringes with the same area on the complementary image. This can almost entirely compensate for the different colours.

Table of contents

1 Introduction

Measuring surfaces and storing the information digitally has several applications that are described in "Moiré Profilometry" by Richard Lang (1998). The theory for the method is described in the report, as well as some explanation of concepts. The following is a quote of the abstract of Langs report:

> *"An experimental system for the generation and analysis of surface contour fringes using moiré interferometry is presented. The system utilises the shadow moiré technique to generate fringes and a CCD camera to capture images directly to a computer. A custom fringe analysis program (FRAN) is used to reduce the images to phase maps using Fourier methods and a phase unwrapping algorithm. These phase maps are further processed and combined using specifically written software to produce height maps of an entire object. Test objects have been profiled to verify the system, followed by the profiling of several real-world objects."*

The present report gives the results of a continuation project and does mainly concern the improvement of the method, both experimentally and analytically. The main specimen of interest has been a cast of an ancient carved Greek tablet. Some of the problems have been solved, and some have only been described. A couple of software tools have been developed for making the process more automated, but the image grabbing still requires some experience. The method seems very promising and it should be able to produce even better results with some further research. This report assumes that the reader already is familiar with several of the concepts presented in Lang (1998).

The measurements referred to in this report are done qualitatively, not quantitatively. The primary interest has been to study what a good moiré image would look like, and how to process these to produce a clear result with few and small discontinuities. There should be no greater difficulty in obtaining qualitative information as well.

2 Experimental work

2.1 Image grabber

It is possible to change a couple of parameters to get a better image. It is usually a good start to have Contrast and Red intensity on max and then vary Gamma and Brightness. Tim Goldrein (1998) says that Gamma is a linearity coefficient which should not be fiddled with, and it might be worth investigating what would happen if you only changed Brightness. It is always good to have a high Contrast since the moiré fringes (which should be nearly black) shall be as clearly visible as possible to get a good result. The images are taken with a resolution of 768x576 (but later cropped to 512x512). The use of Red intensity will probably depend on how red the specimen is, but this has not been studied. Experience has shown that Gamma is more potent than Brightness, i.e. if Gamma is increased by ten this looks brighter than if Brightness is increased by ten.

2.2 Quality

The quality of the image is good when there is a high contrast between the moiré fringes and the rest if the image. However, it is not enough with a good contrast; the moiré fringes also have to be separated and clearly distinguishable from each other. Apart from good quality of the image, a good resolution is also desirable.

A coarser grating gives a better contrast and less blurring of the moiré fringes. The problem is that the resolution is proportional to the resolution according to Lang (1998), which results in a worse resolution with a coarser grating.

A high contrast is obtained by placing the camera so that it has its focus on the moiré fringes. This distance depends on the grating that is used, i.e. how many lines per inch (lpi) it has. For the Greek tablet a 30-lpi grating [Figure 1b, c] proved to be good. A 20-lpi grating [Figure 1a] did not give quite as good resolution, though the focus and contrast were excellent. The 40-lpi [Figure 1d] grating did not have enough contrast and therefore the resulting image contained many small discontinuities.

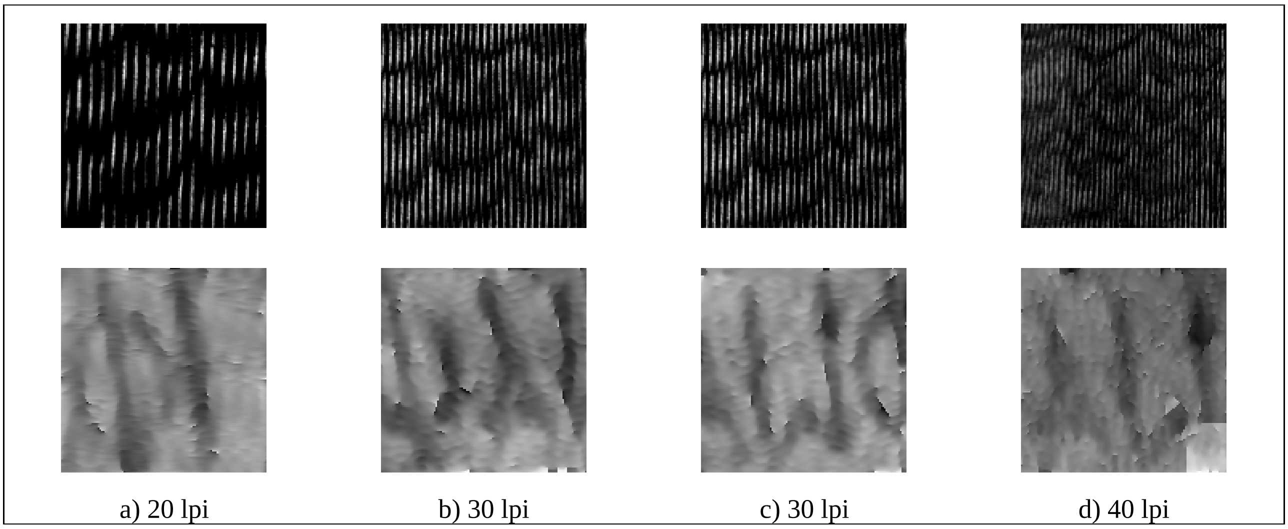


Figure 1: *Images of the tablet. First row are moiré fringe images, second row are the result after processing.*

The small discontinuities, 'knuckles', connected to the 40 lpi grating used with the tablet, were also present in the 30 lpi image but to a much lesser extent. The effect also shows on a plane surface, so it is not the natural bending of the surface that is causing it. There has been no proper investigation of the cause of these but it might be that the grating contains scratches, and that this becomes more prominent with a higher number of lpi. If the specimen had a dark surface, a coarser grating is needed in order to get a higher contrast, as the contrast gets better with a coarser grating. For a piece of plasticine (blue), a grating of 20 lpi was needed to achieve a satisfactory contrast. The method has not been tested for anything else than two coloured specimens and a study of multicoloured specimens with inscriptions should be done.

A possible improvement of the set-up is to shine more light on the specimen, or light that operates mainly in the red wavelengths, as the camera has best sensitivity there. By doing this more light will pass through the grating and hence the contrast to the dark moiré fringes will be higher. Another possibility is to put powder of some sort on the specimen to make it reflect better but this might cause some problems with fixed, vertical objects. It is an advantage if the rest of the room is dark when the pictures are taken. This gives a slightly better contrast.

It is also be possible to improve the grating, using one that is less spotted, or a grating which has a higher difference between the light and the dark areas. The spots seem to make more difference when it is darker. Hence, a better grating is required to study the latter. In order to get a good image the moiré fringes should be rather horizontal. The grating is vertical and with the horizontal moiré fringes, this gives the best visibility. To get horizontal moiré fringes the specimen should be slightly tilted towards the camera, and somewhat tilted away from the top of the grating. All my measurements have been made with a vertical (| | |) grating and horizontal moiré fringes. The tilt of the specimen creates some minor problems that can be solved by using a slightly different experimental set-up. The interesting area has to be rather close to the grating so that the camera can focus on both the grating and the moiré fringes. At the same time there has to be a certain tilt of the specimen so that the moiré fringes becomes vertical, i.e. perpendicular to the grating. To study an area that is rather high up on an object it must be possible to move it along its axis [Figure 2], in order to preserve the distance to the grating and the tilt of the specimen. The set-up in Figure 2 (and in the real set-up) would not be fit for this purpose, as the higher parts of the specimen has to be too far from grating due to the tilt. This puts a limit to how large objects that can be studied.

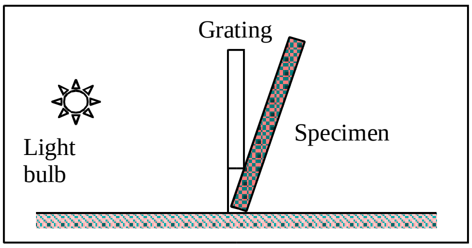


Figure 2: *Good images are achieved by tilting the specimen, but this makes the top part of it to come too far from the grating.*

It is possible to improve the resolution of the method by using a larger camera angle [Figure 3]. When the angle is increased, the moiré fringes become more closely packed, thereby giving a better resolution. There are two main problems about this; the first one is that the focus of the camera on covers a small depth, thereby making the actual area of view very small. The other problem is that as the moiré fringes gets more closely packed, they tend to blur into each other hence creating discontinuities in the phase map.

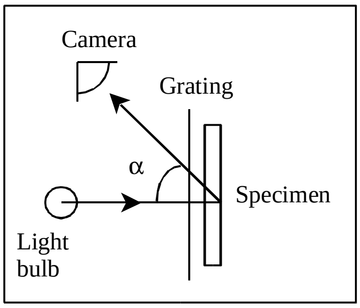


Figure 3: *Experimental set-up. A high angle* α *gives better resolution, but also worse focus.*

2.3 Camera

The camera is a crucial point in the method. With the eyes, it is possible to see the moiré fringes very clearly, even if they are densely packed [cf. 2.2]. Therefore, it should be possible to get very good images, if the right combination of aperture, lens and camera is used. It is probably worth to make some further studies with high angles to get a better insight in the problems and eventually find improvements. High angles are desirable as they give a better resolution. When the experimental set-up was changed from using an ordinary lens to using a bellows the results improved remarkably. The reason for this being that the resolution increased as the camera could be put closer to the specimen (shorter focus) and that the focus was better than before. The focus must be rather good on both the moiré fringes (i.e. the specimen) and the grating. If the grating is out of focus this will blur the whole image. As the camera is moved closer it is worth having in mind that the light might not be collimated anymore thereby possibly changing the relation between phase and height.

It should be possible to solve some of the aforementioned problems (like blurring fringes with bad contrast or bad resolution of dark specimens) by the use of a proper lens and aperture. This might be the key to the entire problem. A program (phase_height) transforms an image in phase scale to an image with mm scale. When trying to get quantitative information out of a phase map, the angle between the camera and the light source must be measured [cf. Figure 3], and fed to phase_height, thereby giving the real heights of the surface.

2.4 Multicoloured specimens

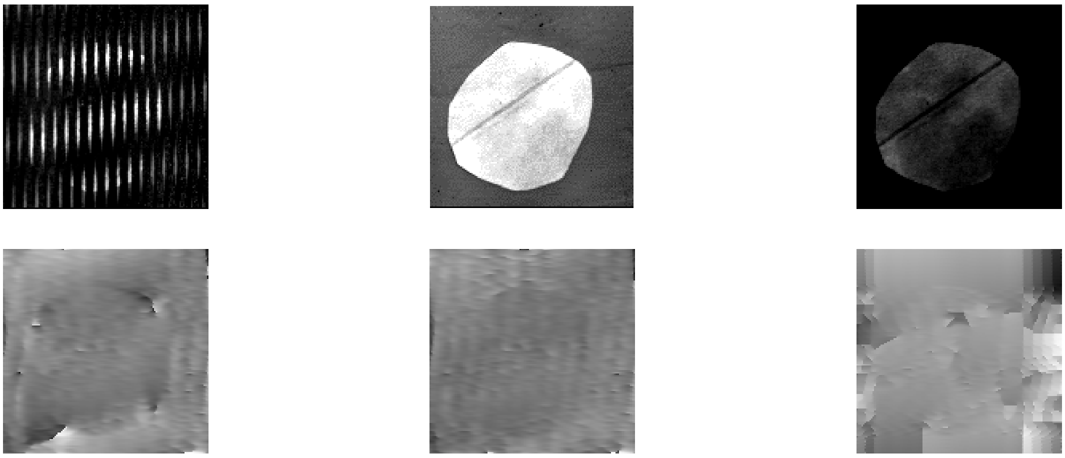
If you have a specimen that is not entirely white then it is possible to improve the quality of the image by capturing another picture without the grating (called complementary images). Division of the original image by the complementary image results in an single coloured image, except for the moiré fringes (and grating lines), which makes them better visible [cf. Sec 2.2]. This is done because otherwise the moiré fringes loose in visibility, i.e. they get more blurred by the darker parts of the specimen. There were some indications that the colour of the specimen would not matter so much if the angle is rather high, but that causes other problems [cf. Sec. 2.3].

A problem with the complementary images is that it is difficult to know what brightness the image should have when taken from the camera. If the complementary image is too dark or too bright, the result is worse than if you had not divided the images (because dividing them introduces discontinuities) [Figure 4c]. The reason why it is not possible to use the same parameter settings is that much of the intensity of the light is lost through the grating, thereby making the complementary image far brighter than the original. The visual brightness of the image depends on all the parameters, but you always want to have parameters that makes the complementary image darker than it was with the same settings as for the moiré image [cf. Sec 2.1]. Values of Brightness and Gamma of 100-150 usually give good results (with Contrast and Red intensity max).

The complementary image is good if the centre of the moiré fringe image corresponds, in colour, to the same area on the complementary image [Figure 4b].

For a white surface (like the tablet form in the lab) it is normally not necessary to take complementary images as the contrast is high enough anyway. On a dark specimen, on the other hand, it can be rather helpful even if it is single coloured.

When dividing the moiré fringe image with the complementary image you can also use specimens with periodic colours, which would be rather difficult otherwise. Masking those frequencies away could solve it but that would also destroy some information. A darker specimen requires a coarser grating to get good contrast because the visibility of the moiré fringes is worse [cf. Sec. Quality]. With a good complementary image, this might be solved, but this has not been investigated yet. The specimens that have been measured so far (with complementary images) are a piece of flat plasticine with a round piece of white paper on, and a piece of plasticine with some holes and hills on it. The first one worked rather nicely and with a visually good complementary image the trace of the white paper was almost erased, [Figure 4b] which was good. The rms-error was less than 0.1 mm, as for a normal plane [cf. Sec. See also Sec 2.4.]. The other one was not quite as good and the complementary images was not of much help. Both of them were measured through a 20-lpi grating. Brightness and Gamma was 110-115 for the first specimen (Contrast and Red intensity were max).



a) Moiré fringe image, without any use of a complementary image. b) Top image a) divided by top image b) [rms 0.1 mm]. c) Top image a) divided by top image c) [rms 0.5 mm].

Figure 4: *Images of flat plasticine. First row are taken images, second row are images processing.*

Another reason for taking complementary images is that it allows you to see what part of the surface that is left after cutting the image (in the program pgm_raw) to 512x512 pixels. It is then easier to see how good the correspondence is between the image after analysis and the surface.

See also Sec 3.5.

3 Analysis

3.1 General

This is a description of all that happens from the grabbing of the image to the resulting height image:
The files are saved in tiff format from the image grabber, then converted to pgm format by xv and then converted to raw format by pgm_raw. The raw format can be read by FRAN, which masks away the irrelevant frequencies (very low, very high and the grating frequency) and saves the image after unwrapping the phase map. The unwrapped phase map is then fitted to a plane to get rid of the original tilt of the specimen [cf. Sec 2.2]. See also 'Moiré Profilometry" by Richard Lang for a more detailed explanation of some parts.

In order for the FRAN program to work with Fourier transforms all images must be an even power of two in height and the same in width. Presently this is achieved by taking out the central 512x512 pixels of the original 768x576 pixels image. Another possibility would be to stretch the image to 1024x1024 size. This might introduce some faults but it should be possible to do it, as the images are rather continuous.

After the unwrapping of the phase map the edges of the images can often be discontinuous, so it might be worth it to cut away those parts as well.

3.2 Automation

For the process to be made automatic, a couple of c-shell scripts have been written. For letting FRAN run a script.crl, it is possible to do this automatically by writing
printf ”o\nscript.crl\nqu\n”|fran
FRAN then processes script.crl and quits afterwards. \n means <Enter>, o, off-line mode, and qu quit. It is important that the qu because otherwise there will be a 'hanging pipe', which will produce an error. There are several scripts described in Appendix A: Description of files, like Conv, Scale, fr, li, mfr, mli and si, that are rather useful. It can be worth noting that FRAN has to have a certain format of the .crl files and that this causes some minor problems. For loading and saving images in the right format there are li and si respectively. For reducing the data to a more convenient size (e.g. for visualising in Gsharp) Scale is good. When displaying the results we have more information than needed, because the resolution is not good enough. This means that the images are rather good even after scaling them to 1/64 of their original size (8*8). For processing an image through FRAN using an appropriate mask for the number of lpi, fr is good, and mfr does the same thing with a list of complementary images. Some of these scripts have a standardised notation [cf. Appendix B: Notations, and Appendix A: Description of files] which should be used for them to work properly.

3.3 Quality

It is desirable to make the moiré fringes as visible as possible, with a high contrast to the other parts of the image. If these conditions are fulfilled the analysis will usually be rather smooth, and the result good. If the image is good, the number and the size of the poles will be small when the phase map is unwrapped. Often several small are better than one big, as the really small ones tend to be 'knuckles', as described in Sec. 2.2. See also Qual_Compl in Appendix A: Description of files for a discussion of the complementary images and a way to determine their quality. This is also discussed in Sec. 2.4.

A surface with many height differences will be harder to measure and would require a good resolution [cf. Sec. 2.2] to produce any reliable results. For a flat plane, the residual mean square error, i.e. the mean deviation from a plane, was about 0.1 mm [cf. Sec. See also Sec 2.4.] and this corresponded roughly to 0.4 in phase scale. A major problem that has to be solved if this is to be handled by someone who is unfamiliar with optics is that it is difficult to know when an image actually is good. There seems to be no sure way of telling this, though many attempts have been made in the world.

3.4 Masks

A mask is a section in Fourier space, where the frequencies are kept, while the rest of the frequencies are erased. The reason for doing this is that the grating frequency can be filtered away, thereby leaving the image (almost) free from it. Some small perturbations can also be filtered in this way. The masking also results in a phase map, which we need. The phase map is acquired by filtering away the lowest frequencies, i.e. the frequencies nearest the centre of Fourier space. The mask is generally rectangular and extended in one direction, see Figure 5. The disadvantage with masking is that some of the information is lost, and to reduce this it can be important to choose an appropriate mask. Basically, the more of the frequencies that can be preserved the better, but the grating must be filtered away. In order to do this there are some predefined masks in mask.crl, and they are:

Lpi	x_{min}	x_{max}	Y_{min}	y_{max}
20	246	266	258	308
30	244	268	258	288
40	236	276	258	288

These masks work rather well for all the specimens that have been tried so far. It is difficult to improve the quality by choosing a different mask, and although a smaller mask will reduce the discontinuities, it will 'smear out' the image, thereby reducing the resolution. It might be worth changing them a bit to see if it is possible to get some better masks, but that is not likely to produce any significantly better images. It might be slightly better to have a narrower mask as the resolution is not good enough to keep all of the information anyway. These masks are made so that they cut right before the grating frequencies and they assume that the grating is vertical (| | |) and the moiré fringes are horizontal.

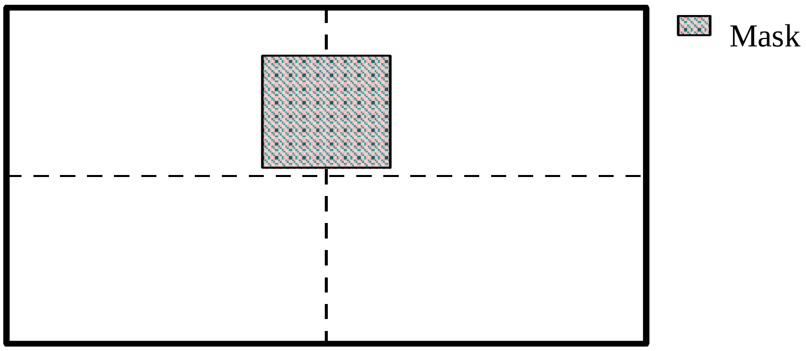


Figure 5: *Illustration of how a mask is chosen in Fourier space.*

3.5 Multicoloured specimens

As an alternative to taking a complementary image, [cf. Sec 2.4] the image of the absolute value (after masking) was used to divide with. This did not give any good results, probably due to the discontinuities in the absolute value image. Another possibility would be to have a set-off brightness and make everything above that brightness white and everything below it black. This gives a white background and clear, black moiré fringes. The problem with this method is that the setoff brightness depends on the image, and that some of the information is lost when masking the image. From the few experiments done using this method it does not seem so good, but it might be worth investigating it a bit further.

How is it then possible to determine the quality of a complementary image so that the result will actually improve after division? This question proved to be rather difficult to answer. It is possible to do it manually [cf. Sec 2.4] but hard to find a reliable way of doing it automatically.

A program "Qual_Compl" was developed in an attempt to make an automatic quality control. The program worked fine for a flat piece of plasticine (blue) with a white piece of paper on it. The best images according to Qual_Compl were much better than the images that Qual_Compl gave a low score. However, the algorithm is rather empirical, and should not be trusted to any higher extent. At least it may be able to give a rough estimate of the quality. It might be worth seeing how good/bad the estimate of Qual_Compl is, even if part of the algorithm is empirical, as it has actually worked so far. It could also be worth looking somewhat at the setoff used in the program. It is now only dependent on the average darkness of the image; perhaps there are more dependencies? Dividing images will not change anything of the real scale as the only thing that changes is the amplitude of the waves, not the frequencies. Therefore, when converting to real scale (e.g. with phase_height) the height will be correct even after division. When having a multicoloured image without dividing by its complementary image this seems to give discontinuities and a lower height by the border between the colours [Figure 4a]. Even when dividing with a good complementary image there might be some problems left [Figure 4b]. On the borders between different colours there are still some small bumps, and the image of the grating gets broader where the surface is dark and thinner were it is light. However, the last observation does not seem to cause any problems, probably because it does not affect the frequencies.

See also Sec 2.4.

4 Discussion

With a good specimen, it is possible to achieve a resolution of about 1mm. A good specimen is light, with the changes in height not too close and not too steep. The residual mean square (rms) error for a flat plane was about 0.3 degrees (in the phase map). In "Moiré Profilometri" by Richard Lang an equation is derived for a conversion between phase and height:

$$h = \frac{p}{2\pi \tan(\alpha)} * \phi$$

$$p = \frac{25.4}{lpi + 1}$$

where p is the period in mm of the grating and α is the angle between the camera and the light source (see Figure 3). For the plane $lpi = 30$ and $\alpha \approx 25°$, which gives a rms error of about 0.1 mm. This shows that the method in itself does not introduce much noise.

The method has several possible applications, like storing the height map of ancient inscriptions, taking imprints of parts of the body, e. g. feet. This study has mainly been concentrated on inscriptions, the goal being a good resolution and the possibility to study multicoloured specimens.

A limitation with the present method is the small area that can be studied with good accuracy, which is about 1 cm^2. In some articles, the area over the resolution is claimed to be 10^4 but the results here have only a quotient of 10^1. This should be possible to improve significantly.

An interesting question is what number of lpi that is needed for different specimens, colours, and shapes. There has been no systematic study of this, and this is important for the understanding of the method.

5 References

Lang, Richard (1998)
Cavendish Laboratory, University of Cambridge
11 May 1998, 22pp
Report written under the supervision of Dr. Tim Goldrein.

Goldrein, Tim (1998)
Personal communication
Cavendish laboratory, PCS group, University of Cambridge
Cambridge CB3 0HE
U.K.
E-mail: htg10@phy.cam.ac.uk

Appendix A: Description of files

You can get help for any program or file by typing Man file, where file is the program or file you want to have help on. This displays what is written under that file in this file. See bin/Man, below. (Note the Capital M, in Man.)

/bin:
Contains all command-scripts written for csh. Use 'Command -h', where Command is the command-name to get help on how to use them.
--

/fortan_code:
Contains the source-code for all the fortran programs. You can read the first lines of each .f file to see a short description of what they do. You can also recompile them (after eventually making some changes) by first typing 'unsetenv LD_LIBRARY_PATH' and then 'f77 name.f', where name.f is the fortran source-code. You then get the compiled program named a.out.
--

/progs:
Contains the compiled programs. Most of them has a corresponding source-code in ~/fortan_code. You can get a short help from some of these by typing 'prog -h', where prog is the name of the program
--

/work:
Contains the current work, the images and this is normally my working directory.
--

bin/Conv:
Converts all files in current working directory with .pgm to .raw and moves
them to ./new/ or to the first argument of the command. It then compresses all the .pgm files first with gzip, then with tar to pgm.tar, so you can retreive them if needed by some reason. It doesn't destroy the file pgm.tar if it exists, but adds the files to the archive. To view the archive, type 'tar tvf pgm.tar'. Conv -h gives you a slightly shorter help.
If the directory doesn't exist it is created. This script uses the fortran-program pgm_raw.
--

bin/Man:
Gives you some futher help on a command described in /home/eje21/Doc/filedescription, i.e. this file. This command displays the lines between the command: and the double dashes in the end. This is usually a bit more than in command -h. This can also be used to get help on files other than commands, for example run2.crl. Note the capital M in Man.
Man -h gives you a slightly shorter help.
--

bin/Scale:
Scales test.fit with a factor and saves it as test{factor}.fit
where {factor} is the scalingfactor, default 8. The scaling makes it much quicker to display the data in Gsharp and doesn't give you a too bad resolution. This script uses the fortran-program Resize. Scale -h gives you a slightly shorter help.
--

bin/fit:
Do a plane fit on 'test.uw' and save as 'test.fit' then scale down
the image to 1/64 of size to be faster to load i Gsharp. The script then

waits for input and if this is other than 'q', it does the same thing again. note that it might take a while before anything happens after you've run it. It displays the rms-residual. It's rahter neat to have this running in conjunction with using run#.crl, which produces files named test.fit, then you only have to move the mouse-pointer to a window having this program running, and hit <enter>, to make a new fit. Then you can load the fitted image into fran by 'li<enter>test.fit<enter>r<enter>array<enter>'. fit -h gives you a slightly shorter help.
--

bin/fr:
This script takes a raw-file and processes this through fran. The first argument is the original image.raw while the second one is the image with which to divide image.raw, to compensate for a multicoloured specimen. Both of the images are assumed to be in ./new/, but if they aren't there the program uses the path given as the argument. The raw images has to have vertical grating lines and horizontal Moire fringes for the masks to work properly. The image.raw has to contain a 2-digit number, where the first digit represents the number of lines per inch (lpi) in 10, i.e. tablet32.raw would have 30 lpi. The script then choses an appropriate mask from mask.crl, and runs the appropriate run#.crl (see their description), where # is the number of lpi. After filtering the image through fran (eventually compensating for different colours), the script fits it to a plane, prints the rms from a plane, and saves the new image as test.fit [real]. Finally it makes a scaled image named test8.fit. If you provide a 3rd argument to fr, this will be added to all file-names, to be able to have several files at the same time. The program also makes a file named run.crl. So, the program needs:
 mask.crl, a run#.crl, and an image to process along with the
 Scale-script, the li-script, and the si-script, as well
 as fran, and analyse_map.
and makes:
 run.crl, test.fit, and test8.fit in the current working directory.
fr -h gives you a slightly shorter help.
--

bin/li:
li takes a file and produces
a line (on stdout) with the correct format for fran to read. As a second argument you can specify what array to load the image to, and as a third, if you want real-format instead of integer, which is default. The output can then be saved to a .crl file. Ex
li new/test.real 5 r > test.crl
Then, in fran, run test.crl in ol-mode. This loads new/test.real to array 5.
li is used by fr, mli.
li -h gives you a slightly shorter help.
--

bin/mc:
Creates a new shell-script with a default-"look", i.e.

```
 #

 if ($1 == "-h") then
  cat $0|head -2
  echo usage: command ''
 else

 endif
```

If the command with argument -h is written, the program will print with a short help (the first two lines of the command), and usage, where you'll have to write something in the '' above.
Without any argument a temporary command name tmp# is created, where # is the lowest unoccupied number below 10, i.e. 1,2,3...,9.
mc -h gives you a shorter help.
--

bin/mfr:
Processes an image with grating with several complementary-images, i.e.
images without grating, to compare how good different complementary images
are. See also fr for further information. Ex:
 mfr T30.raw T30c1.raw T30c2.raw
or, for short
 mfr T30.raw T30c*
It also makes a vague estimation of the quality of the Complementary images
with the program qual. See help on this program. If the files lie in new/,
you don't have to type new/ before the filename.
mfr -h gives you a shorter help.
--

bin/mli:
Make a list of files to be loaded into different arrays in fran and saves
the it depending on the format. For real images the name is fit.crl, and for
integer images raw.crl. If the type is b=both it assumes that the real files
are named test.fit#, with # a number as produced by for example mfr. In
that case there has to be an equal number of test.fit# files and files
gven as the second argument and forth. If thefiles exist before, they are
overwritten.
Ex:
mli b new/T20.raw
Uses: li
Produces: fit.crl and or raw.crl
mli -h gives you a shorter help.
--

bin/si:
Works exactly like li, except that it prints a save image line instead of a
load image line. See Help li. Default is real.
Ex:
 si test.fit
Uses: -
Produces: a correct line for fran's ol, on standard output.
si -h gives you a shorter help.
--

progs/Qual_Compl:
This gives a ROUGH estimate of the quality of the complementary image, ie
the image without grating. The quality is quality with respect to the image
after dividing the Moire-fringe image with the complementary image.
The algorithm is quite empirical, but is based on that it compares the parts
of the images that is above a certain darkness (this is dependent on the
mea-value of the image) to get rid of the grating parts of the image,
which are very dark. The program makes an estimate of how different they
are (which should be changed) and produces a result. The result is either
good, or bad and a number, where a higher number means 'better' quality.
Ex:
 Qual_Compl<enter>
 tablet30.raw<enter>
 tablet30c.raw<enter>
Uses: two input files to be compared
Produces: an estimate on stdout. The higher the better (eventually).
Qual_Compl -h gives you a shorter help.
--

progs/Resize:
Scales an image down by a factor of 2, 4, or 8, to get them nicer to handle
in Gsharp. This is done by taking every second (forth or eighth) pixel in as
well x- as y-direction.

```
.....              .   .
.....
..... would become       with a scaling of 4.
.....
.....              .   .
```

Ex:
 Resize<Enter>
 tablet20.fit<Enter>
 4<Enter?
 tablet20.8.fit<Enter>
or
 printf "tablet20.fit \n 4 \n tablet20.8.fit \n" | Resize
Uses: an input file to be scaled
Produces: a scaled output file
Resize -h gives you a shorter help.
--

progs/Annulus:
Creates a halfcircle with a hole in it (a half annulus) , and saves it as a mask, named annulus.mask. You give the inner and the outer radius in the progrma.

```
    _--_
   /    \
  /      \
  |__/\__|
```

This can be loaded into fran with the lm command.

Ex:
 Annulus<Enter>
 5<Enter>
 15<Enter>
Uses: -
Produces: logical (mask) file annulus.mask
Annulus -h gives you a shorter help.
--

progs/analyse_map:
Fits a plane to a real 512x512 image and subtracts it from the image. This file has no working analyse_map.f file. It has to be combined, with some other programs, but Tim nows how to do it, should you want to change it. The image is much easier to see in fran if you first run this program on a (real) file.
Ex:
 From running run2.crl in fran you get a test.uw file. Now
 analyse_map<Enter>
 test.uw<Enter>
Uses: File to subtract a fitted plane from.
Produces: File with the fitted plane subtracted.
--

progs/cos_fringe:
Creates a cosinus-grating to be printed. I have't tried this program.
I have found no source in rl202/fort_source
Ex:
 ?
Uses: ?
Produces: ?
--

progs/dat_real:
Converts direct real to unformatted real. See ~rl202/fort_source/dat_real.f.
I haven't tried this program.
Ex:
 dat_real<Enter>
 test.dat<Enter>
 test.real<Enter>
Uses: a direct real file
Produces: an unformatted real file
--

progs/im_combine:

If you take to images beside each other, this program is supposed to be able to combine them. With this you could theoretically 'piece together', the entire tablet by using part images, and combining them. The program has not been tested in practice.

```
 Ex:
  ____          ____
 |   _|__      |    |__
 |  | |  |  => |       |
 |__|_|  |     |__     |
    |____|        |____|
```

or something like that.
Source available at ~rl202/fort_source/im_combine.f
im_match should be used first, but read the source. I haven't tried this program.
Uses: Two images, and an offset file (through im_match)
Produces: a combined image
--

progs/im_match:
Auxilliary program for im_combine.
~rl202/fort_source/im_match.f
I haven't tried this program.
Uses: Two files
Produces: offset file
--

progs/pgm_raw:
Program to convert pgm to raw. Assumes images are 768*576 and outputs the central 512*512 region. See also Man Conv, and source at
~eje21/fortan_code/pgm_raw.f
Note that this source is slightly different from
~rl202/fortran_source/pgm_raw
Ex:
```
 pgm_raw<Enter>
 tablet30.pgm<Enter>
```
gives the outputfile tablet30.raw
Uses: pgm-file 768*576
Produces: raw file for fran 512*512
pgm_raw -h gives you a shorter help.
--

progs/phase_height:
Converts a phase-map to a map with mm as unit. Asks for the grid (period), and the incident angle. I haven't tried this program.
Ex:
 ?
Uses: file with an unwrapped phase-map.
Produces: file with scale in mm.
--

progs/raw_mask:
Program to create masks from raw images. I haven't tried this program.
Ex:
 ?
Uses: raw file
Produces: mask file
--

progs/raw_pgm:
Program to convert between RAW format for Fran and true PGM format
for xv and other graphics programs. Assumes 512*512 input image.
I haven't tried this program.
Ex:
 ?
Uses: raw file
Produces: pgm file

--

progs/real_pgm:
Program to convert between fortran real output and a grayscale M. The
real array is scaled so the PGM uses the full range of grays.
I haven't tried this program.
Ex:
 ?
Uses: real file
Produces: pgm file
--

work/2D.gsl:
This is a script for taking an image scaled down by 4 and display it as a
colour map in Gsharp. See also 3D.gsl, which usually is better, and works
with images scaled by 8 instead. If you use it, change the path in
import_binary on the first line for loading a different file.
Used by: Gsharp
--

work/3D.gsl:
This is a script for taking an image scaled down by 8 (as from the output
of fr, mfr, Conv) display it as a 3D surface in Gsharp. To load a
different file, change the path after import_binary. This script is better
than 2D.gsl.
Used by: Gsharp
--

work/mask.crl:
This is used to define suitable masks for different number of lpi. These
masks work rather well, and it's difficult to really make any big
difference by choosing other masks. Mask 5 is used by the unwrapping part,
of run#.crl [#=1,2,3,4] and mask 1 can be changed to whatever size the
user wants. It's very narrow as default.
It's good to run this control-file first thing after loading fran, that is
fran<Enter>o<Enter>mask.crl<Enter>.
See also run1.crl.
Used by: fran, run#.crl, fr, mfr
--

work/image_size.def:
This file defines how large and how many the arrays shall be in fran.
To add another array [default 8, in ~eje21/work 16], just add another
line.
Used by: fran
--

work/mask_size.def:
This file defines how large and how many the masks shall be inte fran.
To add another mask [default 4, in ~eje21/work 5], just add another line.
Used by: fran
--

work/run1.crl:
This is a control file for fran (script), which takes the image from array
1, fourier-transforms it, masks it with mask 1 (which is very narrow as
default), inverse fourier transforms it, calculates the phase, unwraps the
phase map and saves the result as test.uw. This file can then be processed
by for example fit. After that you can load test.fit as real and display
the result. run1.crl needs a mask 1 defined and therefore you
might want to run mask.crl first. See also run#.crl, mask.crl and fit.
The control-file uses only arrays 1-4.
Used by: fran
--

work/run2.crl:

This is for analysing 20 lpi images with vertical grating lines and horizontal Moire fringes.
This is a control file for fran (script), which takes the image from array 1, fourier-transforms it, masks it with mask 2 (which is default for 20 lpi, if you've run mask.crl), inverse fourier transforms it, calculates the phase, unwraps the phase map and saves the result as test.uw. This file can then be processed by for example fit. After that you can load test.fit as real and display the result. run2.crl needs a mask 2 defined and therefore you might want to run mask.crl first. See also run#.crl, mask.crl and fit.
The control-file uses only arrays 1-4.
Used by: fran
--

work/run3.crl:
This is for analysing 30 lpi images. with vertical grating lines and horizontal Moire fringes. This is a control file for fran (script), which takes the image from array 1, fourier-transforms it, masks it with mask 3 (which is default for 30 lpi, if you've run mask.crl), inverse fourier transforms it, calculates the phase, unwraps the phase map and saves the result as test.uw. This file can then be processed by for example fit. After that you can load test.fit as real and display the result. run3.crl needs a mask 3 defined and therefore you might want to run mask.crl first. See also run#.crl, mask.crl and fit.
The control-file uses only arrays 1-4.
Used by: fran
--

work/run4.crl:
This is for analysing 40 lpi images. with vertical grating lines and horizontal Moire fringes. This is a control file for fran (script), which takes the image from array 1, fourier-transforms it, masks it with mask 4 (which is default for 40 lpi, if you've run mask.crl), inverse fourier transforms it, calculates the phase, unwraps the phase map and saves the result as test.uw. This file can then be processed by for example fit. After that you can load test.fit as real and display the result. run4.crl needs a mask 4 defined and therefore you might want to run mask.crl first. See also run#.crl, mask.crl and fit.
The control-file uses only arrays 1-4.
Used by: fran
--

work/unwrap.settings:
Preferences for the unwrapping procedure in fran. This does not have very much effect on the quality of the results as far as I've seen.
Used by: fran
--

/fran:
Fringe-analysis program. Used to remove the perturbations and the grating (through masking) from an image, so that the Moire-fringes will be better visibile. Also used to convert the image to a phase map.
Type ? in program to get list of commands.
Note that in ol mode the filenames can't be longer than 20 characters.

Commands: (in the order normally used)

ol   change to off-line analysis

li   load image
di   display image

[dv   divide two images - when compensating for the natural colour
            of the specimen]

za   set array to zero

```
ff          fourier transform

de          define mask
(lm         load mask)
ma          mask array

if          inverse Fourier transform

ph          phase/modulus
uw          unwrap

si          save image

(rf         rotate/flip image - as some of the images are upside-down)
```

Used by: fr, mfr
--

/fran.crl:
Control-file created by fran as a log of everything you do (in fran). If you want to run it but remain in fran, make sure to edit away the 'qu' in the end.
Used by: fran
--

Appendix B: Notations

The notations with a * before are required for some scripts to work properly [cf. Sec 3.2].

See Appendix A: Description of files, to see which scripts needs which files.

In image filenames:

T	Tablet, the lowest part to the right.
P	A piece of two-coloured plasticine.
1x	10 lines per inch [x is a digit].
* 2x	20 lines per inch.
* 3x	30 lines per inch.
* 4x	40 lines per inch.
c	Complementary image, i.e. image without grating.
fit	Real format, a plane fitted to the surface has been subtracted.
raw	Raw data (integer), as read to FRAN. Converted from pgm by Conv.
digit	Counting number

Ex:

P23c2.raw is the 2nd complementary image to the 3rd Plasticine image with 20 lines per inch, and in raw (integer) format.

T32.fit is the 2nd Tablet image with 30 lines per inch in real format. This has already been processed, and can be viewed in phase-format.

* The unwrapped phase-images are called test.uw and the phase-images with no tilt (i.e. images that a fitted plane has been subtracted from) are called test.fit, both eventually with a number after.

Appendix C: Full scale pictures

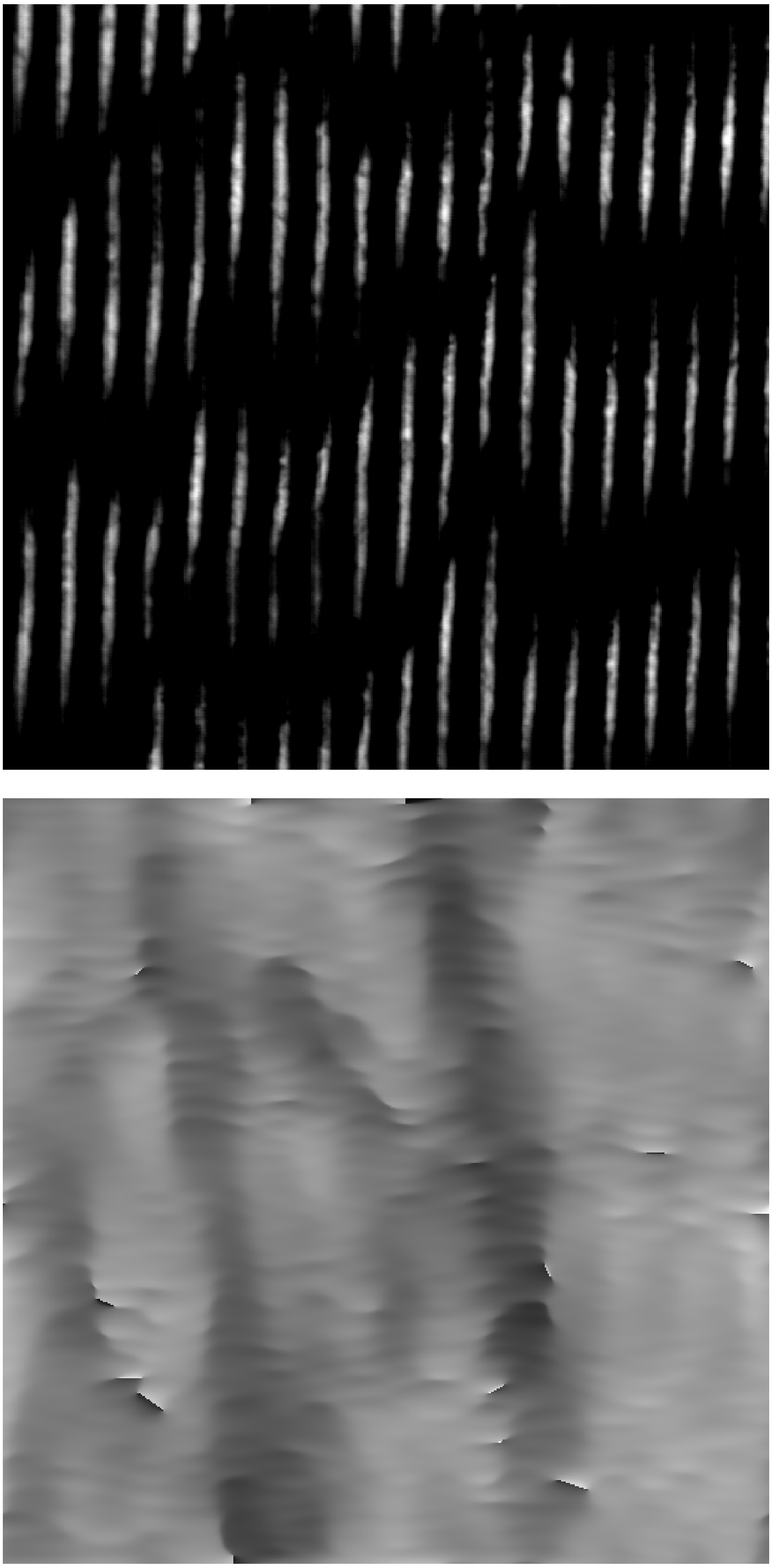


Figure 1a

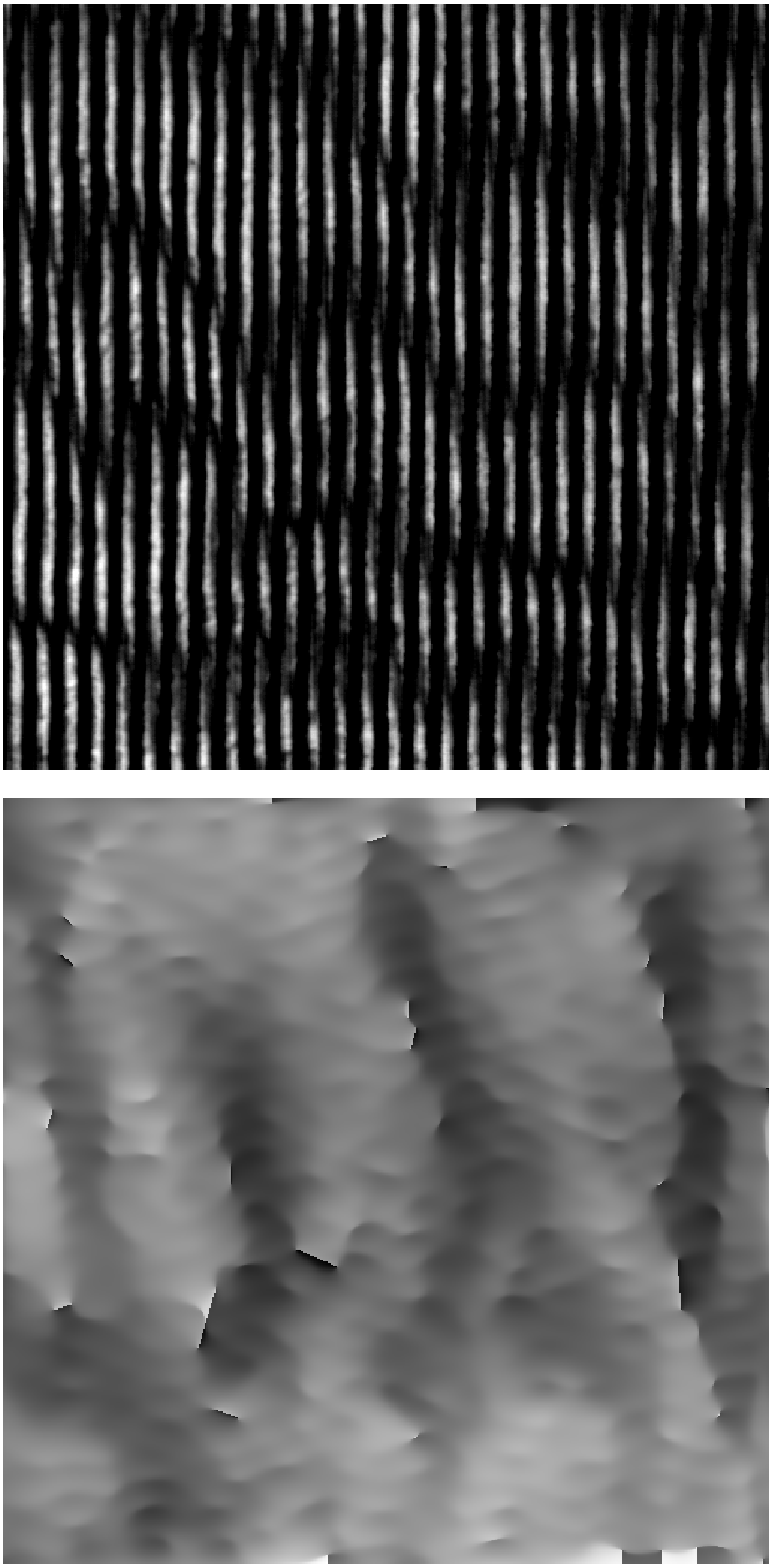


Figure 1b

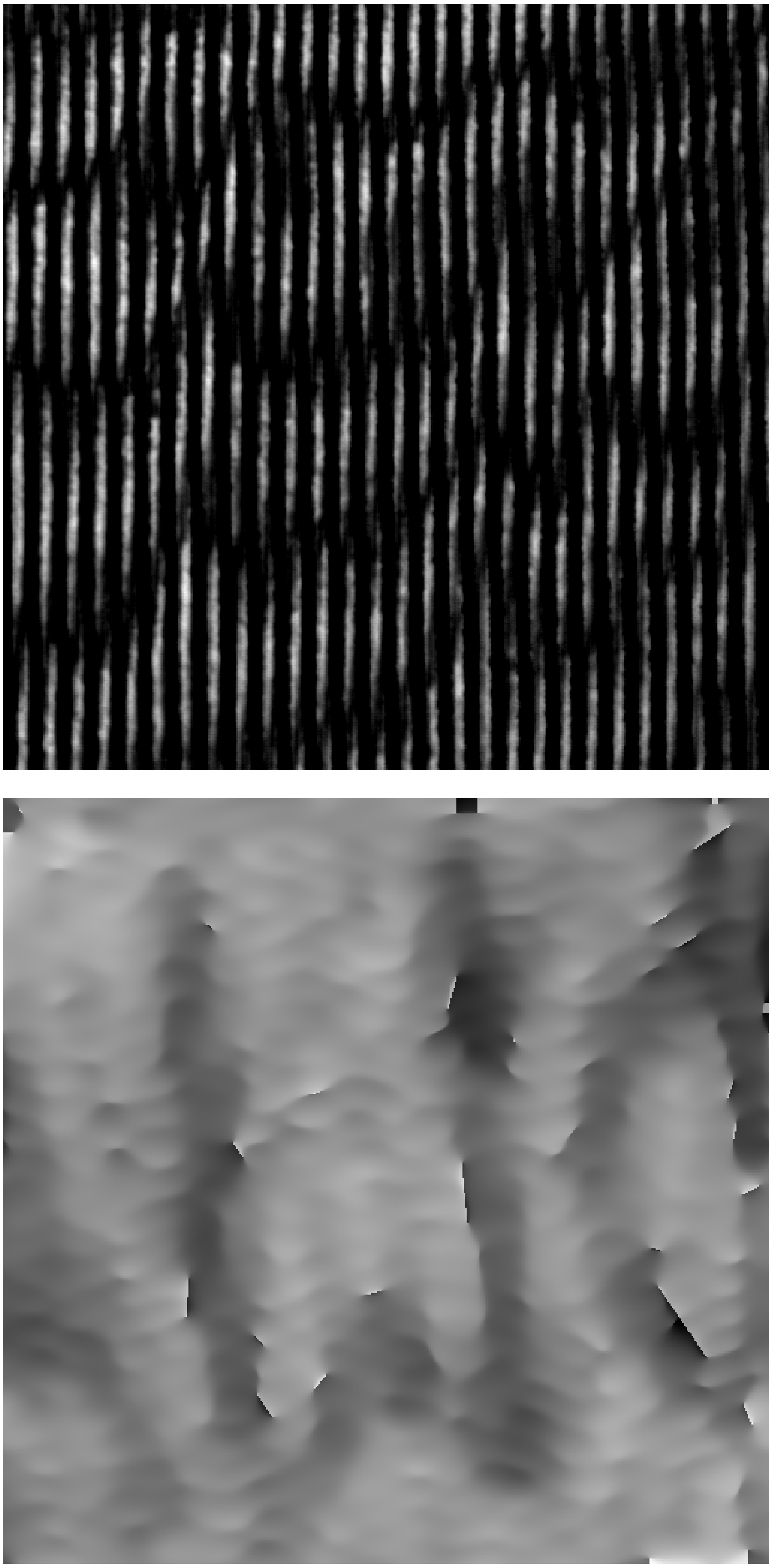


Figure 1c

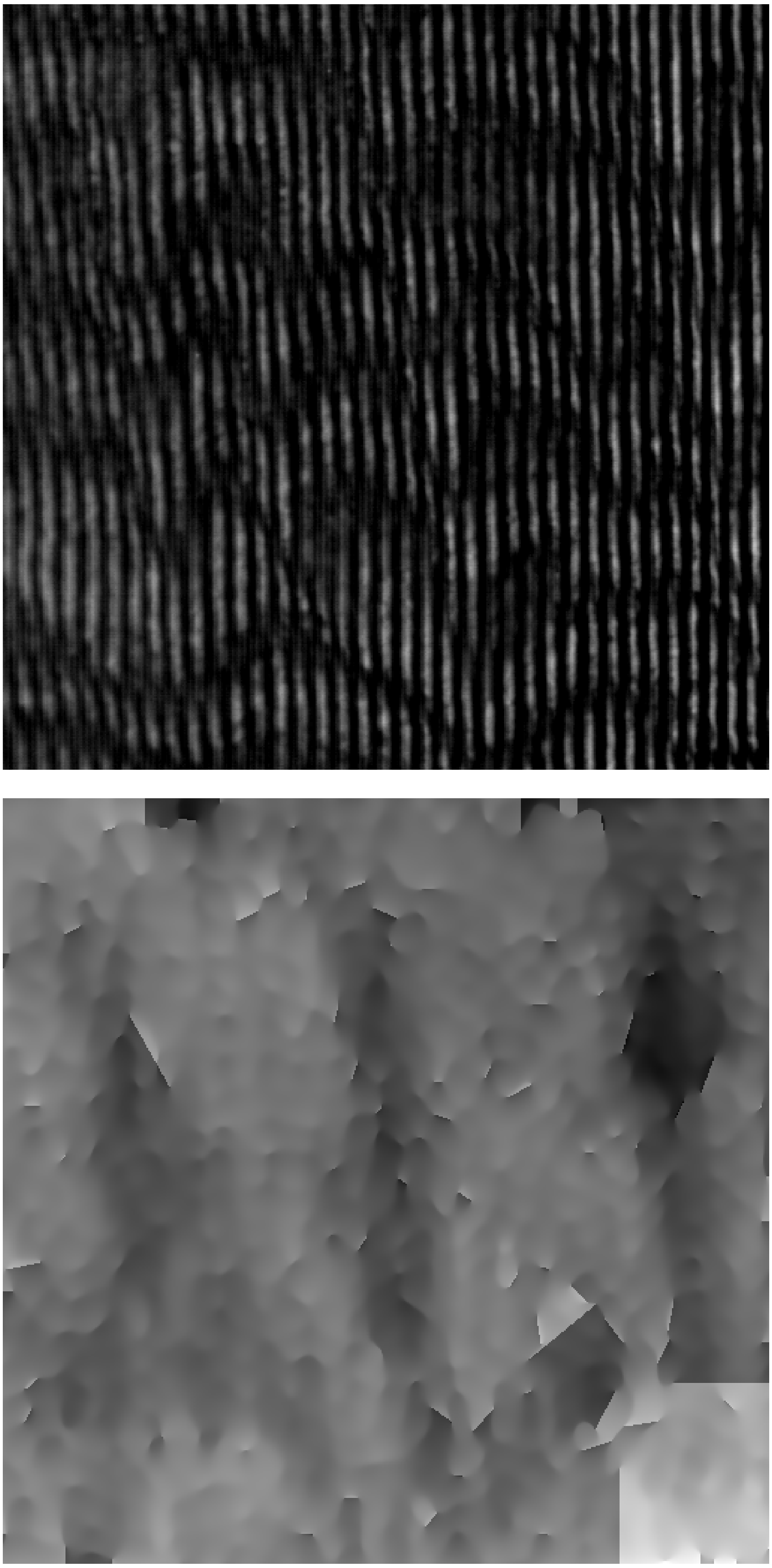


Figure 1d

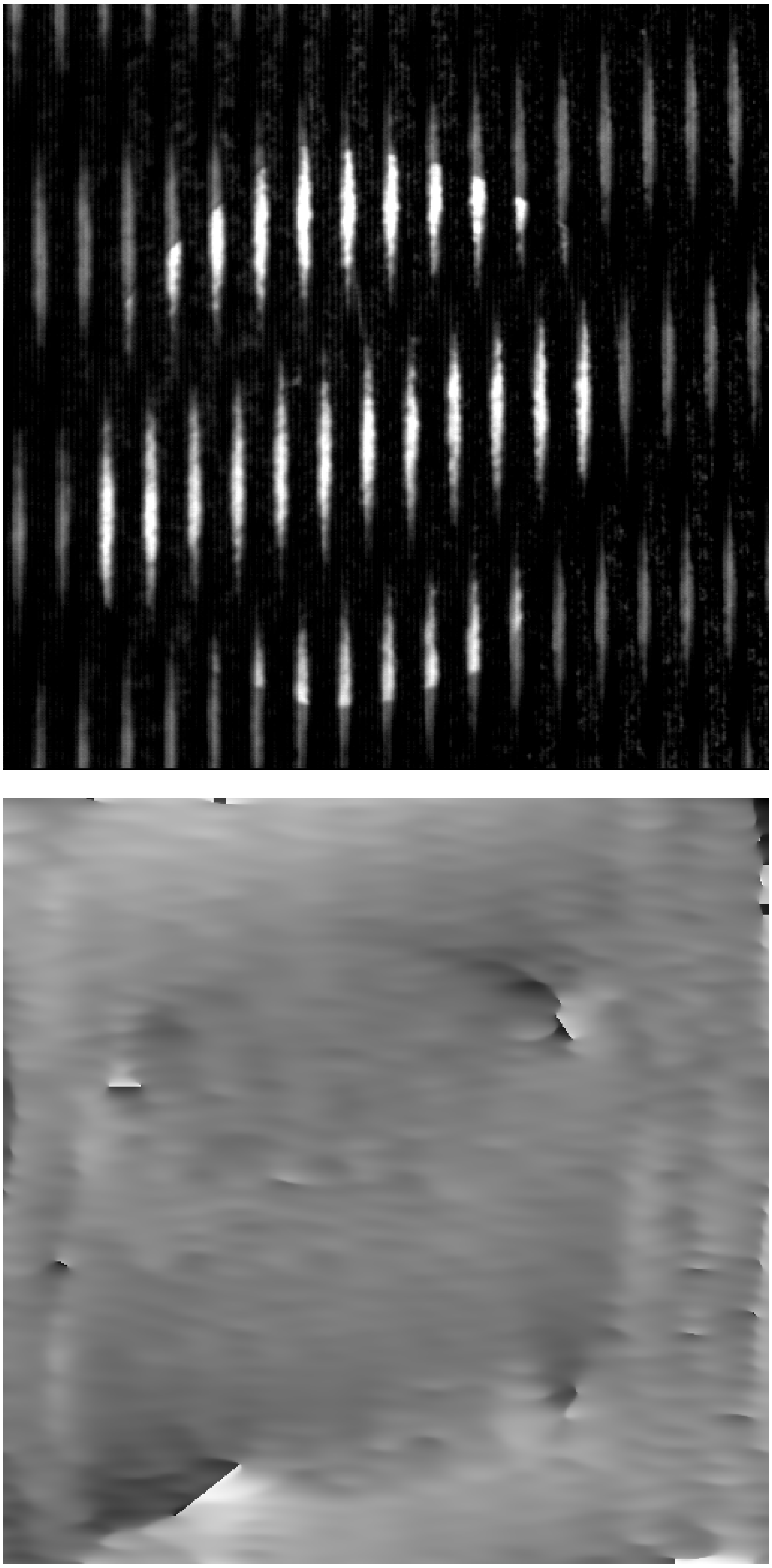


Figure 4a

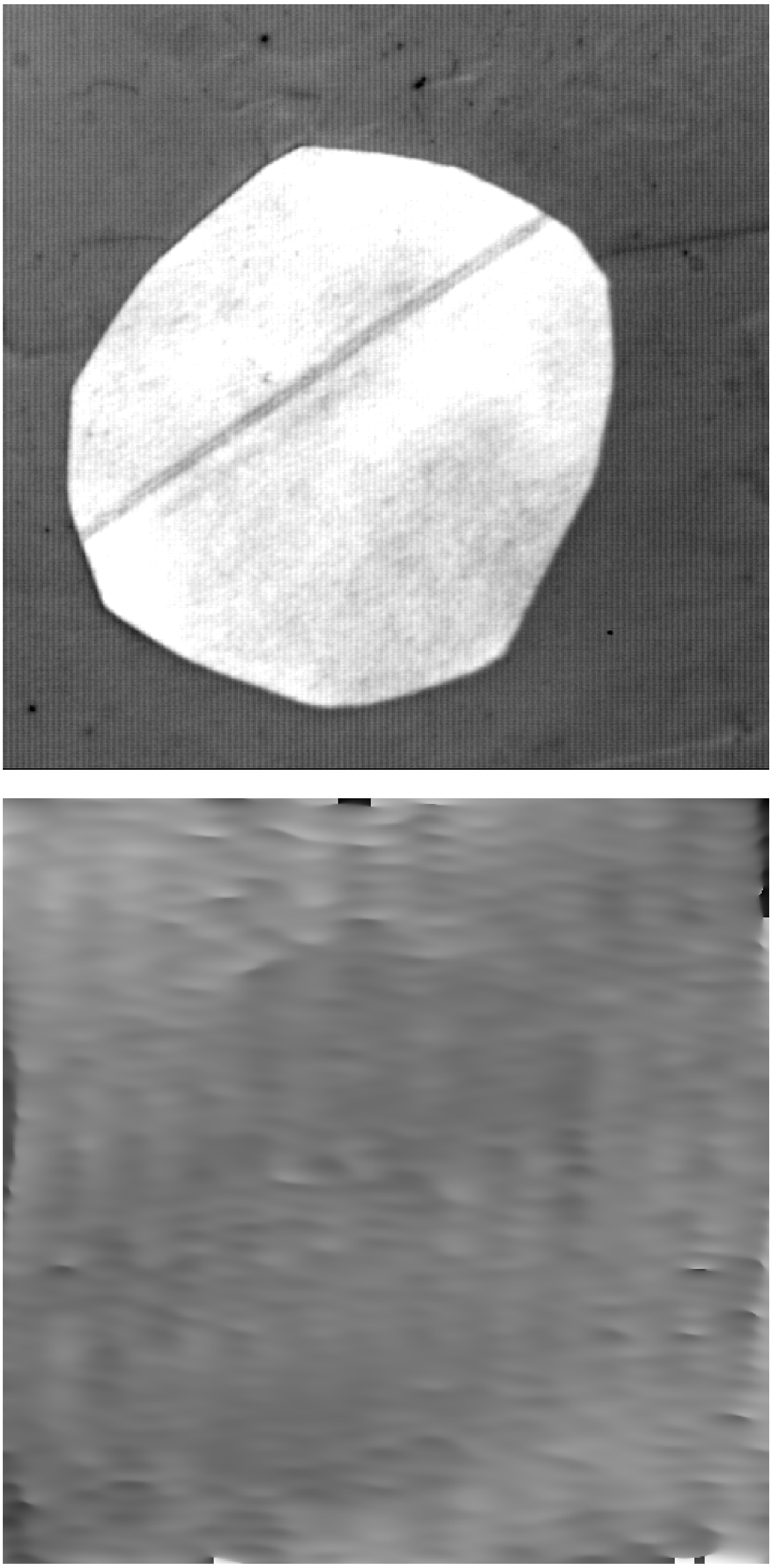


Figure 4b

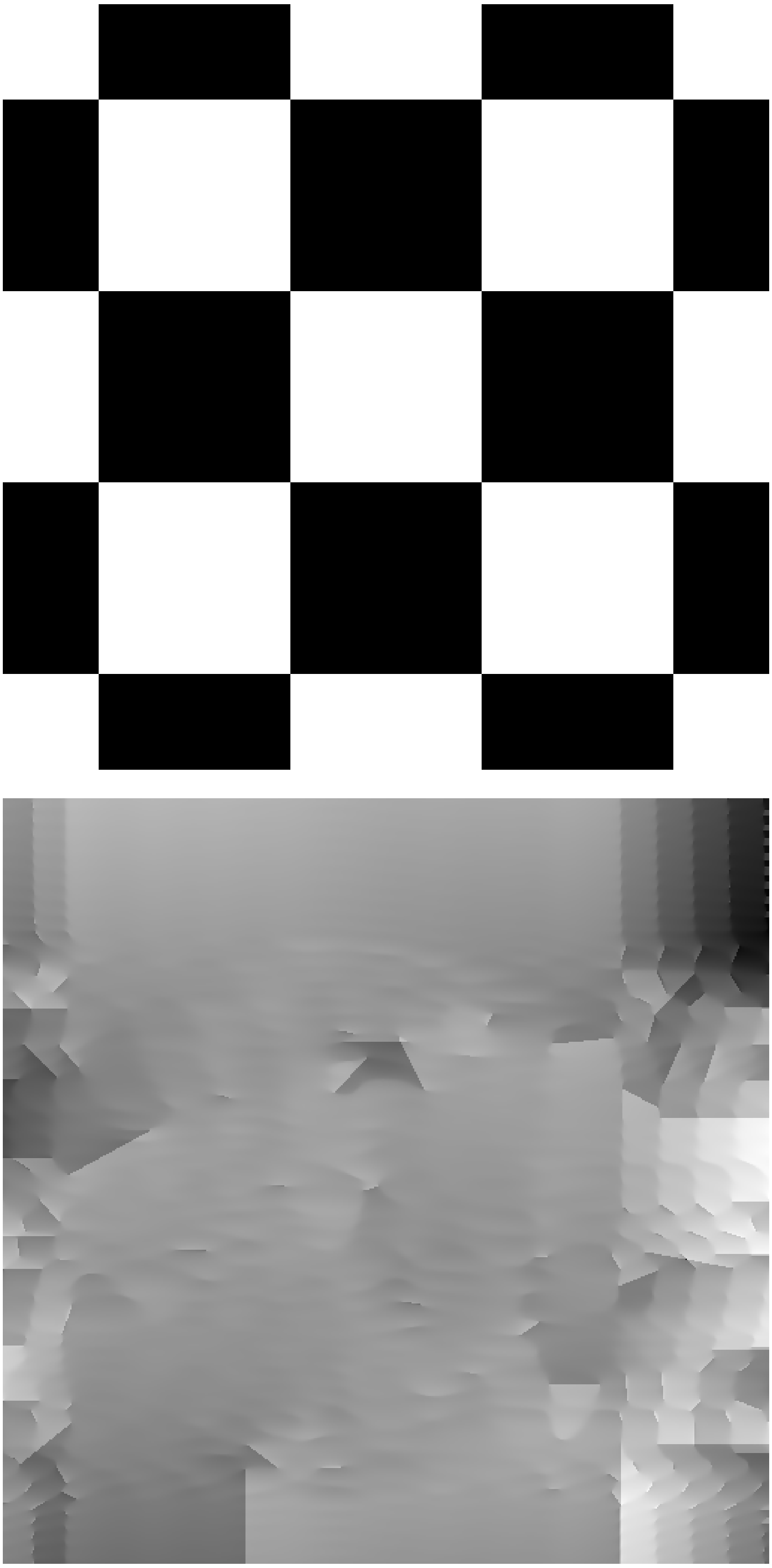


Figure 4c