

Problem 1.

- (a) The image `pattern.pgm` contains a texture consisting of parallel sine waves.

The middle of the three points is located exactly in the centre. It corresponds to a frequency 0, i.e. to a constant component in the image (rescaled average grey value).

The other two points, located at positions symmetric to the centre, correspond to one single frequency namely that of the “wave” that fills the image.

A frequency here is a vector, having a direction and a norm. The direction corresponds to the “front propagation” direction of the wave (and is therefore perpendicular to the direction of the stripes) while the norm (distance to the centre) expresses the density of the periodic stripes.

There are two explanations why the points come in pairs. First, one verifies using the formula of the DFT that a real-valued image must have a point-symmetric Fourier spectrum (Fourier coefficients at symmetric positions are complex conjugates of each other). Second, it is impossible to distinguish frequencies which are equal up to the sign.

- (b) **Cut-off errors.** Each of the three Gaussians is cut at the image boundaries, and treated as periodically repeated by the Fourier transform. While for the small kernels `gauss1.pgm` and `gauss2.pgm` the values at the image boundaries are close enough to 0 to leave no visible effects in the Fourier spectrum, the cutting off of fairly large values in `gauss3.pgm` spoils the rotational invariance of the Gaussian itself and therefore of its Fourier spectrum.

- (c) The lines in `tile.pgm` induce in the Fourier spectrum visible beams starting off in the centre and directed perpendicular to the corresponding lines.

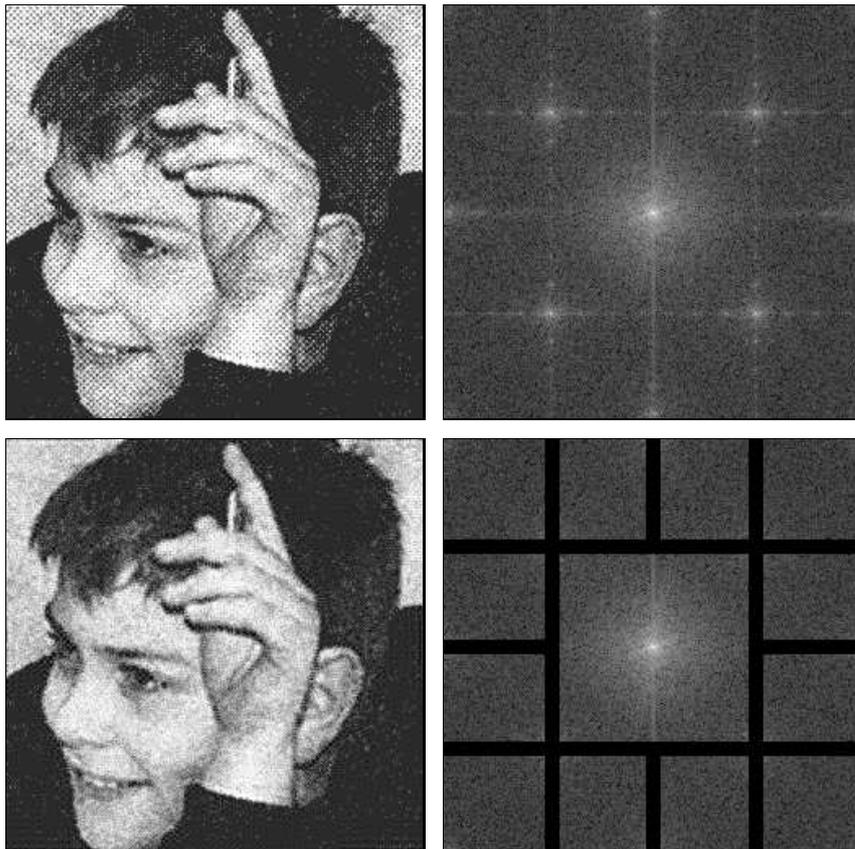
The reason is that a single edge in the spatial domain is represented by a superposition of wave-like patterns of different frequencies but equal direction. The phases and amplitudes of these waves are adjusted such that their slopes add up to give the edge at the specified location but cancel elsewhere.

A close look at the spectrum of `tile.pgm` reveals that there do not go out radially from the centre but from the image boundary. These are traces of aliasing effects.

Remembering that images and also Fourier spectra are treated periodically by the DFT we see that some of the radial lines extending from the centre do not end at the image boundaries but are prolonged beyond that boundary, wrapping around to the opposite image boundaries. Translated into frequencies: These lines depict high frequencies which don't fit in our Fourier spectrum but are represented in it by lower frequencies. This is aliasing.

Problem 2.

- (a) The fourier spectra of pisa.pgm and dancing.pgm show noticeable contributions in the high frequency range. Obviously these coefficients are related to the undesired pattern that has a relatively high frequency itself. By setting these coefficients zero the pattern can be removed successfully.



/* ----- */

```

void filter_pisa

    (float    **ur, /* real part of Fourier coeffs, changed */
     float    **ui, /* imag. part of Fourier coeffs, changed */
     long     nx,   /* pixel number in x direction */
     long     ny)  /* pixel number in y direction */
/*
allows to modify the Fourier coefficients
*/

{
long    i, j;          /* loop variables */

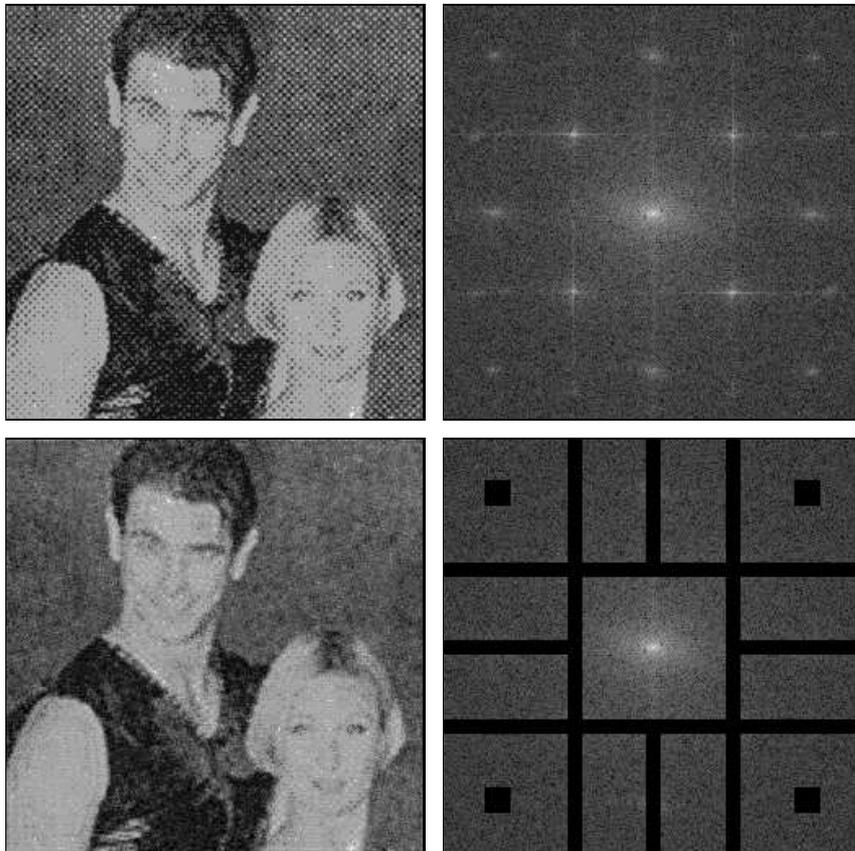
for (i=0; i<=nx-1; i++)
for (j=0; j<=ny-1; j++)
    {
        /* long left vertical line */
        if ((i >= 62) && (i <= 70))
        {
            ur[i][j] = 0.0;
            ui[i][j] = 0.0;
        }
        /* long right vertical line */
        if ((i >= 187) && (i <= 195))
        {
            ur[i][j] = 0.0;
            ui[i][j] = 0.0;
        }
        /* long upper horizontal line */
        if ((j >= 62) && (j <= 70))
        {
            ur[i][j] = 0.0;
            ui[i][j] = 0.0;
        }
        /* long lower horizontal line */
        if ((j >= 187) && (j <= 195))
        {
            ur[i][j] = 0.0;
            ui[i][j] = 0.0;
        }
        /* two short vertical lines */
        if ((i >= 124) && (i <= 132) && ((j <= 62) || (j >= 195)))
        {

```

```

        ur[i][j] = 0.0;
        ui[i][j] = 0.0;
    }
    /* two short horizontal lines */
    if ((j >= 124) && (j <= 132) && ((i <= 62) || (i >= 195)))
    {
        ur[i][j] = 0.0;
        ui[i][j] = 0.0;
    }
}
}
/* ----- */

```



```

/* ----- */
void filter_dancing
(
    float    **ur, /* real part of Fourier coeffs, changed */
    float    **ui, /* imag. part of Fourier coeffs, changed */

```

```

        long    nx,    /* pixel number in x direction */
        long    ny)   /* pixel number in y direction */
/*
  allows to modify the Fourier coefficients
*/

{
  long    i, j;      /* loop variables */

for (i=0; i<=nx-1; i++)
  for (j=0; j<=ny-1; j++)
    {
      /* long left vertical line */
      if ((i >= 76) && (i <= 84))
        {
          ur[i][j] = 0.0;
          ui[i][j] = 0.0;
        }
      /* long right vertical line */
      if ((i >= 173) && (i <= 181))
        {
          ur[i][j] = 0.0;
          ui[i][j] = 0.0;
        }
      /* long upper horizontal line */
      if ((j >= 76) && (j <= 84))
        {
          ur[i][j] = 0.0;
          ui[i][j] = 0.0;
        }
      /* long lower horizontal line */
      if ((j >= 173) && (j <= 181))
        {
          ur[i][j] = 0.0;
          ui[i][j] = 0.0;
        }
      /* two short vertical lines */
      if ((i >= 124) && (i <= 132) && ((j <= 76) || (j >= 181)))
        {
          ur[i][j] = 0.0;
          ui[i][j] = 0.0;
        }
      /* two short horizontal ines */

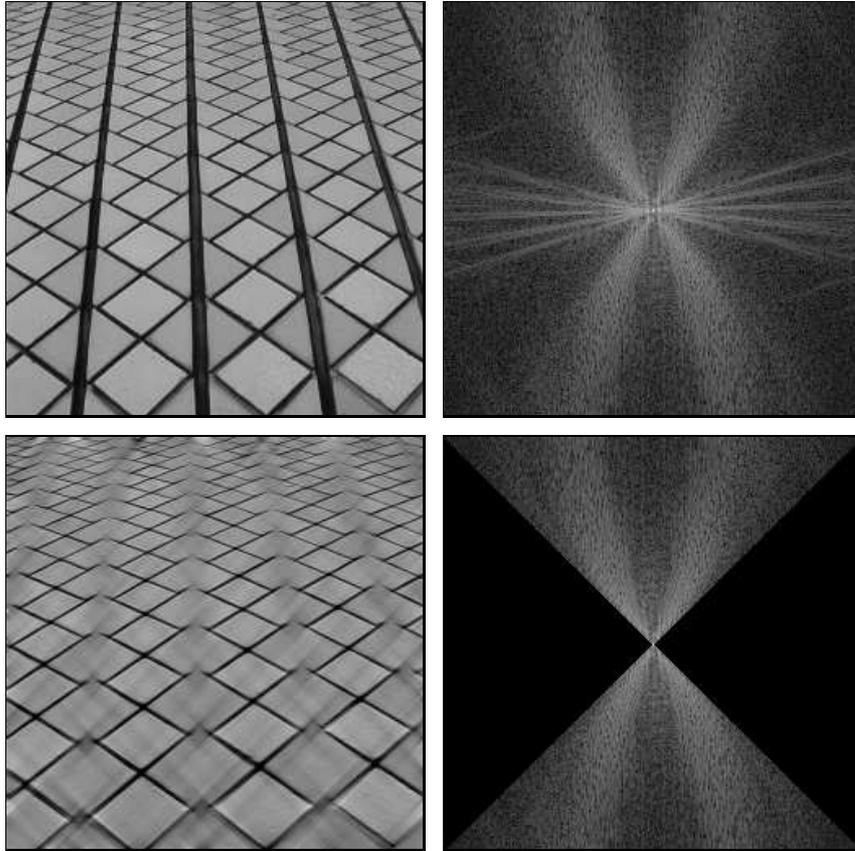
```

```

if ((j >= 124) && (j <= 132) && ((i <= 76) || (i >= 181)))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
/* upper left square */
if ((j >= 25) && (j <= 40) && ((i >= 25) && (i <= 40)))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
/* upper right square */
if ((j >= 215) && (j <= 230) && ((i >= 25) && (i <= 40)))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
/* lower left square */
if ((j >= 25) && (j <= 40) && ((i >= 215) && (i <= 230)))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
/* lower right square */
if ((j >= 215) && (j <= 230) && ((i >= 215) && (i <= 230)))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
}
}
/* ----- */

```

- (b) The dominant black lines (vertical) are represented by the five horizontal lines in the Fourier spectrum. By setting the corresponding coefficients in the Fourier domain zero, the dominant lines in the original image disappear.



```

/* ----- */
void filter_tile

    (float    **ur, /* real part of Fourier coeffs, changed */
     float    **ui, /* imag. part of Fourier coeffs, changed */
     long     nx,   /* pixel number in x direction */
     long     ny)  /* pixel number in y direction */
/*
  allows to modify the Fourier coefficients
*/
{
long   i, j;          /* loop variables */

for (i=0; i<=nx-1; i++)
  for (j=0; j<=ny-1; j++)
  {
    /* set left triangle zero */
    if ((i < 128) && (i-128 < j-128) && (i-128 < -j+128))

```

```
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}

/* set right triangle zero */
if ((i > 128) && (128-i < 128-j) && (128-i < j-128))
{
    ur[i][j] = 0.0;
    ui[i][j] = 0.0;
}
}
/* ----- */
```