Differential Imaging using Hard X-Rays from a Laser-Produced Plasma

Diploma Paper
by
Christer Andersson
Lund Reports on Atomic Physics, LRAP-174
April, 1995

Abstract

A laser beam from a terawatt laser was focused onto a metal target with high atomic number (gadolinium and tantalum). The intensity in focus was about 10^{18} W/cm^2 and a plasma was produced. The plasma emission creates a spectrum composed of continuous Bremsstrahlung with a superimposed discrete characteristic line spectrum. Two images were taken of objects containing for example a gadolinium solution, which worked as a contrast medium. One image was taken using tantalum as a target and one with gadolinium. The images were read out and stored as two digital images. The images were spatially overlapped and subtracted, and small changes in the images could be seen, depending on how much the absorption of the contrast medium varied in the two exposures. These changes appeared only when gadolinium was used as a contrast medium, meaning that the method is element selective. The method can be used for medical imaging of blood-vessels, which are normally not detectable with a conventional X-ray tube.

Table of Contents

1 Introduction and theory

Most people are familiar with the X-ray examination of, for example, a broken bone. Since 1895 X-ray examination has been an important diagnostic method and still is. Nevertheless, the conventional methods are limited, for instance, normally invisible organs, such as blood-vessels, cannot be seen. Thus, new techniques are required to be able to detect them. In the 1970's developments in digital electronics made it possible to record and stores the images digitally. The digital technique opened the field to new methods of medical X-ray imaging.

1.1 Digital image

A digital image is essentially a rectangular matrix. Each picture element in the matrix is referred to as a pixel. The brightness of each pixel is represented by an integer, known as the pixel value, and the range of possible integers is coded in a grey scale or in colours. The advantages of digital images compared to traditional X-ray film is that two images can be, for instance, subtracted or divided. A subtraction is done pixel by pixel for every matrix element in the two images. Division is working the same way, except that the pixel values are divided.

1.2 Medical imaging using an X-ray tube

Medical X-ray tubes use accelerating voltages in the range 40-150 kV and currents of up to 1300 mA for a single exposure and up to 2 mA for transillumination imaging. Tungsten anodes are the most frequently used to produce an X-ray energy spectrum up to 150 keV. Photons at energies lower than 30 keV do not contribute to the image formation, and therefore thin layers of aluminium of the order of a few millimetres are used as filters to reduce unnecessary patient exposure [1]. The X-ray imaging in medicine is performed using a shadow-recording technique. For example, bones absorb more of the emitted X-rays from the X-ray tube then tissue. Therefore bones will give a darker shadow than the tissue.

1.3 Differential imaging

Differential imaging is a new imaging technique involving important concepts as X-ray source, absorption of characteristic radiation and contrast medium. There are a few different methods, for example:
- Temporal subtraction
- Using the hard X-rays from a laser-produced plasma

There are two similarities in those techniques; a contrast medium is used and the images undergo image processing. The essential procedure for medical differential imaging is that one image is taken of a region of the body, and thereafter a contrast medium is injected in that region before a second image is taken (Temporal subtraction). In the second method mentioned above the

contrast medium is injected before the two images are taken. The two images are taken with two different X-ray sources and the contrast medium will absorb differently but the surrounding tissue will absorb about the same amount of the emitted radiation (see *section 1.3.2-4*).

The aim of the injected contrast medium is that it provides more absorption than the tissue. After the two images have been subtracted the resulting image would ideally only show the contrast medium in the region where it has been injected. Other tissue and bones have the same absorption in the two images and will therefore, ideally, disappear after a subtraction.

1.3.1 Temporal subtraction

An example of this method is when the arteries of two knees are examined [2]. First an image is recorded of the two knees without any contrast medium (see figure 1.1a). Next step is to inject the contrast medium into the arteries and take a second image (see figure 1.1b). The two images look identical. The differences in them are small and it is impossible to point out which image is with contrast medium and which is without contrast medium by the eye. Nevertheless, if the first image A is subtracted from the second image B, a third new image C is constructed (see figure 1.1c). In figure 1.1c all bones are gone and only the arteries are visible.
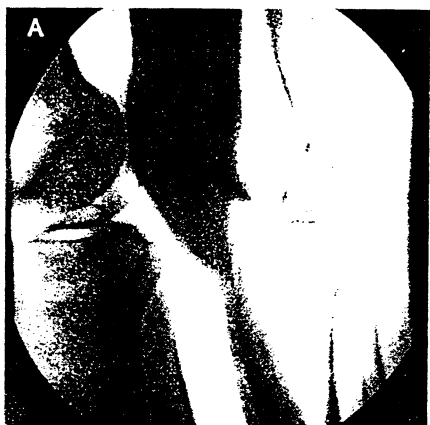

Figure 1.1a X-ray image of two knees with no contrast medium in the arteries.
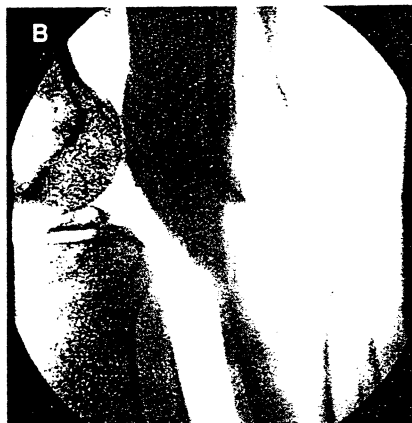

Figure 1.1b X-ray image of two knees with contrast medium injected in the arteries. This image looks the same by eye as the image in figure 1.1a. Small changes can be seen after image processing (see figure 1.1c).
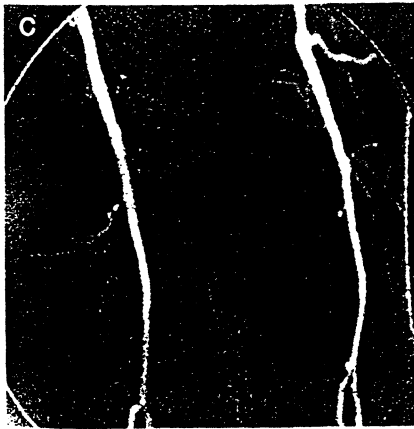
Figure 1.1c
The result after the subtraction of image A from image B. Only the arteries are visible and the bones and the tissue disappears. The bones and the tissue have the same absorption of X-rays in the two images and will therefore cancel out each other after a subtraction

1.3.2 Using the hard X-rays from a laser-produced plasma

In this method the X-ray source is a laser-produced plasma. A pulsed high power laser is focused onto a metal target. The target reaches very high temperature on the focal spot. At these temperatures the metal is transformed from its solid state to a plasma (see figure 1.2). Plasma is described as highly ionised atoms at very high temperature. This has been described as a fourth state of matter [3]. The plasma emits both continuous and characteristic radiation [4]. The continuous radiation is denoted as Bremsstrahlung. Bremsstrahlung originates from the interaction between the free electron and the charged nucleus in the plasma. An electron passing closely to the nucleus will decelerate (change direction) and emit a photon, because of electron-nucleus interaction (see figure 1.3). Depending on how close to the nucleus and what velocity the electron has, different energies are expected of the emitted photons. In the plasma an inverse process to Bremsstrahlung is

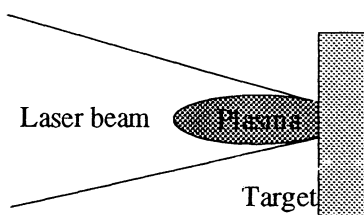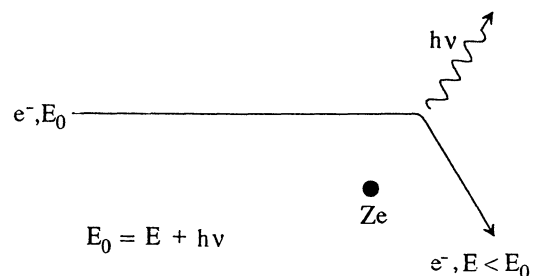*Figure 1.2 The laser light is focused onto the target and a plasma is generated.*

$$E_0 = E + h\nu$$

Figure 1.3 An electron with energy E_0 is deflected in the field of a nucleus and slowed down and a photon is emitted. This is the origin of X-ray Bremsstrahlung.

happening as well. Photons from the laser are absorbed by the ion-electron system and the free electron gain energy. Essentially the energy of the incoming photon is converted into an increase of kinetic energy for the free electron [4].

The energy states in an atom are discrete. The characteristic radiation is emitted when electrons from a high-energy state are transferred to a low-energy state. Every electron at the sub-level has a set of quantum numbers. Electron transitions must occur according to certain selection rules. The selection rules are derived from quantum mechanics [5-6]. The quantum number j denotes the spin-orbit coupling and the quantum number l denotes the orbital degeneracy. For these quantum numbers the allowed transitions for the electrons are [7]:

$$\Delta l = \pm 1 \tag{1}$$
$$\Delta j = 0, \pm 1 \tag{2}$$

The emitted radiation from the plasma depends on the atomic number of the target material. An increase of the atomic number will result in an increase of the energy of the emitted (characteristic) radiation, according to Mosely's law [8]:

$$\frac{1}{\sqrt{\lambda}} = \sqrt{\frac{E}{hc}} = C(Z - \sigma) \tag{3}$$

λ is the wavelength and E is the energy of a particular X-ray line, h is Planck's constant and c is the velocity of light in vacuum, and Z is the atomic number. Here C and σ are constants characterising a particular spectral series [8].
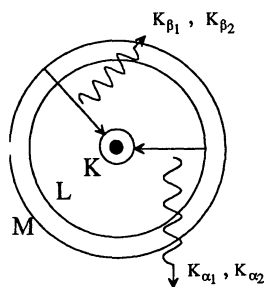

Figure 1.4 The place of an ionised electron in the K-shell is filled of another electron from the L- or the M-shell. The system looses energy and emits a photon.
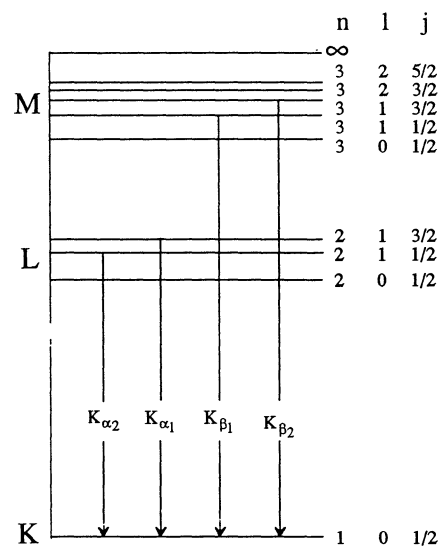

Figure 1.5 A general X-ray spectrum. The quantum numbers l and j is referred to as the orbital degeneracy and spin-orbit coupling respectively.

Target	K_{α_1} (keV)	K_{α_2} (keV)	K_{β_1} (keV)	K_{β_2} (keV)
Gadolinium	42.996	42.309	48.7	50.0
Tantalum	57.532	56.277	65.2	67.0

Table 1.1 The energy of the characteristic lines for gadolinium and tantalum [9].

High energetic X-rays, so-called hard X-rays (>10 keV), have a higher penetration in tissue than soft X-rays (0.1-1 keV). For example photons with the energy of 4 keV have a 1/e transmission of 0.2 mm in tissue and photons with the energy of 50 keV have a 1/e transmission of 24 cm in tissue. Gadolinium and tantalum both have a high atomic number, Z_{Gd} = 64 and Z_{Ta} = 73. As illustrated in figure 1.4 the characteristic lines K_{α_1}, K_{α_2}, K_{β_1} and K_{β_2} occur when an electron from the L- and M-shell fill the place from the ionised electron in the K-shell. The electron looses energy and emits a photon. The allowed transitions with highest energies are the characteristic lines K_{α_1}, K_{α_2}, K_{β_1} and K_{β_2}. Figure 1.5 illustrates a general energy diagram with the corresponding quantum numbers and the K_{α_1}, K_{α_2}, K_{β_1} and K_{β_2} transitions. The quantum energies for gadolinium and tantalum are listed in table 1.1 [9]. The modelled emission spectra from a gadolinium and a tantalum plasma are shown in figure 1.6a-b. The continuous radiation from the two targets are almost the same, because they have almost the same atomic numbers, Z_{Ta} ≈ Z_{Gd}, while the characteristic lines occur at significantly different energies. The intensity of the continuous radiation is proportional to the atomic number [7].
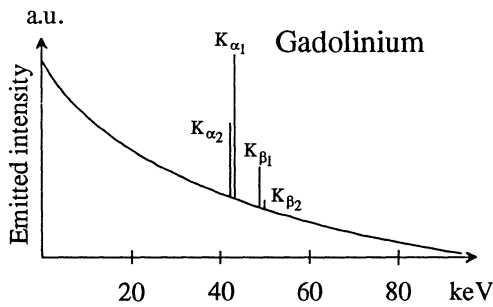
Figure 1.6a The modelled emission spectrum (Bremsstrahlung and the characteristic lines) for a gadolinium plasma.
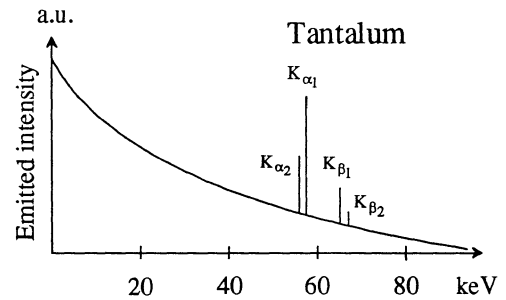
Figure 1.6b The modelled emission spectrum (Bremsstrahlung and the characteristic lines) for a tantalum plasma.

1.3.3 Absorption of continuous radiation

To reduce the low energetic continuous radiation, a filter can be used. Filters made of aluminium and copper have been chosen as filters in the experimental set-up. The transmitted intensity through a material depends exponentially on the thickness x, according to Beers-Lambert's law [7]:

$$I(Z,E) = I_0 \cdot e^{\mu(Z,E) \cdot x} \tag{4}$$

Where I_0 is the incident intensity and I is the transmitted intensity in the absorbing material. μ is the absorption coefficient, dependent on the atomic number of the material, Z, and E is the energy of the X-rays and where [7]:

$$\mu(Z, E) \cong \frac{Z^\alpha}{E^3} \quad \text{with} \quad 3 \leq \alpha \leq 4 \tag{5}$$

The absorption coefficients of aluminium and copper are plotted as functions of the photon energy in figure 1.7a-b [8]. Both absorption curves contain a strong so called K-absorption edge. The K edge correspond to the energy limit below which no ionisation of electrons from the K shell can occur. The ratio of the absorption just above and below the K-absorption edge is about 5-10. The position of the K edge for example copper is 8.98 keV means that the work of removing an electron from the K shell, where it experience the field of the nearly unshielded nuclear charge of the copper nucleus, is 8.98 keV. The modelled radiation intensity for a gadolinium and a tantalum plasma transmitted through a 3 mm aluminium filter, are shown in figure 1.8a-b. The low energy radiation is absorbed in the filter, because aluminium has a high absorption coefficient (μ) at low energies whereas it does not absorb at high energies, which is clearly shown in figure 1.7a. The Bremsstrahlung in figure 1.8a-b are assumed to be the same because gadolinium and tantalum have almost the same atomic number. It is only the

Figure 1.7a   The absorption coefficient of aluminium as a function of the photon energy. The K-absorption edge is at the energy 1.56 keV.
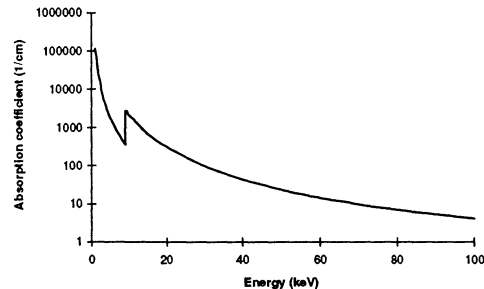
Figure 1.7b The absorption coefficient of copper as a function of the photon energy. The K-absorption edge is at the 8.98 keV.
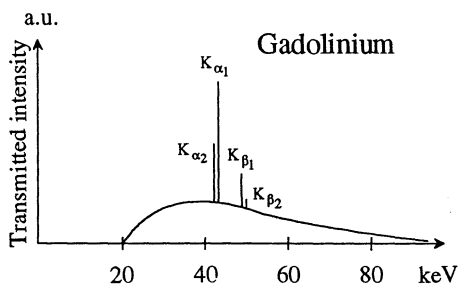
Figure 1.8a   The modelled spectrum for the emitted radiation from a gadolinium plasma after a 3 mm thick aluminium filter.
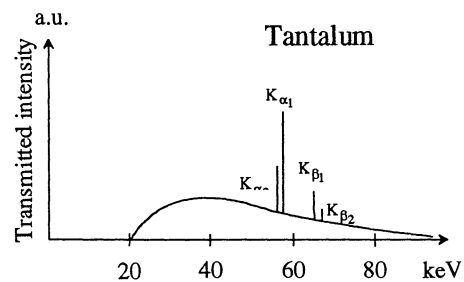
Figure 1.8b The modelled spectrum for the emitted radiation from a tantalum plasma after a 3 mm thick aluminium filter.

characteristic lines that comes at different energies. The characteristic lines of gadolinium are found at lower energies compared the characteristic lines of tantalum.

1.3.4 The contrast medium

The contrast medium is chosen so that it will absorb more of the emitted radiation from the gadolinium than from the tantalum plasma. A contrast medium containing a solution of gadolinium has its K-absorption edge at 50.2 keV [10] (see figure 1.9), that is between the characteristic line K_{β_2} of gadolinium and characteristic line K_{α_2} of tantalum. In figure 1.10a-b the modelled radiation intensity for a gadolinium and a tantalum plasma transmitted through a 3 mm aluminium filter and a gadolinium solution (contrast medium) are shown. When two images are taken, one with a gadolinium plasma and one with a tantalum plasma, they will differ where the gadolinium solution is. The characteristic lines from the gadolinium plasma are at lower energies than the K-absorption edge for the gadolinium solution (contrast medium) and will therefore not be absorbed as much in the contrast medium as the characteristic lines from the tantalum plasma. The characteristic lines of tantalum are at higher energies (above the K-absorption edge for the gadolinium solution) and will therefore be absorbed more in the contrast medium compared to the characteristic lines of gadolinium. The contrast medium will absorb the same amount of the Bremsstrahlung of the two targets. Therefore the integrated intensity for the transmitted radiation from the gadolinium source will be higher than that from the tantalum source.
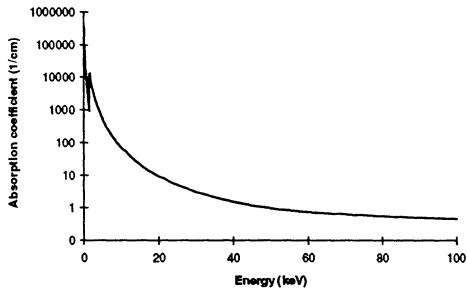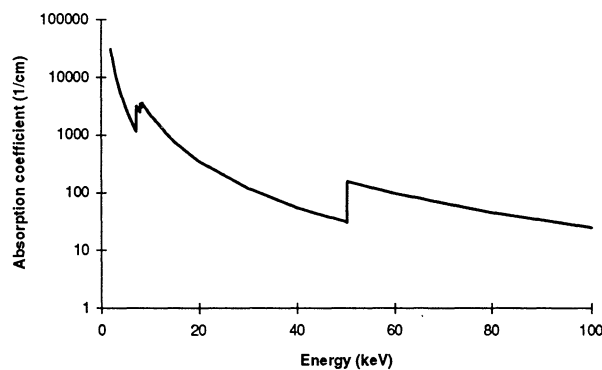

Figure 1.9 The absorption coefficient of gadolinium as a function of the photon energy. The K-absorption edge is at the energy 50.2 keV.
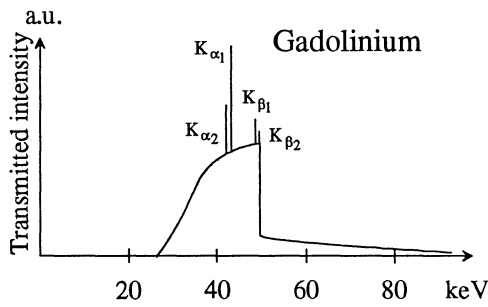
Figure 1.10a The modelled transmission spectrum for a gadolinium plasma after an aluminium filter and a gadolinium solution (contrast medium).
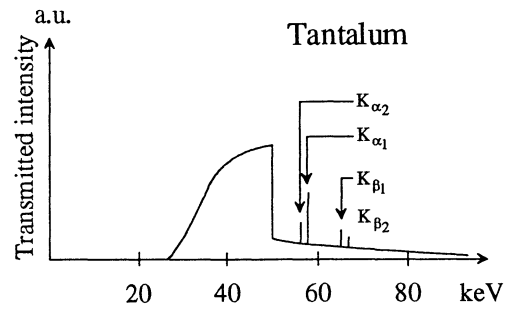
Figure 1.10b The modelled transmission spectrum for a tantalum plasma after an aluminium filter and a gadolinium solution (contrast medium).

1.4 Purpose of this project

For more than two years, experiments with laser-produced plasmas have been made at the Department of Atomic Physics at Lund Institute of Technology, Sweden. My part in this project was to:

- Make new test cells for test solutions.

- Try out suitable concentrations of gadolinium. Also to try out other substances with a K-absorption edge just above and below the K-absorption edge of gadolinium to investigate the element selectivity of differential imaging.

- Inject solutions into biological samples, in order to investigate whether the differential imaging technique is working on tissue or not.

- Develop a computer program that makes sure that the two images spatially overlap each other before a subtraction or a division of the images is made.

2 Experimental set-up and operation

To produce the plasma, a high power laser is used. The laser beam is focused onto a metal target in a vacuum chamber. The laser-produced plasma generates X-rays that are used for imaging. The images are *recorded* on image plates.

2.1 Laser system

The laser system which has been used is a terawatt (10^{12} W) laser [11]. It is based on chirped pulse amplification in titanium doped sapphire. The laser generates radiation tuneable in the 760-840 nm region with a maximum power of 1.5 TW in 150 fs ($150 \cdot 10^{-15}$ s) pulses with a repetition rate of 10 Hz.

2.2 X-ray equipment

The laser beam is reflected by gold-coated mirrors and continues into the vacuum chamber through a glass window. In the chamber the laser beam is focused by a gold-coated off-axis parabolic focusing mirror onto the target. A pressure of 20-30 Torr is maintained in the chamber to prevent the target material from coating the optics [12-14]. Still, small particles from the plasma sputter the focusing mirror. A thin piece of glass, a pellicle, is inserted between the mirror and the target to prevent destruction of the mirror. The focusing mirror has a focal length of 50 mm. The size of the focal spot is a few micrometers in diameter and the intensity on that spot is about 10^{18} W/cm^2. To make sure that the next laser shot will hit a fresh spot, the target is mounted on a rotating disc. The target can also be translated sideways, which allows usage of the full target area. The generated X-ray continues through another window (number 7 in figure 2.1). It is important that is made of a material that this is transparent to hard X-rays. Therefore a plastic film has been used as (X-ray) window material.

Samples, containing for instance a solution of gadolinium, a filter and an image plate is inserted in front of the "X-ray window" (see figure 2.1).

Image plate (12)

Lead brick (11)

Filter (8)

X-rays (9)

Samples (10)

Laser (1)

4

6

7

5

3

2
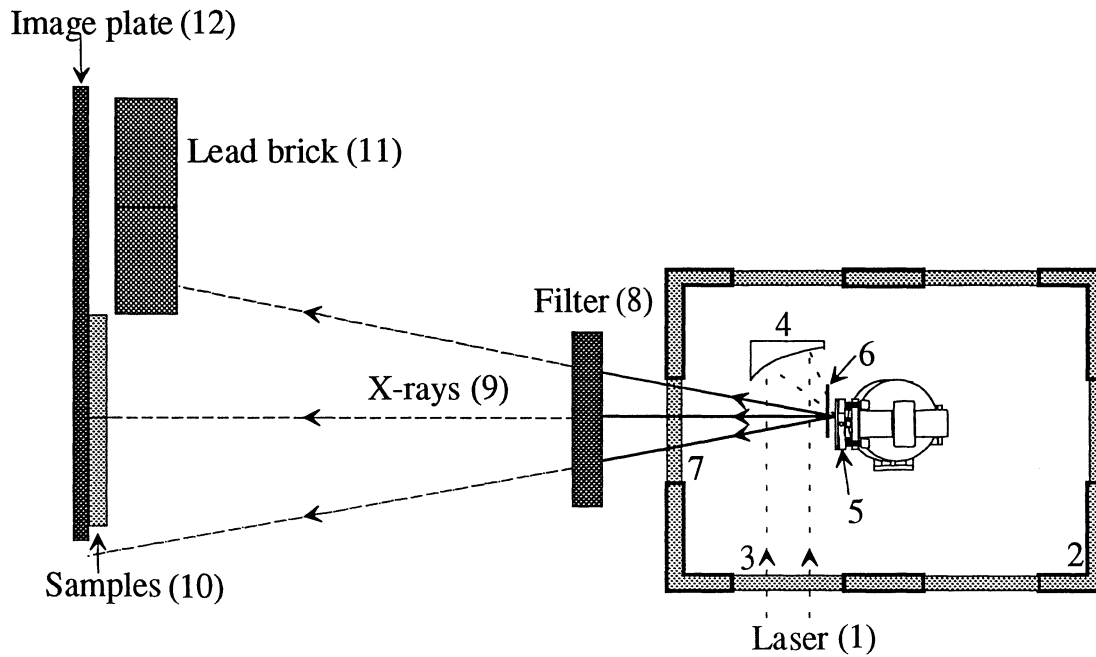
Figure 2.1 The experimental set-up for X-ray generation from a laser-produced plasma. The laser beam (1) enters the vacuum chamber (2) through a glass window (3). The beam is focused by a gold coated focusing mirror (4) onto the rotating target (5). A thin piece of glass (6) is put between the target and the mirror to prevent destruction of the mirror from sputtering particles from the plasma. The emitted X-rays from the laser-produced plasma continues through another window (7). The low energy Bremsstrahlung is absorbed by a filter (8) and the high energy X-rays (9) continues to the samples (10). Lead bricks (11) are used to protect the second half of the image plate (12), in the set-up only 20×20 cm of the image plate is used at one exposure. At the second exposure the target is changed and the image plate flipped over and the second half is exposed.

2.3 Image plates

The image system used in the experiments is a Fuji Film computed radiography (CR) system [15]. The image plate exists in a number of sizes. In the experimental set-up 40 × 20 cm plates have been used. The image plate is composed of three layers. The first layer is to protect the second layer, that contains X-ray absorbing crystals (with a grain size of about 5 μm), and the third layer is to support the image plate (see figure 2.2). The photo-stimulate crystals in the plates are barium fluorinebromine together with a small amount of europium that works as a luminescence centre. To read an image plate, after having been exposed, a He-Ne laser (λ=633 nm) is used. The photons from the laser stimulate the crystals and they emit radiation. This is called photostimulated luminescence (PSL). The PSL will be collected by a light collection guide into a photomultiplier tube and converted to electric signals. The signals are converted to digital signals of 8 to 12 bits [13]. In the experimental set-up the signals have been converted into digital signals of 8 bits so the images processing can be done in PC software. This means that the digital signals will vary from 0 to 255. In a grey scale the pixel value 0 is equivalent to black (no radiation from the crystals) and the pixel value 255

equivalent to white (maximum radiation from the crystals). The resolution of the image limited by the optics and the mechanism in the scanner. The resolution can be set to either 100 μm or 200 μm.

←——Protective layer (10 μm)

←——Photo-stimulate
phosphor layer (150 μm)

←——Support (250 μm)
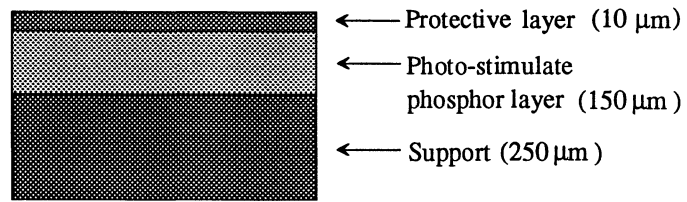
Figure 2.2 A cross-section of Fuji Film image plate [13].

3 Test Cell Array

The Test Cell Array, TCA, is used to for evaluation of contrast media. A good contrast medium is a metal salt, which have a high atomic number of the metal compound and which is easily dissolvable in water. Different metal salts have been tried and the results are discussed in *section 4*.

3.1 The advantages of a TCA

The TCA presents an easy way to optimise the absorption of X-rays in a solution. Using the test cells to find a suitable solution allows by-pass of usage on biological samples, such as rats, before success is predicted. The TCAs are made of plexiglass which has almost the same absorption coefficient as tissue and water.

3.2 The development of the TCA

The test cells used in an earlier stage of the project were made of plastic straws, sealed with epoxy glue at top and bottom together with a plexiglass frame. The straws were put together in rows of twelve, standing perpendicular to the radiation flux. The straw tubes have to be put in a container of water during exposure. Then the water absorption in the cylindrical tubes is the same. This is done to make sure that the recorded images only show the absorption of the metal salt. Also, leakage from the straw set-up was frequent even though many layers of glue were applied. This demanded a new type of cell construction. The new test cells should be improved in three areas:

- uniform X-ray exposure thickness of the test cell
- simple preparation and filling
- liquid proof cell

The size of the old test cells was determined by the size of the image plate. Full usage of the sensitive area of the image plate is wanted mainly because
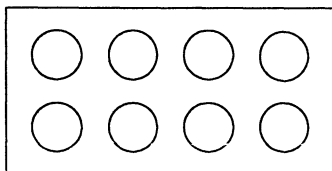
Figure 3.1   A test cell array with round holes
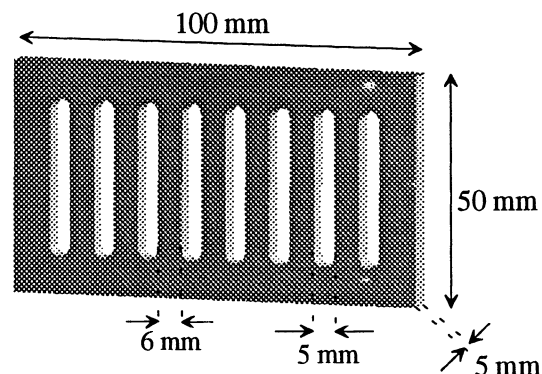
Figure 3.2   A test cell array with oblong holes. The separation between the cells has to be large enough to prevent scattered photons from the other cells.

18

comparison of many solutions developed under the same conditions (same exposure) are desirable. The expense of developing the image plates also requests an optimised use of area. The image plates are 20 x 40 cm. The old test cells were 10 x 5 cm and sixteen test cells could be exposed on the same plate. A TCA was made in the same rectangular size as the old test cells.

To fulfil the first request three flat pieces of plexiglass were used. One piece (the middle piece) was chosen thicker (5 mm) than the other two (the top and the bottom piece, 1 mm thick). First, round holes were drilled out in the middle piece. The holes were 15 mm in diameter and eight holes fitted nicely (see figure 3.1). Unfortunately, there were a few problems with the drilling. Drill sticks that did not ruin the edges of the holes had to be found. The solution was a specially designed drill with a carving-knife on top. However,

Figure 3.3 The mould and the three plexiglass plates that makes a TCA. The top piece is used only when force is applied.

another problem was induced by this drill. It created an edge above the surface when the drill stick got through the plexiglass. Since the surface smoothness is important when the pieces are assembled, this drill was not satisfactory either. Agreeable holes with sharp edges can be milled with high speed cutting tools. Instead of making round holes, oblong holes were cut in the plates (see figure 3.2).

The next step in the process of making a TCA is to find the best way to apply the bottom and the top on the middle piece. A mould is used when a TCA is assembled (see figure 3.3). To start with, green and red caramel colours were used as test solutions. The green and red colour makes it easy to detect any eventual leaking in the cells. Four kinds of glue were tried out to assemble the TCA. The glues were:

- Acrafix 106
- Epoxy
- Perma Bond C6C
- Silicon all-round, manufactured by Bostik

All the glues showed some crucial drawback. Some of the cells could keep the liquid, but often with an unfortunate dissolving of glue into the cells. The glue could interfere with the results taken later. Another method of TCA construction had to be found. Different kinds of tapes were then tried. Tape shows many advantages in comparison to glue. No unknown substance like glue will interfere with the solution in the cell, moreover the mounting time is no longer important. A tape was found which satisfied the criteria of simple preparation and filling as well as the demand of a liquid proof cell. The tape is TESA, fabricated by Beirsdorf AG Hamburg, Germany. To make sure the tape will stick tightly on the plexiglass, alcohol is used to clean the surfaces.

The newly developed TCA together with the tape used in the manufacturing will fulfil the demands listed above.
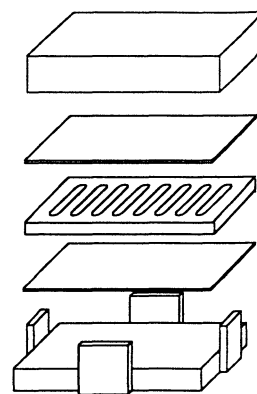
4 Solutions in TCA

In *Appendix A*, data for eight TCAs are tabulated. The metal salts are heavy metals with a high atomic number, bound to lighter atoms. It is only those heavy metals that are of interest. Heavy atoms have a higher absorption coefficient than lighter atoms. That means that it is only the metal compound in the salt that is able to absorb hard X-rays. The lighter compounds will be transparent at those high energies. The absorption edge of metals and their chemical compound in the salt are listed in *Appendix A* and every element in the salt with their atomic number and the K-absorption edge are listed in table 4.1 [8].

Name	Symbol	Atomic number	K edge (keV)
Hydrogen	H	1	0.013
Carbon	C	6	0.283
Nitrogen	N	7	0.402
Oxygen	O	8	0.532
Phosphorus	P	15	2.146
Sulphur	S	16	2.472
Chlorine	Cl	17	2.822
Cesium	Cs	55	35.984
Barium	Ba	56	37.440
Cerium	Ce	58	40.443
Gadolinium	Gd	64	50.239
Ytterbium	Yb	70	61.332
Tungsten	W	74	69.525
Lead	Pb	82	88.004
Thorium	Th	90	109.650

Table 4.1 The atomic number and the K-absorption edge for all the chemical compound in the salts.

In *Appendix B* the absorption coefficients are plotted as a function of the photon energy for the metals and their lighter compounds in the salt [10]. It is only gadolinium that has its K-absorption edge between the characteristic line of gadolinium K_{β_2} (50.0 keV) and the characteristic line of tantalum K_{α_2} (56.3 keV). The other metals have their absorption edge below and above the characteristic lines for gadolinium and tantalum. In figure 4.1a-b, two images are recorded, one with gadolinium and one with tantalum as a target. To filter the X-ray radiation a 0.30 mm copper and 1.00 mm aluminium filters have been used. The exposure time was 15 minutes. It is not possible to see any difference by eye between the two. The images were "overlapped" and subtracted. The subtraction is made in Publisher's Paintbrush (version 2.00, ZSoft Corporation). Before the subtraction, an offset level has to be set. This offset level is introduced to prevent negative pixel values, which otherwise could be a problem. It can be put from 0 to 255, which is the range that the pixel values can have (see *section 2.3*). The offset is set to 100. The subtracted image looks grey and no contrasts are visible. The pixel values in the image are between 96 and 104, which is a small range in comparison to the full pixel

range, 0 to 255. In Publisher's Paintbrush it is possible to expand the 96-104 to full pixel value range. This is done in figure 4.2.
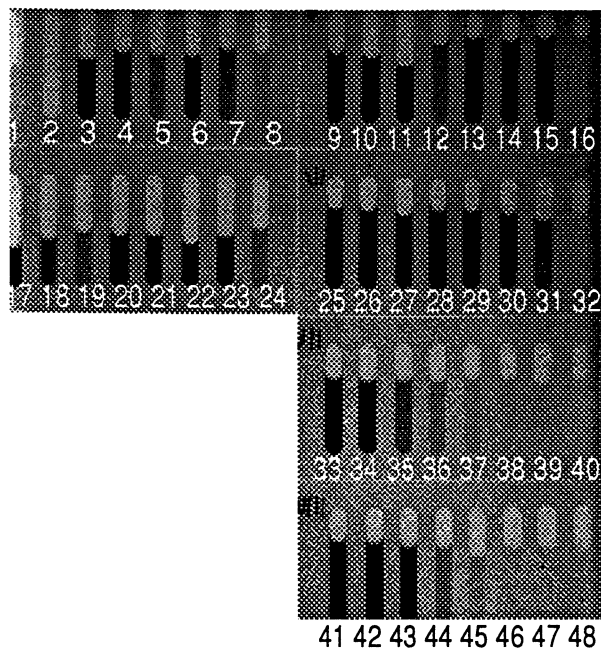
Figure 4.1a   TCAs with different absorbing solutions and concentrations in the cells. Gadolinium is in this image used as a target. The TCAs are listed in Appendix A.
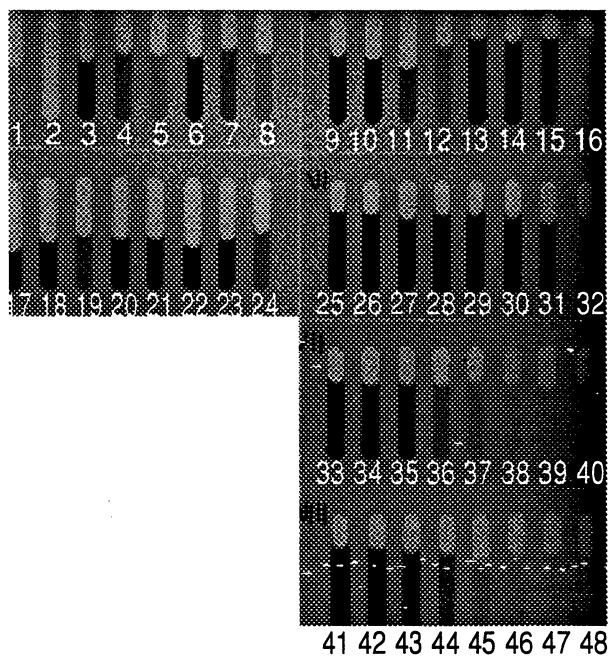
Figure 4.1b   Same TCAs as in figure 4.1a. Tantalum is here used as a target.
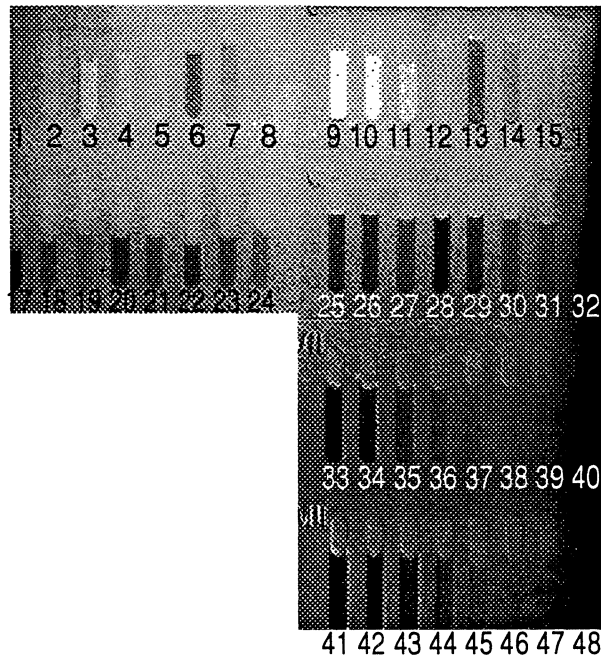
Figure 4.2   The interesting range of pixel values is expanded from 96 - 104 to 0 - 255. The TCAs and their cells are now visible. The cells numbered 2-5 and 9-12 are containing a gadolinium solution. Cell number 9 is brightest because it has the highest gadolinium concentration.

The TCAs in figure 4 are listed in table 4.2. The cells with numbers 3-5 and 9-12 are cells with a solution of a gadolinium salt. In figure 4.2, cell number 9 and 10 are brightest, because the gadolinium concentrations are highest in these two cells. The brightness in figure 4.2 is depended on how many photons have transmitted through the cells. A bright cell has transmitted more photons than a dark cell. Cells containing a solution of gadolinium are transmitting more photons when gadolinium is used as a target than when tantalum is used as a target. Therefore are they brighter when the two images are subtracted. The other cells containing a metal solution with a K-absorption edge above or below the characteristic lines from the emitted radiation of the two targets are looking the same. If the two images were subtracted in reverse order the cells containing a solution of gadolinium would be darker than the rest of the cells. It is interesting to note, for example, that the cell numbers 25 and 34 look the same even though they are containing cerium and lead solutions respectively. Lead has a higher absorption coefficient than cerium (see *Appendix B*). The transmissions for a few cells are listed in table 4.3. This demonstration verified, in a controlled manner, that differential imaging was working in practice. The next step was to try differential imaging in biological samples.

Cell number	TCA number
1 - 8	3
9 - 16	5
17 - 24	4
25 - 32	6
33 - 40	7
41 - 48	8

Table 4.2 The TCA numbers refers to Appendix A.

Cell	Metal	Metal conc. (mole/dm^3)	Transmission, Gd-target (%)	Transmission, Ta-target (%)
3	Gd	1.0	70	65
9	Gd	2.27	49	42
10	Gd	1.14	67	60
11	Gd	0.57	81	75
6	Ce	1.0	67	75
7	Ce	0.5	81	87
25	Ce	2.28	39	47
26	Ce	1.14	56	63
17	Pb	1.0	58	72
18	Pb	0.5	88	78
33	Pb	1.48	44	52
34	Pb	0.74	65	72

Table 4.3 The transmission in the cells in figure 4.1a-b. The table shows clearly that the transmission in the gadolinium solution is higher when gadolinium is used as a target. In the cells containing solutions of cerium and lead have a higher transmission when tantalum is used as a target.

5 Solutions injected in biological samples

Two sacrificed rats were prepared by injecting gadolinium and cerium solution into their stomachs. The filter in the X-ray generation set-up was 0.45 mm copper. The images in figure 5.1a-b are spatially overlapped according to *section 6*. In figure 5.2 they are subtracted with an offset of 100. The range is "expanded" from 90-115 to 0-255. It is easy to see the stomach with the gadolinium solution in figure 5.2. The tube that is attached to the syringe and goes to the stomach is also visible. The stomach injected with a cerium solution is visible as well, but it is not as bright as the stomach injected with gadolinium.

Figure 5.1a The rat to the left is injected with cerium and the one to the right is injected with gadolinium. Gadolinium has here been used as target.

Figure 5.1b Same objects as in figure 5.1a. Tantalum has here been used as target.
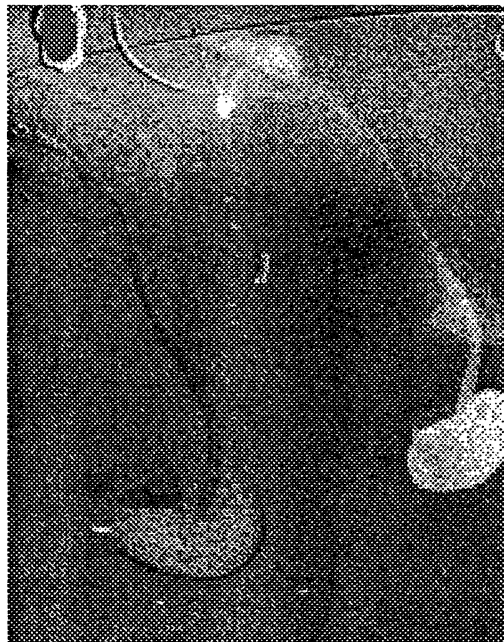
Figure 5.2 The stomach to the right is injected with gadolinium and the one to the left with cerium. The rat injected with gadolinium is brighter because of the K-absorption edge of the gadolinium solution is between the two targets emitted characteristic lines.

6 Shift and rotation of images

To get an accurate subtraction of two images they have to be spatially overlapped. Three problems can occur. (1) The image plate is not placed at the same distance from the object at the two exposures. That is a magnification problem. It is easy to prevent this if the set-up is rigid and it is impossible for the plate to move in the direction from the target when the plate is changed for the second exposure. (2) The object is not placed on the same sidewise position. Before comparison there might be a need to shift the images sidewise to give the object in them the same pixel co-ordinates. (3) The images are rotated. The cassette holding the image plates during exposure is slightly larger than the image plate itself. When the image plate is changed or if the cassette is flipped over, the image plate is dislodged from its position in the cassette. A Pascal program has been written, that can compensate for the last two problems (see *Appendix D*). In *Appendix* C is a description of the procedures in the program and of which parameters are input respectively output data. The

Figure 6.1   The co-ordinates (x_1, y_1) and (x_2, y_2) are input data to the program. The Pascal program in Appendix D is written that the two sets of co-ordinates in the figure make a rectangle (dotted line) and are stored as a matrix in the program.

program needs two references marks on the object as input data (see figure 6.1). A schematic description of how two images are spatially overlapped is illustrated in figure 6.2a-d.
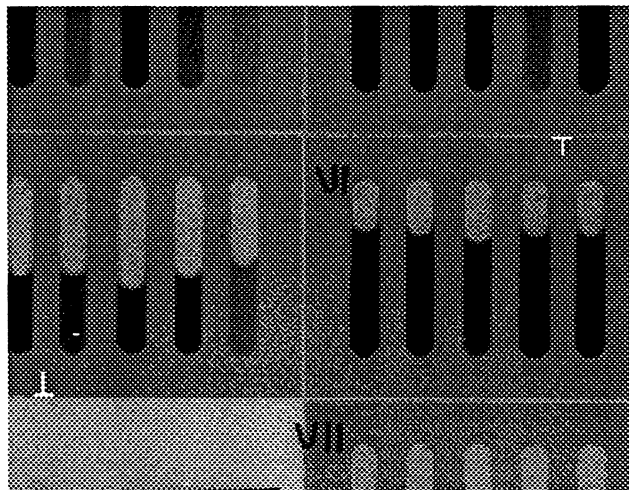
Figure 6.2a   A TCA with solutions is the first image. The reference marks are clearly shown in the figure. The co-ordinates for these reference marks are input data in the Pascal program in Appendix D.
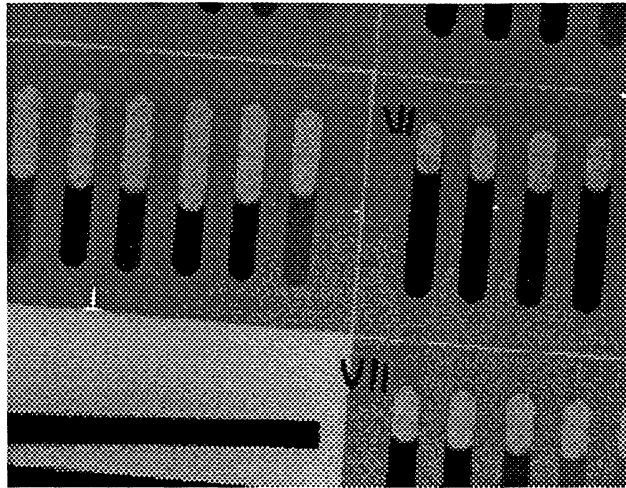
Figure 6.2b A TCA with solutions in the second image. The reference marks are also here clearly shown. The image is a little tilted and has to be rotated. The image has also to be shifted so the two images can be spatially overlapped.
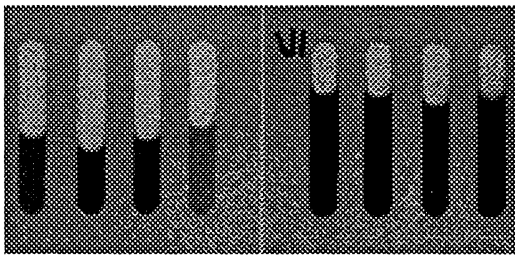
Figure 6.2c The first image is cut out and is now ready for image processing.
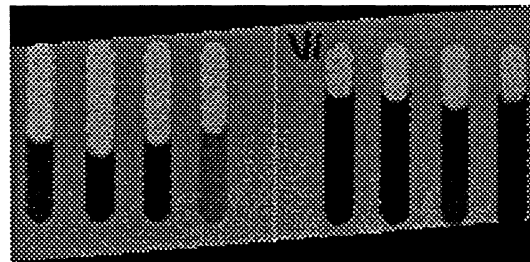
Figure 6.2d The second image is cut out and rotated to spatially overlap the first cut out image. The black area around the image is set to the pixel value 0 (black) in the Pascal program in Appendix D because there are no information about that area when the image is cut out in figure 6.2b.

The images in figure 6.2a-b are now spatially overlapped and ready to be subtracted or divided.

6.1 Software

The computer program that makes the shifting and rotation of images is written in Borland Pascal 7.0 for Windows environment. The algorithm is shown schematically in figure 6.3.
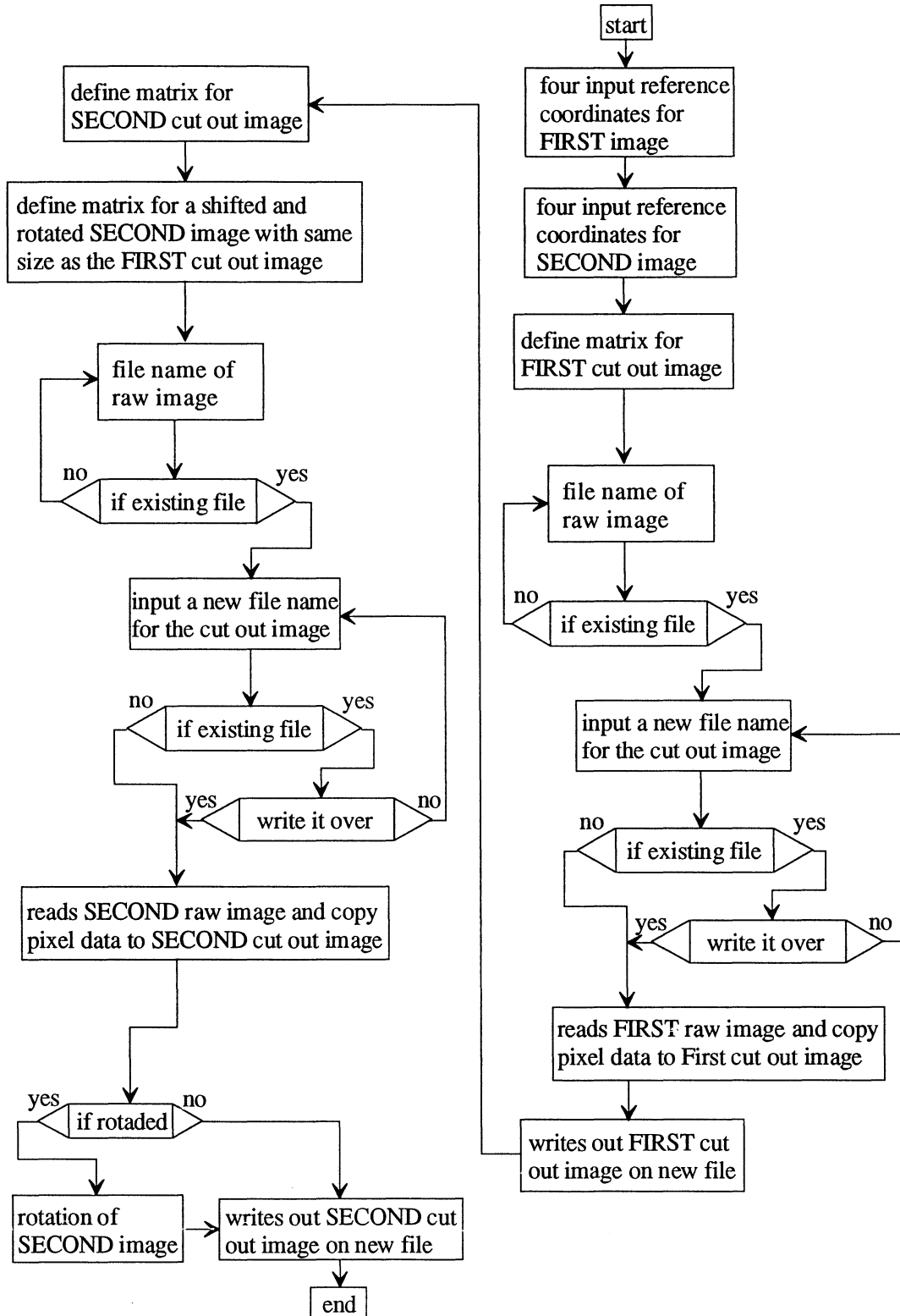
Figure 6.3 The algorithm for the program that makes sure that the two images are spatially overlapped before image processing.

27

7 Conclusion

Differential imaging using hard X-rays from a laser-produced plasma is a new imaging technique and it has shown initial success. The method is working with the described set-up in *section 2*. However, it required high concentrations of the gadolinium solutions to show any differential effect.

7.1 Future and related work

To be able to see any differential effect the gadolinium solution had to have a high concentration. Lower concentrations can be achieved if the relation characteristic X-rays to Bremsstrahlung is improved. Presently, the hard X-ray production varies since the target leaves the exact focus position for the laser during rotation of the target disc. At high laser intensity, characteristic radiation from K_α and K_β are high in intensity. If the focusing onto the target is optimised, an increase in the characteristic and continuous radiation ratio is expected. That would lead to that shorter exposure times can be used. A position control for laser focusing conditions is under development and when a stable X-ray production is achieved, the concentration of gadolinium can be decreased [16]. The reason X-ray tubes are not used is that it is assumed that the relative intensity of characteristic emission lines is stronger for laser-produced plasma compared to conventional X-ray tubes.

Procedures can easily be implemented in the program in *Appendix D*, for subtraction or division of two images. That operation is then not needed to be made in Publisher's Paintbrush.

8 Acknowledgements

I would like to thank my supervisors Ian Mercer and Carl Tillman for all their help through the project. I would also like to thank Matthias Grätz who helped me with software problems and Kockums Marine AB for letting me use their workshop. Finally, I would like to thank Sune Svanberg and all the people, not mentioned by name, at the Department of Atomic Physics.

Appendix A

TCA no. 1			
Cell	Substance	Metal	Metal concentration (mole/dm^3)
1	Air		
2	Water		
3	GdCl$_3$·6H$_2$O	Gd	0.036
4	GdCl$_3$·6H$_2$O	Gd	0.018
5	H$_2$WO$_4$	W	0.036
6	H$_2$WO$_4$	W	0.018
7	(NH$_4$)$_4$Ce(SO$_4$)$_4$·2H$_2$O	Ce	0.036
8	(NH$_4$)$_4$Ce(SO$_4$)$_4$·2H$_2$O	Ce	0.018

TCA no. 2			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Air		
2	Water		
3	GdCl$_3$·6H$_2$O	Gd	0.637
4	GdCl$_3$·6H$_2$O	Gd	0.468
5	BaCl$_2$·2H$_2$O	Ba	0.637
6	BaCl$_2$·2H$_2$O	Ba	0.3185
7	GdCl$_3$·6H$_2$O	Gd	0.3185
8	Water		

TCA no. 3			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Air		
2	Water		
3	GdCl$_3$·6H$_2$O	Gd	1.0
4	GdCl$_3$·6H$_2$O	Gd	0.5
5	GdCl$_3$·6H$_2$O	Gd	0.25
6	CeCl$_3$·7H$_2$O	Ce	0.5
7	CeCl$_3$·7H$_2$O	Ce	0.25
8	CeCl$_3$·7H$_2$O	Ce	0.125

TCA no. 4			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Pb(NO$_3$)$_2$	Pb	1.0
2	Pb(NO$_3$)$_2$	Pb	0.5
3	Pb(NO$_3$)$_2$	Pb	0.25
4	BaCl$_2$·2H$_2$O	Ba	1.0
5	BaCl$_2$·2H$_2$O	Ba	0.5
6	CsCl	Cs	1.0
7	CsCl	Cs	0.5
8	CsCl	Cs	0.25

TCA no. 5			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Gd(NO$_3$)$_3$·6H$_2$O	Gd	2.27
2	Gd(NO$_3$)$_3$·6H$_2$O	Gd	1.14
3	Gd(NO$_3$)$_3$·6H$_2$O	Gd	0.57
4	Gd(NO$_3$)$_3$·6H$_2$O	Gd	0.28
5	Yb(NO$_3$)$_3$·5H$_2$O	Yb	1.7
6	Yb(NO$_3$)$_3$·5H$_2$O	Yb	0.85
7	Yb(NO$_3$)$_3$·5H$_2$O	Yb	0.43
8	Water		

TCA no. 6			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Ce(NO$_3$)$_3$·6H$_2$O	Ce	2.28
2	Ce(NO$_3$)$_3$·6H$_2$O	Ce	1.14
3	Ce(NO$_3$)$_3$·6H$_2$O	Ce	0.57
4	H$_3$[P(W$_3$O$_{10}$)$_4$]·X H$_2$O	W	3.33
5	H$_3$[P(W$_3$O$_{10}$)$_4$]·X H$_2$O	W	1.67
6	H$_3$[P(W$_3$O$_{10}$)$_4$]·X H$_2$O	W	0.83
7	H$_3$[P(W$_3$O$_{10}$)$_4$]·X H$_2$O	W	0.42
8	Water		

TCA no. 7			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Pb(NO$_3$)$_2$	Pb	1.48
2	Pb(NO$_3$)$_2$	Pb	0.74
3	Pb(NO$_3$)$_2$	Pb	0.37
4	Pb(NO$_3$)$_2$	Pb	0.19
5	Pb(NO$_3$)$_2$	Pb	0.093
6	Pb(NO$_3$)$_2$	Pb	0.046
7	Pb(NO$_3$)$_2$	Pb	0.023
8	Water		

TCA no. 8			
Cell	Substances	Metal	Metal concentration (mole/dm^3)
1	Th(NO$_3$)$_4$·5H$_2$O	Th	2.0
2	Th(NO$_3$)$_4$·5H$_2$O	Th	1.0
3	Th(NO$_3$)$_4$·5H$_2$O	Th	0.5
4	Th(NO$_3$)$_4$·5H$_2$O	Th	0.25
5	Th(NO$_3$)$_4$·5H$_2$O	Th	0.125
6	Th(NO$_3$)$_4$·5H$_2$O	Th	0.062
7	Th(NO$_3$)$_4$·5H$_2$O	Th	0.031
8	Water		

Appendix B

Hydrogen, K edge at 0.013 keV
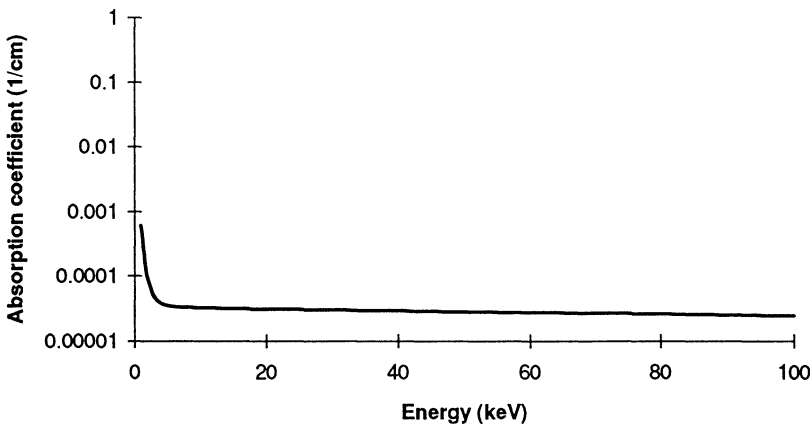
Carbon, K edge at 0.283 keV
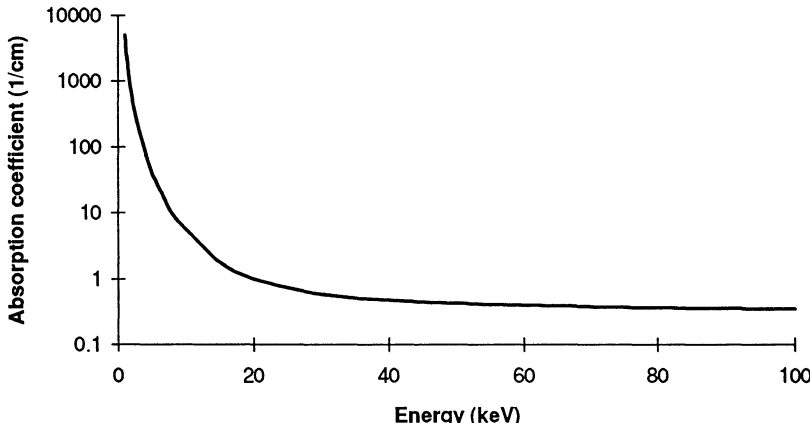
Nitrogen, K edge at 0.402 keV
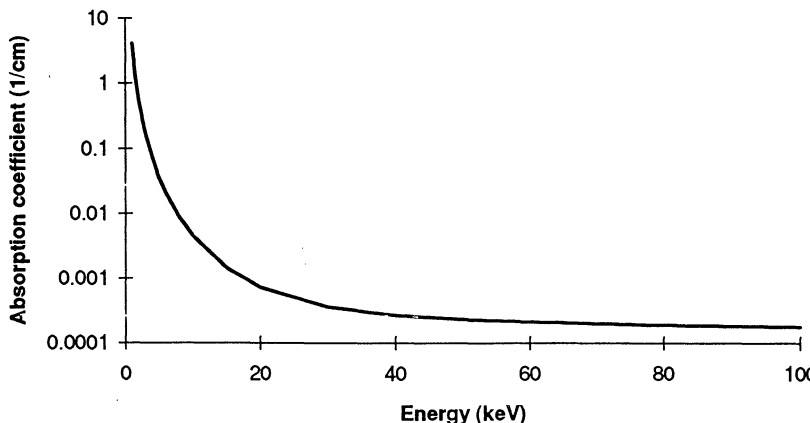
Oxygen, K edge at 0.532 keV
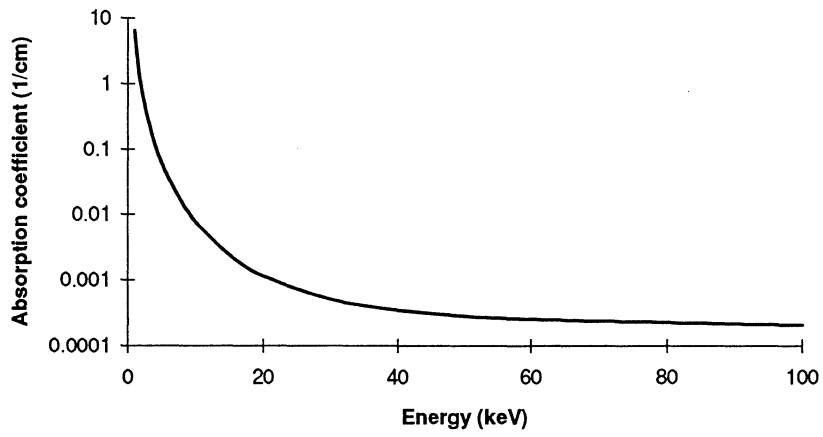
Phosphorus, K edge at 2.146 keV
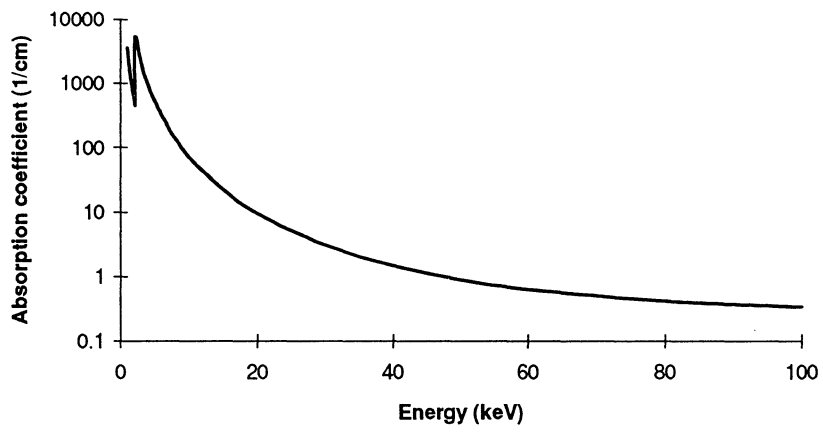
Sulfur, K edge at 2.472 keV
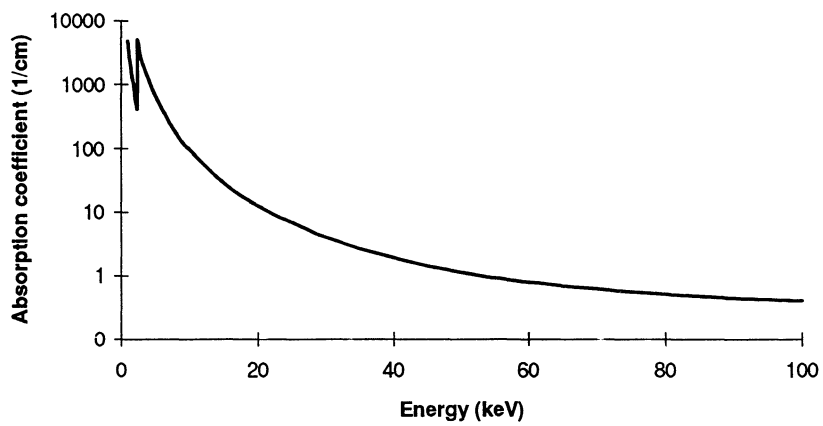
Chlorine, K edge at 2.822 keV

Cesium, K edge at 35.98 keV

Barium, K edge at 37.44 keV

Cerium, K edge at 40.44 keV

Gadolinium, K edge at 50.24 keV

Ytterbium, K edge at 61.33 keV

35

Tungsten, K edge at 69.53 keV

Lead, K edge at 88.00 keV

Thorium, K edge at 109.7 keV

Appendix C

First = true

ReadSize(CoordFirstImage,CoordSecondImage)
Reads the co-ordinates for the two images.

CoordFirstImage
CoordSecondImage

DefineMatrix(CoordFirstImage,CutOutImageMatrix1)
Defines a matrix for the pixel values in the first cut out image.

CutOutImageMatrix1

FileNameRawImage(DirIn1)
Reads the file name of the first image.

DirIn1

FileNameCutOutImage(DirOut1)
Reads the new file name for the new cut out first image.

DirOut1

ReadRawImage(DirIn1,CutOutImageMatrix1,BufferFirstImage,CoordFirstImage,ImageInFile1)
Reads the first cut out image. The BMP-data is stored in the vector BufferFirstImage and the image (pixel values) is stored in the matrix CutOutImageMatrix1. ImageInFile1 is assign to DirIn1.

WriteCutOutImage(DirOut1,CutOutImageMatrix1,COIM1,BufferFirstImage, CoordFirstImage, CoordSecondImage,First,ImageOutFile1)
The first cut out image is written out to the new file DirOut1. ImageOutFile1 is assigned to DirOut1. The variabels COIM1, CoordFirstImage and CoordSecondImage is of no interest here. However the same procedure is used for the second image and the variabels are of great importence there.

First = false

DefineMatrix(CoordSecondImage,CutOutImageMatrix2)
Defines a matrix for the pixel values in the second cut out image.

CutOutImageMatrix2

DefineMatrix(CoordFirstImage,COIM3)
Defines a matrix big as the matrix for the first cut out image.

COIM3

FileNameRawImage(DirIn2)
Reads the file name of the second image.

DirIn2

FileNameCutOutImage(DirOut2)
Reads the new file name for the new cut out second image.

DirOut2

ReadRawImage(DirIn2,CutOutImageMatrix2,BufferSecondImage,CoordSecondImage,ImageInFile2)
Reads the second cut out image. The BMP-data is stored in the vector BufferSecondImage and the image (pixel values) is stored in the matrix CutOutImageMatrix2. ImageInFile2 is assign to DirIn2.

WriteCutOutImage(DirOut2,CutOutImageMatrix2,COIM3,BufferSecondImage, CoordFirstImage, CoordSecondImage,First,ImageOutFile2)
The second cut out image is written out to the new file DirOut2. ImageOutFile2 is assigned to DirOut2. The co-ordinates for the two images are used to calculate how much the second image is rotated compare to the first image. The matrix element in the CutOutImageMatrix2 is shifting position to satisfy the positioning of the matrix element in the first image. The shifted elements are then put in the matrix COIM3, which has the same size as the first image matrix.

37

Appendix D

```pascal
program   Image(infile,outfile,input,output);

(*----------------------------------------------------------------------------------------------------- *)
(*                                                                                                       *)
(*Author       Christer Andersson                                                                        *)
(*             Master paper work at the Department of Atomic Physics at Lund Institute *)
(*             of Technology, Sweden.                                                                     *)
(*Date         March, 1995                                                                               *)
(*Comments  This program reads first a selected piece of the FIRST image and writes it  *)
(*             out to a new file. There after the SECOND selected image piece are read    *)
(*             and rotated before it is written out to a new file.                                       *)
(*                                                                                                       *)
(*----------------------------------------------------------------------------------------------------- *)


uses    crt,dos,graph,WinAPI;

const   CharConst=20;
        ImageSize =1000;
        BufferSize =42;

type    BMPHEADTYPE = record
            BFTYPE           :string[2];
            BFSIZE           :longint;
            BFRESERVED1   :word;
            BFRESERVED2   :word;
            BFOFFBITS        :longint;
        end;
        BMIHEADTYPE = record
            BISIZE                   :longint;
            BIWIDTH                  :longint;
            BIHEIGHT                 :longint;
            BIPLANES                 :word;
            BIBITCOUNT               :word;
            BICOMPRESSION            :longint;
            BISIZEIMAGE              :longint;
            BIXPELSPERMETER :longint;
            BIYPELSPERMETER :longint;
            BICLRUSED                :longint;
            BICLRIMPORTANT :longint;
        end;
        RGBQTYPE = record
            RGBBLUE        :byte;
            RGBGREEN       :byte;
            RGBRED         :byte;
            RGBRESERVED  :byte;
        end;
        XCol = array[1..ImageSize] of byte;
        ImageMatrixType = array[1..ImageSize] of ^XCol;
        IMT = ImageMatrixType;
        BufferType = array[0..BufferSize] of byte;
        Coord = array[1..4] of integer;
        IFT = file of byte;      (*IFT=ImageFileType*)


var     DirIn1,DirIn2,DirOut1,DirOut2                        :string;
        CutOutImageMatrix1,CutOutImageMatrix2    :ImageMatrixType;
        COIM1,COIM3                                          :ImageMatrixType;
        PixelValue                                           :byte;
        BMIHEAD                                              :BMIHEADTYPE;
```

38

```
    BMPHEAD                                                    :BMPHEADTYPE;
    RGBQ                                                       :array[0..255] of RGBQTYPE;
    ImageInFile1,ImageInFile2,ImageOutFile1,ImageOutFile2      :IFT;
    BufferFirstImage,BufferSecondImage                         :BufferType;
    CoordFirstImage,CoordSecondImage                           :Coord;
    ChString                                                   :string[2];
    i,x1,x2,y1,y2,xx1,xx2,yy1,yy2,Elementx,Elementy            :longint;
    First,FileExist                                            :boolean;
    a,N                                                        :char;


procedure ReadSize(var CoordFirstImage,CoordSecondImage:Coord);
(*Reads the co-ordinates which limit the cut out images*)
begin
    writeln('The co-ordinates for the images have to be taking as follows: ');
    writeln;
    writeln('  _____  ');
    writeln(' |                                            | ');
    writeln(' |                                            | ');
    writeln(' |                        *(x2,y2)            | ');
    writeln(' |                                            | ');
    writeln(' |                                            | ');
    writeln(' |                                            | ');
    writeln(' |                                            | ');
    writeln(' |            *(x1,y1)                        | ');
    writeln(' |_____| ');
    writeln;
    writeln('Write the co-ordinates for the FIRST cut out image: ');
    writeln;
    write('x1: ');
    readln(CoordFirstImage[1]);
    write('y1: ');
    readln(CoordFirstImage[2]);
    write('x2: ');
    readln(CoordFirstImage[3]);
    write('y2: ');
    readln(CoordFirstImage[4]);
    writeln;
    writeln('Write the co-ordinates for the SECOND cut out image: ');
    writeln;
    write('x1: ');
    readln(CoordSecondImage[1]);
    write('y1: ');
    readln(CoordSecondImage[2]);
    write('x2: ');
    readln(CoordSecondImage[3]);
    write('y2: ');
    readln(CoordSecondImage[4]);
end; (*ReadSize*)




procedure DefineMatrix(CoordImage:Coord;var CutOutImageMatrix:ImageMatrixType);
(*In this procedure are all the elements in the cut out image are set to zero.*)
(*I have been taking in concern that the matrix have to be even divided by four*)
var     xconst,yconst,i,j,xplus,xmod:integer;
begin
    xconst:=CoordImage[3]-CoordImage[1]+1;     (*x2-x1+1*)
    yconst:=CoordImage[2]-CoordImage[4]+1;     (*y1-y2+1*)
    if xconst mod 4 = 0 then
        xplus:=0
    else if xconst mod 4 = 1 then
        xplus:=3
```

39

```
        else if xconst mod 4 = 2 then
            xplus:=2
        else
            xplus:=1;
        for i:=1 to yconst do
            CutOutImageMatrix[i]:=GlobalAllocPtr(gmem_Moveable+gmem_ZeroInit,xconst+xplus);
    end; (*DefineMatrix*)




function FileTest(Dir:string):boolean;
(*FileTest is used to make sure to not*)
(*over write a already existing file. *)
var     F:file;
begin
    {$I-}
    assign(F,Dir);
    reset(F);
    {$I+};
    if IOResult = 0 then
        FileTest:=true
    else
        FileTest:=false;
end;




procedure FileNameRawImage(var DirIn:string);
(*The directory for the file which are going to be read*)
var D:string;
begin
    writeln('Write the file and the directory for the image you will READ (include *.bmp): ');
    writeln;
    readln(D);
    DirIn:=D;
end; (*FileNameRawImage*)




procedure FileNameCutOutImage(var DirOut:string);
(*The directory for the file which the new cut out image are going to be stored*)
var D:string;
begin
    writeln('Write the file name and directory for the NEW cut out image (include *.bmp): ');
    writeln;
    readln(D);
    DirOut:=D;
end;  (*FileNameCutOutImage*)




procedure ReadRawImage(Dir:String;var COIM:ImageMatrixType;var Buffer:BufferType;
                        CoordI:Coord ;var ImageInFile:IFT);
(*Reads all information in the raw image*)
(*COIM=CutOutImageMatrix, CoordI=CoordImage*)
var     Byte0,Byte1,Byte2,Byte3                          :byte;
        Element,Elementx,Elementy,xconst,Ex,xmod,xplus,C32BL    :longint;
        (* C32BL=Const32BytesLong *)
begin
    assign(ImageInFile,Dir);
    reset(ImageInFile);
    C32BL:=256;
    with BMPHEAD do
```

40

```
(*Are reading 14 bytes*)
begin
    read(ImageInFile,Buffer[1],Buffer[0]);
    BFTYPE[1]:=chr(Buffer[1]);
    BFTYPE[2]:=chr(Buffer[0]);
    read(ImageInFile,Byte3,Byte2,Byte1,Byte0);
    (*BFSIZE:=Byte0*C32BL*C32BL*C32BL+Byte1*C32BL*C32BL+Byte2*C32BL+Byte3; *)
    read(ImageInFile,Buffer[2],Buffer[3]);
    BFRESERVED1:=Buffer[2]*C32BL+Buffer[3];
    read(ImageInFile,Buffer[4],Buffer[5]);
    BFRESERVED2:=Buffer[4]*C32BL+Buffer[5];
    read(ImageInFile,Buffer[6],Buffer[7],Buffer[8],Buffer[9]);
    BFOFFBITS:=Buffer[9]*C32BL*C32BL*C32BL + Buffer[8]*C32BL*C32BL + Buffer[7]*C32BL +
            Buffer[6];
end;
with BMIHEAD do
(*Are reading 40 bytes*)
begin
    read(ImageInFile,Buffer[10],Buffer[11],Buffer[12],Buffer[13]);
    BISIZE:=Buffer[13]*C32BL*C32BL*C32BL + Buffer[12]*C32BL*C32BL + Buffer[11]*C32BL +
            Buffer[10];
    read(ImageInFile,Byte3,Byte2,Byte1,Byte0);
    BIWIDTH:=Byte0*C32BL*C32BL*C32BL+Byte1*C32BL*C32BL+Byte2*C32BL+Byte3;
    read(ImageInFile,Byte3,Byte2,Byte1,Byte0);
    BIHEIGHT:=Byte0*C32BL*C32BL*C32BL+Byte1*C32BL*C32BL+Byte2*C32BL+Byte3;
    read(ImageInFile,Buffer[14],Buffer[15]);
    BIPLANES:=Buffer[15]*C32BL+Buffer[14];
    read(ImageInFile,Buffer[16],Buffer[17]);
    BIBITCOUNT:=Buffer[17]*C32BL+Buffer[16];
    read(ImageInFile,Buffer[18],Buffer[19],Buffer[20],Buffer[21]);
    BICOMPRESSION:=Buffer[21]*C32BL*C32BL*C32BL + Buffer[20]*C32BL*C32BL +
                    Buffer[19]*C32BL + Buffer[18];
    read(ImageInFile,Byte3,Byte2,Byte1,Byte0);
    BISIZEIMAGE:=Byte0*C32BL*C32BL*C32BL+Byte1*C32BL*C32BL+Byte2*C32BL+Byte3;
    read(ImageInFile,Buffer[22],Buffer[23],Buffer[24],Buffer[25]);
    BIXPELSPERMETER:=Buffer[25]*C32BL*C32BL*C32BL + Buffer[24]*C32BL*C32BL +
                    Buffer[23]*C32BL + Buffer[22];
    read(ImageInFile,Buffer[26],Buffer[27],Buffer[28],Buffer[29]);
    BIYPELSPERMETER:=Buffer[29]*C32BL*C32BL*C32BL + Buffer[28]*C32BL*C32BL +
                    Buffer[27]*C32BL + Buffer[26];
    read(ImageInFile,Buffer[30],Buffer[31],Buffer[32],Buffer[33]);
    BICLRUSED:=Buffer[33]*C32BL*C32BL*C32BL + Buffer[32]*C32BL*C32BL + Buffer[31]*C32BL
                    + Buffer[30];
    read(ImageInFile,Buffer[34],Buffer[35],Buffer[36],Buffer[37]);
    BICLRIMPORTANT:=Buffer[37]*C32BL*C32BL*C32BL + Buffer[36]*C32BL*C32BL +
                    Buffer[35]*C32BL + Buffer[34];
end;
for i:=0 to 255 do
(*Are reading 1024 bytes*)
begin
    read(ImageInFile,Buffer[38],Buffer[39],Buffer[40],Buffer[41]);
    with RGBQ[i] do
    begin
        RGBBLUE:=Buffer[38];
        RGBGREEN:=Buffer[39];
        RGBRED:=Buffer[40];
        RGBRESERVED:=Buffer[41];
    end;
end;
x1:=CoordI[1];
y1:=CoordI[2];
x2:=CoordI[3];
y2:=CoordI[4];
```

```
writeln('The co-ordinates are: ',x1:5,y1:5,x2:5,y2:5);
with BMIHEAD do
begin
    if BIWIDTH mod 4 = 0 then
        xplus:=0
    else if BIWIDTH mod 4 = 1 then
        xplus:=3
    else if BIWIDTH mod 4 = 2 then
        xplus:=2
    else
        xplus:=1;
    if x1 > 0 then
    begin
        for Element:=1 to ((BIWIDTH+xplus)*(BIHEIGHT-(y1+1))+x1) do
        begin
            read(ImageInFile,Pixelvalue);
        end;
    end;
    for Elementy:=1 to (y1-y2+1) do
    begin
        for Elementx:=1 to (x2-x1+1) do
        begin
            read(ImageInFile,PixelValue);
            COIM[Elementy]^[Elementx]:=PixelValue;
        end;
        for Element:=1 to BIWIDTH-(x2+1)+xplus+x1 do
            read(ImageInFile,Pixelvalue);
    end;
end;
end;  (*ReadRawImage*)




procedure WriteCutOutImage(DirO:string;var COIM,COIM2:IMT; Buffer:BufferType; CFI,CSI:Coord;
                                     F:boolean;var IOF:IFT);
(*Writing out the cut out image in a new file*)
(*DirO=DirOut, COIM=CutOutImage, CFI=CoordFirstImage*)
(*CSI=CoordSecondImage, F=First, IOF=ImageOutFile*)
var     Byte0,Byte1,Byte2,Byte3,B,BNoll      :byte;
        xconst,yconst,Ex,xmod,xplus,C32BL     :longint;
        t                                     :THandle;

    procedure Rotation(CoordImage1,CoordImage2:Coord;COIM,COIM2:ImageMatrixType);
    var     NoRotation,NotClockWR,ClockWR                        :boolean;
            xnew,ynew,sizex,sizey,sizexx,sizeyy,Row,Col,PixelSum,zeros    :longint;
            Fi1,Fi2,phi,alpha,xtemp,ytemp,r                     :extended;
            x,y                                                 :longint;
    begin
        NoRotation:=false;
        NotClockWR:=false;
        ClockWR:=false;
        x1:=CoordImage1[1];
        y1:=CoordImage1[2];
        x2:=CoordImage1[3];
        y2:=CoordImage1[4];
        xx1:=CoordImage2[1];
        yy1:=CoordImage2[2];
        xx2:=CoordImage2[3];
        yy2:=CoordImage2[4];
        if x1-x2=0 then
            Fi1:=pi/2
        else
            Fi1:=arctan((y2-y1)/(x1-x2));
```

```
if xx1-xx2=0 then
    Fi2:=pi/2
else
    Fi2:=arctan((yy2-yy1)/(xx1-xx2));
phi:=abs(Fi1-Fi2);
if Fi1 = Fi2 then
begin
    writeln('The SECOND image is rotated 0 degrees.');
    NoRotation:=true;
end
else if Fi1 > Fi2 then
begin
    writeln('The SECOND image is rotated ',phi*180/pi,' degrees counter clockwise');
    NotClockWR:=true;
end
else
begin
    writeln('The SECOND image is rotated ',phi*180/pi,' degrees clockwise');
    ClockWR:=true;
end;
writeln('phi = ',phi);
writeln('Fi1 = ',Fi1);
writeln('Fi2 = ',Fi2);
if ClockWR then
begin
    sizex:=x2-x1+1;
    sizey:=y1-y2+1;
    sizexx:=xx2-xx1+1;
    sizeyy:=yy1-yy2+1;
    for Elementy:=1 to sizeyy do
    begin
        for Elementx:=1 to sizexx do
        begin
            x:=Elementx-1;
            y:=Elementy-1;
            r:=sqrt(x*x+y*y);
            if (x=0) and (y=0) then
                alpha:=0
            else if (x=0) and (y<>0) then
                alpha:=pi/2
            else
                alpha:=arctan(y/x);
            xtemp:=r*cos(alpha-phi);
            ytemp:=r*sin(alpha-phi);
            if (xtemp>=0.0) and (ytemp>=-0.5) and (xtemp+0.5<=sizex) and
                    (ytemp+0.5<=sizey) then
            begin
                if ytemp = -0.5 then
                    ytemp:=0
                else
                    ytemp:=abs(ytemp);
                xnew:=round(xtemp);
                ynew:=round(ytemp);
                COIM2[ynew+1]^[xnew+1]:=COIM[Elementy]^[Elementx];
            end;
        end;
    end;
    for Elementy:=1 to sizey do
        t:=GlobalFreePtr(COIM[Elementy]);
end;
if NotClockWR then
begin
    sizex:=x2-x1+1;
```

43

```pascal
sizey:=y1-y2+1;
sizexx:=xx2-xx1+1;
sizeyy:=yy1-yy2+1;
for Elementy:=1 to sizeyy do
begin
    for Elementx:=1 to sizexx do
    begin
        x:=Elementx-1;
        y:=Elementy-1;
        r:=sqrt(x*x+y*y);
        if (x=0) and (y=0) then
            alpha:=0
        else if (x=0) and (y<>0) then
            alpha:=pi/2
        else
            alpha:=arctan(y/x);
        xtemp:=r*cos(alpha+phi);
        ytemp:=r*sin(alpha+phi);
        if (xtemp>=0.0) and (ytemp>=-0.5) and (xtemp+0.5<=sizex) and
                (ytemp+0.5<=sizey) then
        begin
            if ytemp = -0.5 then
                ytemp:=0
            else
                ytemp:=abs(ytemp);
            xnew:=round(xtemp);
            ynew:=round(ytemp);
            COIM2[ynew+1]^[xnew+1]:=COIM[Elementy]^[Elementx];
        end;
    end;
end;
    for Elementy:=1 to sizey do
        t:=GlobalFreePtr(COIM[Elementy]);
end;
if NoRotation then
    COIM2:=COIM;
xconst:=x2-x1+1;
yconst:=y1-y2+1;
writeln('xconst=',xconst,' yconst=',yconst);
if xconst mod 4 = 0 then
    xplus:=0
else if xconst mod 4 = 1 then
    xplus:=3
else if xconst mod 4 = 2 then
    xplus:=2
else
    xplus:=1;
for Elementy:= 1 to yconst do
begin
    for Elementx:= 1 to xconst do
    begin
        PixelValue:=COIM2[Elementy]^[Elementx];
        if PixelValue = 0 then
        begin
            if (Elementy>1)and(Elementy<yconst)and(Elementx>1)and(Elementx<xconst)
                    then
            begin
                x:=Elementx;
                y:=Elementy;
                if (COIM2[y]^[x-1]=0) or (COIM2[y]^[x+1]=0) or (COIM2[y+1]^[x]=0) or
                        (COIM2[y-1]^[x]=0) then
                    PixelValue:=0
                else
```

44

```pascal
                        PixelValue:=COIM2[y]^[x-1];
                end;
            end;
            write(IOF,PixelValue);
        end;
        if xplus <> 0 then
        begin
            for Elementx:=1 to xplus do
            begin
                PixelValue:=0;
                write(IOF,PixelValue);
            end;
        end;
    end;
    for Elementy:=1 to sizey do
    t:=GlobalFreePtr(COIM2[Elementy]);   (* The memory for the matrix gets free *)
end;  (*Rotation*)


begin
    assign(IOF,DirO);
    rewrite(IOF);
    x1:=CFI[1];
    y1:=CFI[2];
    x2:=CFI[3];
    y2:=CFI[4];
    xx1:=CSI[1];
    yy1:=CSI[2];
    xx2:=CSI[3];
    yy2:=CSI[4];
    with BMPHEAD do
    (*Are writing 14 bytes*)
    begin
        Buffer[1]:=ord(BFTYPE[1]);
        Buffer[0]:=ord(BFTYPE[2]);
        write(IOF,Buffer[1],Buffer[0]);
        xconst:=x2-x1+1;
        if xconst mod 4 = 0 then
            xplus:=0
        else if xconst mod 4 = 1 then
            xplus:=3
        else if xconst mod 4 = 2 then
            xplus:=2
        else
            xplus:=1;
        BFSIZE:=(x2-x1+1+xplus)*(y1-y2+1)+40+4*256+14;  (*BFSIZE are changed*)
        Byte3:=BFSIZE mod 256;
        Byte2:=(BFSIZE div 256) mod 256;
        Byte1:=(Byte2 div 256) mod 256;
        Byte0:=(Byte1 div 256) mod 256;
        write(IOF,Byte3,Byte2,Byte1,Byte0);

        write(IOF,Buffer[2],Buffer[3]);                   (*BFRESERVED1 are not changed*)
        write(IOF,Buffer[4],Buffer[5]);                   (*BFRESERVED2 are not changed*)
        write(IOF,Buffer[6],Buffer[7],Buffer[8],Buffer[9]);   (*BFOFFBITS are not changed*)
    end;
    with BMIHEAD do
    (*Are writing 40 bytes*)
    begin
        write(IOF,Buffer[10],Buffer[11],Buffer[12],Buffer[13]);  (*BISIZE are not changed*)

        BIWIDTH:=(x2-x1+1);                               (*BIWIDTH are changed*)
        Byte3:=BIWIDTH mod 256;
```

45

```
Byte2:=(BIWIDTH div 256) mod 256;
Byte1:=(Byte2 div 256) mod 256;
Byte0:=(Byte1 div 256) mod 256;
write(IOF,Byte3,Byte2,Byte1,Byte0);

BIHEIGHT:=(y1-y2+1);                                    (*BIHEIGHT are changed*)
Byte3:=BIHEIGHT mod 256;
Byte2:=(BIHEIGHT div 256)mod 256;
Byte1:=(Byte2 div 256) mod 256;
Byte0:=(Byte1 div 256) mod 256;
write(IOF,Byte3,Byte2,Byte1,Byte0);

write(IOF,Buffer[14],Buffer[15]);                       (*BIPLANES are not changed*)
write(IOF,Buffer[16],Buffer[17]);                       (*BIBITCOUNT are not changed*)
write(IOF,Buffer[18],Buffer[19],Buffer[20],Buffer[21]); (*BICOMPRESSION are not changed*)

xconst:=x2-x1+1;
if xconst mod 4 = 0 then
     xplus:=0
else if xconst mod 4 = 1 then
     xplus:=3
else if xconst mod 4 = 2 then
     xplus:=2
else
     xplus:=1;
BISIZEIMAGE:=(x2-x1+1+xplus)*(y1-y2+1);                 (*BISIZEIMAGE are changed*)
Byte3:=BISIZEIMAGE mod 256;
Byte2:=(BISIZEIMAGE div 256) mod 256;
Byte1:=(Byte2 div 256) mod 256;
Byte0:=(Byte1 div 256) mod 256;
write(IOF,Byte3,Byte2,Byte1,Byte0);

write(IOF,Buffer[22],Buffer[23],Buffer[24],Buffer[25]); (*BIXELSPERMETER are not changed*)
write(IOF,Buffer[26],Buffer[27],Buffer[28],Buffer[29]); (*BIYELSPERMETER are not changed*)
write(IOF,Buffer[30],Buffer[31],Buffer[32],Buffer[33]); (*BICLRUSED are not changed*)
write(IOF,Buffer[34],Buffer[35],Buffer[36],Buffer[37]); (*BICLRIMPORTANT are not changed*)
end;
BNoll:=0;
for B:=0 to 255 do      (*RGBBLUE,RGBGREEN,RGBRED;RGBRESERVED are not changed*)
begin
     with RGBQ[B] do
     (*Are writing 1024 bytes*)
     begin
          write(IOF,B,B,B,BNoll);
     end;
end;

if not F then                    (*If First image, write out it with no shifting or rotation*)
     Rotation(CoordFirstImage,CoordSecondImage,CutOutImageMatrix2,COIM3)
else
begin
     xconst:=x2-x1+1;
     yconst:=y1-y2+1;
     if xconst mod 4 = 0 then
          xplus:=0
     else if xconst mod 4 = 1 then
          xplus:=3
     else if xconst mod 4 = 2 then
          xplus:=2
     else
          xplus:=1;
     for Elementy:= 1 to yconst do
```

```
        begin
            for Elementx:= 1 to xconst do
            begin
                PixelValue:=COIM[Elementy]^[Elementx];
                write(IOF,Pixelvalue);
            end;
            if xplus <> 0 then
            begin
                for Elementx:=1 to xplus do
                begin
                    PixelValue:=0;
                    write(IOF,PixelValue);
                end;
            end;
        end;
    end;
    for Elementy:=1 to yconst do
    t:=GlobalFreePtr(COIM[Elementy]);
end; (*WriteCutOutImage*)




begin (*Main program*)
    First:=true;                              (*The first image is taking care of now*)
    ReadSize(CoordFirstImage,CoordSecondImage);
    DefineMatrix(CoordFirstImage,CutOutImageMatrix1);
    writeln;
    writeln('Following data is about the FIRST image! ');
    FileNameRawImage(DirIn1);
    FileExist:=FileTest(DirIn1);
    while not FileExist do
    begin
        writeln('The file ',DirIn1,' does not exist. Try again!');
        FileNameRawImage(DirIn1);
        FileExist:=FileTest(DirIn1);
    end;
    FileNameCutOutImage(DirOut1);
    FileExist:=FileTest(DirOut1);
    while FileExist do
    begin
        writeln('The out put file ',DirOut1,' is an existing file.');
        writeln('Do you want to  write it over (Y/N)?');
        readln(a);
        if (a = 'n') or (a = 'N') then
        begin
            FileNameCutOutImage(DirOut1);
            FileExist:=FileTest(DirOut1);
        end
        else if (a = 'y') or (a = 'Y') then
            FileExist:=false
        else
            FileExist:=true;
    end;
    ReadRawImage(DirIn1,CutOutImageMatrix1,BufferFirstImage,CoordFirstImage,ImageInFile1);
    WriteCutOutImage(DirOut1,CutOutImageMatrix1,COIM1,BufferFirstImage,CoordFirstImage,
                    CoordSecondImage,First,ImageOutFile1);
    writeln;
    First:=false;                  (*The second image is taking care of now*)
    DefineMatrix(CoordSecondImage,CutOutImageMatrix2);
    DefineMatrix(CoordFirstImage,COIM3);
    writeln;
    writeln('Following data is about the SECOND image! ');
    FileNameRawImage(DirIn2);
```

```
FileExist:=FileTest(DirIn2);
while not FileExist do
begin
    writeln('The file ',DirIn2,' does not exist. Try again!');
    FileNameRawImage(DirIn2);
    FileExist:=FileTest(DirIn2);
end;
FileNameCutOutImage(DirOut2);
FileExist:=FileTest(DirOut2);
while FileExist do
begin
    writeln('The out put file ',DirOut2,' is an existing file.');
    writeln('Do you want to  write it over (Y/N)?');
    readln(a);
    if (a = 'n') or (a = 'N') then
    begin
        FileNameCutOutImage(DirOut1);
        FileExist:=FileTest(DirOut1);
    end
    else if (a = 'y') or (a = 'Y') then
        FileExist:=false
    else
        FileExist:=true;
end;
ReadRawImage(DirIn2,CutOutImageMatrix2,BufferSecondImage,CoordSecondImage,
            ImageInFile2);
WriteCutOutImage(DirOut2,CutOutImageMatrix2,COIM3,BufferSecondImage,CoordFirstImage,
            CoordSecondImage,First,ImageOutFile2);
writeln;
writeln('The program is over. The two cut out images are in the');
writeln('files ',DirOut1,' and ',DirOut2, '.');
end.
```

References

[1] K.P. Maher and J.F. Malone, *Digital fluoroscopy: a new development in medical imaging,* Contemp. Phys, vol. 27, NO. 6, 533-552, 1989.

[2] W.R. Brody, *Digital Radiography,* Raven Press, New York, 1984.

[3] H.J. Gray and A. Isaacs, *A New Dictionary of Physics,*
Longman Group Limited, London, 1975.

[4] P.K. Carroll and E.T. Kennedy, *Laser-Produced Plasmas,*
Contemp. Phys, 1981, vol. 22, NO. 1, 61-96.

[5] E. Merzbacher, *Quantum mechanics,*
Second edition, Wiley International Edition, New York, 1970.

[6] G. Ohlén, *Quantum mechanics 2,*
Paper, Lund Institute of Technology.

[7] H. Haken and H.C. Wolf, *Atomic and Quantum Physics,*
Springer-Verlag, Berlin, 1987.

[8] S. Svanberg, *Atomic and Molecular Spectroscopy,*
Springer-Verlag, New York, 1992.

[9] R.C. Weast (ed.), *Handbook of Chemistry and Physics,*
54th edition 1973-1974, CRC Press

[10] WM.J. Veigele, *Photon cross sections from 0.1 keV to 1 MeV for elements Z = 1 to Z = 94.* Atomic Data, vol. 5, NO. 1, 1973.

[11] S. Svanberg, J. Larsson, A. Persson and C.-G. Wahlström,
Lund High-Power Laser Facility - Systems and First Results,
Physica Scripta, vol. 49, 187-197, 1994.

[12] J.D. Kmetec, *Ultrafast laser generation of hard X-rays,*
Thesis, Stanford University, 1992.

[13] K. Herrlin, G. Svahn, C. Olsson, H. Petersson, C. Tillman, A. Persson,
C.-G. Wahlström and S. Svanberg,
Generation of X-rays for medical imaging by high-power lasers: Preliminary results, Radiology, vol. 189, 1993.

[14] C. Tillman, A. Persson, C.-G. Wahlström and S. Svanberg,
Imaging using hard X-rays from a laser-produced plasma,
Submitted to Applied Physics B, October 1994.

[15] J. Miyahara, *The imaging plate: A new radiation image sensor*, Chemistry Today, October 1989, page 29-36.

[16] C. Lindheimer, *Optimising hard X-ray generation from laser-produced Plasma*, LRAP-162, 1995.