Presentation of a Virtual Architectural Model Considering New Multimedia Standards in HTML5

Günter Pomaska

Faculty of Architecture and Civil Engineering, University of Applied Science,

Bielefeld, Germany, gp@imagefact.de

Abstract: During the 20th century tens of millions of inhabitants of Europe became victims of flight, expulsion and forced migration. Loss of home land is a painful experience for all effected people and an evidence for the madness of war. A large number of initiatives indicate that displaced people need to document and remember their roots. One of those activities among dozens of others tells about the modern history of the former German Gumbinnen.

1945 the town was almost utterly destroyed by war damage. Today descendants of the former residents and the present residents are interested in the past. The remembrance is stored in a collection of items including a large database of photographs and a wooden architectural model. Exhibiting the items in a museum provides limited access only. The web is the appropriate media for distributing this kind of information to the targeted audience.

The focus on the present contribution lies on digitizing the model applying photogrammetric methods and designing a web user interface with the new multimedia possibilities of HTML5. Modern browsers execute different graphics formats native. SVG, the video tag and the canvas element with its 2D context enable high quality graphics without plug-ins. WebGL is part of a web browser that renders 3D content hardware accelerated. Control code is written in JavaScript, shader code is running on the GPU. X3DOM integrates the X3D graph directly into the HTML document object model. Some of those new concepts are discussed and implemented into the above mentioned project.

Keywords: Photogrammetry, HTML5, Video, Canvas, WebGL, X3DOM

Introduction of the author: Dr. Günter Pomaska, Prof. for Surveying and Informatics, Bielefeld University of Applied Sciences, Germany, Faculty of Architecture and Civil Engineering. Research topics: Photogrammetry, 3D computer graphics, web publishing.

1. Gumbinnen is today Gusev

Bielefeld city government took over partnerships to former East German cities. The cultural heritage of those cities is preserved in history rooms and archives. The municipal archive Bielefeld exposes a city model of Gumbinnen (East Prussia), today Gusev (Oblast Kaliningrad). It is noteworthy that the model and other items will be transferred to Gusev for the purpose of establishing a museum there.

Gumbinnen was first mentioned in records 1556 as a village and scaled up to a marketplace. After the great Black Death 1709/10 Friedrich Wilhelm I, King of Prussia supported the rehabilitation and Gumbinnen received 1724 municipal rights. Gumbinnen was capital of the administrative district of the same name. 1939 the population was about 24 500 inhabitants. After World War II the destroyed city was repopulated and given its new Russian name Gusev.

The architectural model was produced in 1967 due to the resolution of the representatives of Gumbinnen based on the official street map scale 1:5000 from 1936, aerial photos, photos, drawings and sketches.

Because of the projected resettlement the request for digitizing the model exists. The virtual model should be used for the purpose of computer animations, reconstruction and web publishing. In the first step a digital city model formulated in CityGML, an international OGC standard is produced by means of photogrammetry and CAD technology. Buildings are defined with different levels of detail for the major structures and roof elements. The model contains about 1200 buildings and is now available in renderings, animations and interactive real time animations and of course for Google Earth.

2. Digitizing and Modeling

2.1 Photogrammetric Recording

The object to be recorded has a surface of about 6.3 square meters. It couldn't be moved and the working space around is very limited. Light conditions must be accepted as found on site. Limitations in recording and evaluation time should be taken into account also. Photos were taken for stereo photogrammetry and multi image photogrammetry as well as photographs for details. Artificial targets are used as ground control points. Co-ordinate measurements came from two different positions with a Topcon Image Station.

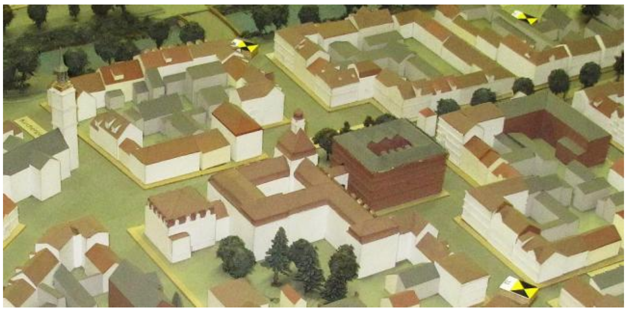

Figure 1. Photograph from the architectural model

The stereo photogrammetry was carried out like an aerial flight. For taking vertical images a bar of approx 3.5 meter length was built from metal pieces and moved crossways. The Canon Powershot G10 camera moved lengthways (flight direction) in stripes with approx. 60% overlapping over the model. Shutter release was provided by a remote control unit. Image resolution is 4416 x 3312 pixels with a pixel size of about 1.7 microns. The focal length setting was 6.1 mm hyper focal. Due to the long exposure time and the small pixel size the images display some noise. The flight simulation provided total 32 images in four stripes. In addition an image bundle for multi image photogrammetry was taken but not used for later evaluations. Structure from motion image based modeling was tested in a small area. Both alternative methods could not compete against the stereoscopic photo arrangement.

2.2 Evaluation

Principle of the photogrammetric evaluation is the parallax measurement (stereoscopic image arrangement). Menci from Italy is the manufacturer of an easy to handle software named ZGlif. The evaluation process is done in three steps:

- loading images and camera definition
- relative and absolute image orientation
- parallax measurement in a CAD like environment

Orientation computes the photo positions and the coordinates of orientation points (automatically detected by the system). Control points are connected with the corresponding position on the images and the software provides the adjustment of the calibration parameters. Quality of the orientation can be read from the residuals given for the corresponding points.

Displaying the model images on two half screens or anaglyph mode enables parallax detection. Measurement is performed by selection of geometric elements like points, lines, polylines and rectangles from the CAD menu and observing vertices in at least three images. Layers are also considered to make easier understanding of different elements in the model. The results are exported into DXF format in order to generate a draft of the 3D model. Those raw values must be connected to a unique CAD structure with grouping, assigning materials and environmental modeling.

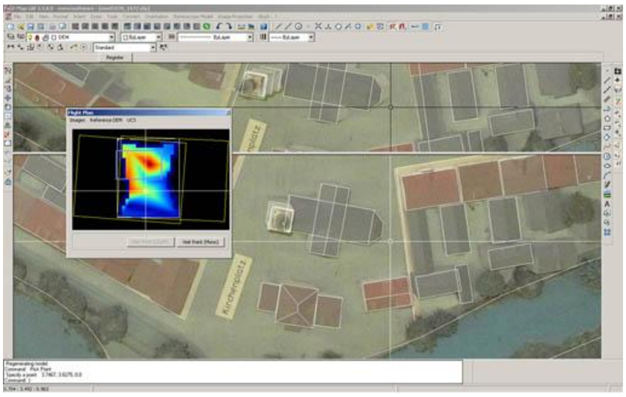

Figure 2 3D evaluation with ZGlif

2.3 CAD Modeling

The raw data coming in DXF format from the Z-Glif software was processed in single models. We did not make an adjustment of the complete image bundle. Transformation from the local system to the global coordinate system was controlled by the underlying scanned map 1:5000. This adjustment was carried out in a graphical manner inside AutoCAD and exported with the WBLOCK command for use inside SketchUp.

SU software is preferred for the 3D modeling because of the opportunity of its push-pull tool and grouping functions. The repeated details of the buildings are modeled as components, particular the structures of the roofs. Working with components adds structure to the model and enables much more efficient construction and modification.

Facades of landmarks are defined in a higher level of detail by using rectified and scaled textures (old images of the buildings). The appropriate materials and colors which are selected similar to the original features of the model are added to surfaces with their different layer structure. As mentioned above, the level of detail in modeling goes hand-in-hand with the OGC standard CityGML. The five levels of detail in CityGML define in level 2 the major horizontal structure and in level 3 the major vertical structure of buildings. The average model detail lies between this levels.

Finally the workflow took benefits from the free SketchUp software package, not to forget, that within a low budget project the team members are equipped with their own workstations.

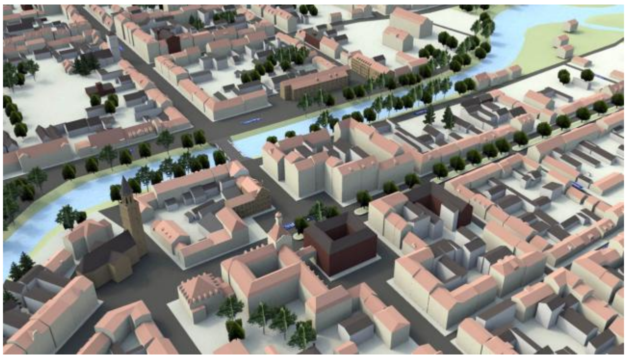

Figure 3 Rendering with image based lightning

2.4 Video Production

The 3d model is the source for fly-over or walk-through animation clips and high end renderings. An orthographic projection including light settings provides a true scale map. We could not apply photo realistic textures in general. Consider that photos don’t match with the model geometry which is sometimes estimated and not accurate enough. Therefore generating a complete photo realistic 3D model is far away from the capabilities. Instead linking of the photo database to the virtual model is an opportunity. Calculating camera positions from the pictures allows blending from virtual data to photorealistic data. Another possibility is given by calculating animations from still images by moving between areas, panning and zooming.

Finally the complete database consists of the following media types:

- 3D model in X3D and COLLADA for interactive walk through
- video clips from the a.m. model
- orthographic projection to the x-y-plane
- high resolution and high end renderings
- geo referenced photos
- video clips from photos and renderings(stills)
- video clips from text and graphics
- sound and text documents

We want to distribute this kind of media on the web, without requiring any plug-in or specific knowledge of the end user. Therefore we count on HTML5, Canvas in the 2D context, the video tag, displaying the video inside canvas and prepare a prospect to canvas 3D context now WebGL.

3. HTML5

In the following sections of this chapter the reader will find some snippets of HTML5 and JavaScript coding. Talking about event listeners and event handlers is not easy to understand without any knowledge in software development. But it may illustrate some of the new graphic concepts in HTML5. Since interactive 3D graphics on the web seems to be still for experts, the implementation of the canvas element and video tag promises success in continuation of this project.

3.1 The Canvas 2D Context

HTML5 (hypertext markup language) is the standard for defining content of web pages, supported by W3C (world wide web consortium) and WHATWG (web hypertext application technology working group), the industry. It replaces HTML 4.0, XHTML 1.0 and DOM level 2.0. Canvas applications are executed inside a defined area of the web browser for immediate mode graphics that can be manipulated with JavaScript.

In the first case study described here a map should be animated along a path. Following the recommendation of [2] a Java-Script template is used for the code. We assume that the browser supports the canvas tag. Initially we add an event listener to the window. The addEventListener() method expects three arguments: the event type, a callback function and the current event phase. After loading the page the canvasApp() function that covers the complete graphic application is called.

```
window.addEventListener('load',
                   eventWindowLoaded, false);
function eventWindowLoaded(){
  canvasApp();
}
function canvasApp(){
}
```

Inside canvasApp() we take use of global variable definitions. All graphic methods must be applied to the graphic context (the canvas). In the following example the canvas surface got the identifier *can* and we reference the 2D drawing context with the following two lines of code.

```
var theCanvas = document.getElementById("can");
var ctx = theCanvas.getContext("2d");
```

An image is displayed applying the drawImage() method with arguments defining the rectangle on the canvas where the image should appear and the displayable area in the image. If we repeat calling the function by a timer with changing the displayable area of the image, it animates. There is a select tag on the page with event handlers for the change event, see figure 4. If the user selects a location from the list, the timer is set and the image moves from the current position to the target position. Reaching the destination stops the timer, furthermore the collection of images changes. A full resolution picture can be displayed in the canvas area by clicking into the thumbnail.

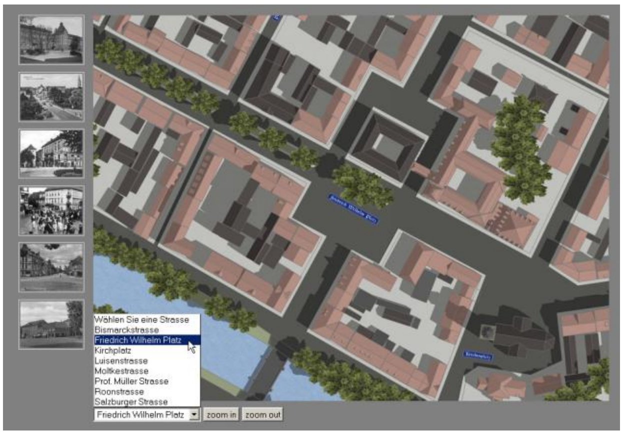

Figure 4 Moving through the city map

Event listeners are added to image thumbnails and the options of the select list. The event handlers share the canvas area for displaying images or animation. A feature for the end user is the geo reference applied to the location names. Here is some code from inside the canvasApp() function that demonstrates the animation and the use of the drawImage() function:

```
var timer;
function streetChanged(e){
  timer = setInterval (displayImage,10);
}
function eventPhotoLoaded(){
  drawScreen();
}
function drawScreen(){
  ctx.drawImage(photo,xNach-winW/2,yNach-
winH/2,
              winW,                    winH,
winX,winY,winW,winH);
}
function displayImage() {
  ctx.clearRect(0,0,winW,winH);
  /* winW, winH: width and height of canvas
*/
  /* change position in the image*/
  x = Math.floor(xVon+dx*step);
  y = Math.floor(yVon+dy*step);
  ctx.drawImage(photo,x-winW/2, y-winH/2,
  winW, winH, winX,winY,winW,winH);
  if (ds <= step++ ) clearInterval(timer)
}
```

The complete code of the software mentioned in this paper can be executed and downloaded from the web. Latest version of Firefox is recommended. Please visit:*http://www.imagefact.de/chdc2012.*

3.2 HTML5 Video Tag

Video support in HTML 5 seems to be very easy by coding the new video tag. The developer can place a video directly into a web page without using any plug-ins. The video tag comes with some attributes for auto-play, loop, volume and poster. A JavaScript API provides methods for controlling the playback. The difficulties are coming still from different video formats. Video formats are free or licensed and differ in quality of playback, size and browser support. Broad support today and in future will be given to OGG, MP4 and WebM encodings. Inside a container the content is coded. Theora (a video codec not supported by Apples iPhone) and Vorbis, an audio codec, stay for the OGG format. The MP4 format includes the H.264 coding. WebM is supported by Google and includes the VP8 codec and the Vorbis audio codec. To ensure that most browsers can playback the video one must apply a strategy for multiple formats. This reads in HTML markup like follows:

```
<video poster ="preview.jpg" controls>
  <source src="videofile.ogg">
  <source src="videofile.webm">
  <a href="videofile.avi">download video</a>
</video>
```

The browser plays the supported format or offers a file for downloading. Converting video formats is supported by freeware software and as well licensed products. A recommendation is the command line tool FFmpeg or Firefogg running in the Mozilla environment.

3.3 Playing a Video on the Canvas

Why should we play a video on the canvas area? It is more flexible for manipulating the video and in supporting interaction. A look to figure 5 may explain some strategy of the user interface in the second case study. On the left and right border of the web page are thumbnail images. In the bottom line is a selection tag. The clips should play in the center area, the canvas. The thumbnails are coming as image tags with an identifier. An array of event listeners is connected to the thumbnails to react on the click event. The event handler changes the source attribute of the video tag displayed on the canvas. The same code is acting to the select tag.

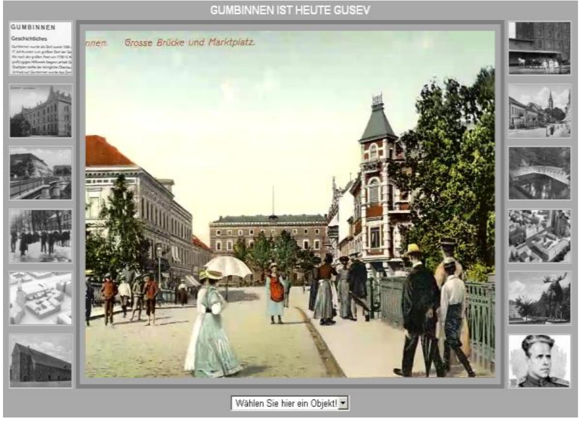
GUMBINNEN IST HEUTE GUSEV
GUMBINNEN
Geschichtliches
Grosse Brücke und Marktplatz.
Wählen Sie hier ein Objekt!

Figure 5 Playing video clips on the canvas

As mentioned under 3.1 initially an event handler is added to the window event load. After loading the page, a non visible video element is created in a div container. The display attribute is set to none. The canvas application function is called if the default video is loaded. The function itself calls the user defined drawScreen() function 33 times a second. The event handler function is coded as follows:

```
function eventWindowLoaded(){
 videoElement =
  document.createElement("video");
  videoDiv = document.createElement('div');
  document.body.appendChild(videoDiv);
  videoDiv.appendChild(videoElement);
  videoDiv.setAttribute(
             "style","display:none");
  videoElement.setAttribute("src", "
  grosseBruecke/Medium/output." + videoType);
  videoElement.addEventListener(
'     canplaythrough',videoLoaded, false);
}
function videoLoaded(){
  canvasApp();
}
```

From inside drawScreen() the video is displayed applying the drawImage() method to the graphic context. A code snippet of that part is given here:

```
function drawScreen(){
  ctx.fillStyle ='#ffffaa';
  ctx.fillRect(0,0,theCanvas.width,
                  theCanvas.height);
  ctx.drawImage(videoElement,0,0);
}
var theCanvas = docment.getElementById
                ("can");
videoElement.play();
setInterval(drawScreen,33);
```

Coding as above is following the recommendations in [2]. Executable code can be accessed from the web.3.1.

4. X3D Web Graphics

Virtual reality modeling language (VRML) is a scene graph file format for modeling interactive scenes. A visitor of the virtual world needs a VRML browser to navigate and interact in real time in the scene. VRML was published in 1997. Recent popular VRML browsers and web plug-ins are Cortona, Octaga and BS Contact. Octaga escaped from the web. Cortona and BS Contact are still existing, but only the latter can handle X3D.

Extensible 3D (X3D) is the successor of VRML in XML formulation. X3D is since 2004 an ISO/IEC standard. The disadvantage of VRML or X3D on the web is the requirement of a particular browser. In 2009 Fraunhofer IGD and the W3C started an initiative for direct integration of 3D content into the HTML5 standard. The existing solutions need plug-ins and handle isolated data models. Therefore the applications could not ensure extensive circulations. The new approach integrates the declarative X3D graph directly into the HTML-DOM and therefore does not need plug-ins-or APIs. Fraunhofer IGD provides an open source framework as a test implementation for evaluation and further development of the integrated data model. The current version of X3DOM needs an InstantReality plug-in, Flash11 or a WebGL enabled browser. As shown in figure 6 X3DOM was successfully tested by converting a cutout of the SketchUp model in X3D. In this case Blender imported 3DS from SU and exported the X3D formulation. Most of the geometry is transformed into indexed face sets. Some problems occurred with texture coordinates during the conversion process. X3DOM is a promising concept for distributing interactive 3D graphics on the web.

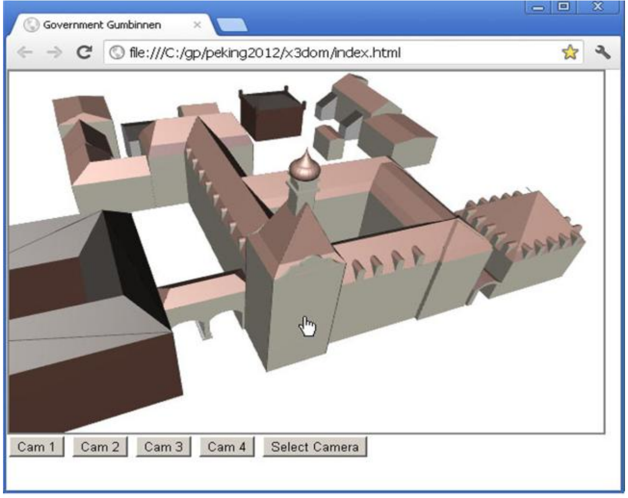

Figure 6 X3DOM application

5. WebGL

WebGL is an application programming interface (API) for advanced 3D graphics on web pages. WebGL provides rendering functionally in HTML5 and JavaScript context. Compared to desktop applications the major advantages of web applications are as follows:

- easy to ensure extensive circulation
- only a compatible web browser is required
- easy to keep the database up-to-date
- cross-platform support for multiple OS

WebGL with its characteristics as an open standard, access to the graphic hardware and running native in a browser will overcome some still existing limitations against native applications.

We have to differ between two fundamentally graphic API modes, the immediate- and retained-mode. An immediate-mode API must have a representation of the scene in the application and the complete scene needs to be redrawn for every frame.In retained-mode the graphics library keeps an internal model or a scene graph. When the application calls the API, it updates the internal model and the graphic library decides for redrawing. The application does not need to call drawing commands. In other words: An immediate-mode API updates the model and calls graphic commands, while a retained-mode API builds the scene only like SVG or X3D.

Getting the impression of realistic objects by light and material is known as shading. WebGL differs between vertex shader and fragment shader. A vertex is an intersection point in the 3D geometry. The vertex shader performs shading for such a vertex. After rasterization geometric shapes are converted to fragments, pixels that are finally drawn to the screen. The fragment shader outputs the color for the fragments, dependent from several (user defined) parameter. It is obvious that WebGL represents a low level API and is not as easy to handle as JavaScript Canvas drawing commands. Therefore some high level APIs accessing WebGL via JavaScript are already published or under development.

Investigating the applicability some of the open source and the licensed CopperLicht could be tested. The latter comes with an interactive scene setup. A WebGL JavaScript API is to mask the involved nature of WebGL from the developer and allows the use of hardware accelerated 3D applications without downloading plugins. Here are some short notes about the test candidates.

GLGE is a library from Paul Brunt, UK. A short test was successful with Firefox. GLGE calls itself *WebGL for the lazy*. It is still under heavy development.OAK3D comes from a developer team with a good documentation and examples. This engine reads XML files. An exporter is provided for 3DS files. Impressive examples of WebGL can be found on the SpiderGL web page from the Visual Computing Lab of ISTI. The author of SceneJS is Lindsay Kay. SceneJS provides a JSON based scene graph. Converter for COLLADA and OBJ are available.C3DL canvas 3D library is in development by a team from Seneca College Canada and other institutions. It was first developed as part of a network for canvas 3D, the precursor of WebGL. The web site provides a good documentation with examples and tutorials.

Numerous open source WebGL JavaSript APIs are available today. All have in common that they are heavily under development. Features differ from library to library. There is no doubt, that WebGL will be the solution for 3D content on the web. Most important for the content creator is a good conversion tool from the construction system to COLLADA or X3D and the handling of large scenes. The end user needs native support in all browsers.

6. Conclusion

To preserve the memory of the town Gumbinnen former inhabitants ordered a virtual 3D model of the town digitized from an architectural model that is exhibited now in Gusev. It could be stated that parallax photogrammetry is still a good choice for digitizing this kind of physical models. Together with a large photo database and other documents the material should give an impression about the cultural heritage before World War II. The Internet is the appropriate media for this project because it can reach a wide audience with minimal cost and high quality material. A concept for implementation is targeted to HTML5 and the canvas tag, using a modern browser without plug-ins. Navigation through 3D stuff is supported by predefined photo positions and video clips and a search engine for streets and objects to guide the visitor to his points of interest. Free navigation in virtual worlds seems not to be the correct answer under the marginal conditions. In future a 3d context (WebGL) to canvas will become a standard. A couple of JavaScript based APIs were tested. It is obvious that the technology HTML, CSS, canvas, JavaScript, DOM and WebGL can be used on the client for dynamic presentations inside the browser. The server only has to deliver the database. With HTML5 related technology and WebGL one more step to the vision of W3C *One Web for Everyone* is done.

References

[1] Anyuru, Andreas: *Professional WebGL Programming – Developing 3D Graphics for the Web*, Wiley&Sons, 2010

[2] Fulton, Steve; Fulton, Jeff: *HTML5 Canvas*, O'Reilly 2011

[3] Pomaska, Guenter: *Web-Visualisierung mit Open Source*, Herbert Wichmann Verlag, Heidelberg, 2007

[4] Pomaska, Guenter: *Webseiten-Programmierung: Sprachen, Werkzeuge, Entwicklung*, Springer Vieweg, Wiesbaden 2012

[5] Fraunhofer X3DOM framework:
www.x3dom.org

[6] WebGL Javascript APIs:
www.glge.org; www.oak3d.com;
spidergl.org; scenejs.org;
www.c3dl.org;www.ambiera.com/copperlicht/

[7] Project web site
www.imagefact.de/gumbinnen

[8] Examples and source code from this paper
www.imagefact.de/chdc2012

Acknowledgements

This project is a co-operation between Kreisgemeinschaft Gumbinnen, University of Applied Sciences Bielefeld, Germany, Izmir Institute of Technology, Turkey and Vologda State University, Russia. Evaluation and modeling is provided by Funda Uygun and Suna Büyükkilic from IYTE during their stay in Bielefeld supported by the European Erasmus Program. Dr. Nikolay Dementiev from Vologda prepared more detailed landmarks.