NORSIGD INFO

Nummer 2 1999

NORSK SAMARBEID INNEN GRAFISK DATABEHANDLING

ISSN 0803-8317

Aktivitetskalender

Hva skjer når og hvor?

Mai 1999	
26–28	**Eurographics - IEEE Symposium on Visualization**, Wien, Østerrike. http://www.cg.tuwien.ac.at/conferences/VisSym99/.
31–(1)	**EGVE'99**, Wien, Østerrike. http://www.cg.tuwien.ac.at/conferences/egve99/.
Juni 1999	
2–4	**6th International Eurographics Workshop on Design, Specification and Verification of Interactive Systems**, Universidade do Minho, Braga, Portugal. http://sim.di.uminho.pt/dsvis99/.
21–23	**10th Eurographics Rendering Workshop**, Granada, Spania. http://alhambra.ugr.es/egrw99.
August 1999	
8–13	**SIGGRAPH'99**, Los Angeles, USA. http://www.siggraph.org/.
22–27	**8th Int. Conference on Human-Computer Interaction**, München, Tyskland.
September 1999	
7–11	**Eurographics'99**, Milano, Italia. http://eg99.dsi.unimi.it/.
Oktober 1999	
5–7	**Pacific Graphics'99**, Seoul, Korea. http://comp.snu.ac.kr/pg99/.
30–(4)	**ACM Multimedia'99 – The 7th ACM International Multimedia Conference**, Orlando, Florida, USA. http://www.acm.org/sigmm/MM99/.
November 1999	
17–19	**Erlangen Workshop'99 – Vision, Modeling, and Visualization**, Erlangen, Tyskland. http://sfb-603.uni-erlangen.de/vmv99/.

Om forsiden
På forsiden viser vi hvordan hallen i Kaiserpfalz i Magdeburg kan ha sett ut. Bygningen ble brukt av Otto den Store i hans regjeringstid på nihundretallet. Forsidebildet er samtidig Figur 9 i artikkelen *Masuch et.al.* i denne utgaven, som omhandler visualisering av gammel arkitektur.

Hilsen fra styret

Kjære medlemmer,

Vårutgaven av NORSIGD Info er spekket med aktuelle temaer. CAVE var tema for det interessante og lærerike NORSIGD seminaret sist høst i Bergen. Da dette fortsatt er et svært aktuelt tema kommer vi med en oversikt hva denne teknologien forteller om i denne utgaven. De andre temaene handler om visualisering av arkitektur og opptak av geometridata ved hjelp av robotstyrte kameraer.

NORSIGD's fagansvarlig har besøkt marskonferansen i Magdeburg, der også fjorårets prosjekt om Grafikk på Web ble presentert. Det viste seg at det er stort behov for metoder som gir tilgang til grafiske ressurser på en nettleser. Mangelen på avansert grafikk på Web er et påfallende problem.

Vi er nå inne i jubileumsåret for NORSIGD. Det gjenvalgte styret planlegger å markere dette med aktuelle seminarer.

Hilsen,

Wolfgang Leister

NORSIGD Info
– medlemsblad for NORSIGD

Utgitt av:	NORSIGD
Ansvarlig:	Wolfgang Leister Norsk Regnesentral Postboks 114 Blindern 0314 OSLO
ISSN:	0803-8317
Utgivelser:	1999: 10/2 20/4 20/9 20/12
Annonsepriser:	Helside kr 5 000 Halvside kr 2 500
Oversettelser:	Wolfgang Leister
Layout:	Wolfgang Leister LaTeX2ε

Ettertrykk tillatt med kildeangivelse

Innhold

CAVE — en oversikt

Heinrich Müller[1]*, André Hinkenjann, Roland Blach, Martin Göbel, Ulrich Lang, Stefan Müller*

CAVE var også tema for NORSIGD seminaret sist høst i Bergen. Dette bidraget er en forkortet versjon av en artikkel presentert under en workshop i forbindelse med GI årsforsamlingen. Forfatterne er ansatt ved universiteter og forskningsinstitutter i Tyskland: Univ. Dortmund, FhG IAO, GMD, Univ. Stuttgart og FhG IGD.

CAVE er et konsept for 3D menneske-maskin-interaksjon. Konseptet ble presentert for første gang i begynnelsen av 90-årene. Til tross for den store tekniske kompleksiteten finner CAVE stadig større utbredelse. Dette bidraget gir en introduksjon til CAVE-teknikken og viser mulige anvendelser. Alternative teknikker for 3D-interaksjon blir også sammenlignet med CAVE. Artikkelen er en forkortet versjon av em artikkel presentert på en workshop i forbindelse med GI årsforsamlingen.

Trenden innen menneske-maskin-interaksjon går mot "virtuelle omgivelser" ved at syntetisk genererte signaler virker inn på sansene til en bruker. Fysikalske eller abstrakte fakta blir videreformidlet til brukeren på en slik måte at vedkommende tror at disse inntrykkene er reelle. Fordi teknikken omfatter både syns-, lyd-, og berørings-inntrykk kan brukeren få en særlig høy grad av integrasjon i en presentasjon.

Dagens VE-systemer (VE = Virtual Environments) tilbyr syntesen av visuelle, akustiske, og haptiske (berøring) inntrykk uavhengig av applikasjonen. Immersjon, som er definert i hvor stor grad man er del av den virtuelle verdenen,forsøkes å gjøres høyest mulig. For grafikkvisualiseringen brukes stereo-teknikker, mens foskjellige akustiske teknikker blir brukt for lyd. For stereosynet leveres adskilte bilder til det venstre og høyre øyet. Lyd blir generert ved hjelp av 3D lyddatamaskiner hvor retning og posisjon kan velges fritt. For navigasjon i rommet og manipulasjon blir spesielle apparater brukt for å spore posisjon og orientering i rommet.

CAVE finnes i stor utbredelse, til tross for de store tekniske kravene som er forbundet med det. CAVE er et akronym for Cave Automatic Virtual Environment, mens det engelske ordet *cave* betyr "hule". CAVE ble utviklet ved university of Illinois av Carolina Cruz-Neira, Dan Sanding, Tom DeFante og studenter, og ble vist ved SIGGRAPH'92. Navnet CAVE er nå et varemerke for universitetet i Illinois, og systemet markedsføres av det amerikanske firmaet Pyramid Systems. I Europa markedsføres produktet av Tan i Tyskland, som har den høyeste tettheten av CAVE-lignende installasjoner i dag.

Oppbygning av en CAVE

Kjernen i en CAVE er et terningformet rom som brukeren befinner seg i. Veggene i en CAVE er vanligvis 2-3m lange. En av veggene og taket er vanligvis åpent. Stereoprosjektører projiserer bilder på de transparente veggene utenifra (på gulvet ovenifra). Dette blir gjort slik at bildene smelter sammen til en enhetlig stereoprojeksjon i rommet. Brukeren har på seg stereobriller mens hun befinner seg i rommet. Brukerens posisjon og synsretning i rommet blir tatt opp av sensorer i brillene, som vanligvis arbeider på grunnlag av elektromagnetiske felt. En datamaskin bygger opp bildene ut ifra brukerens posisjon og synsvinkel, slik at han får korrekt synsinntrykk.

For interaksjon med grafikken som presenteres har brukeren et pekeredskap, hvis posisjon i rommet også blir tatt opp av sensorer. Pekeredskapet kan også brukes for å velge fra menyer som projiseres i rommet.

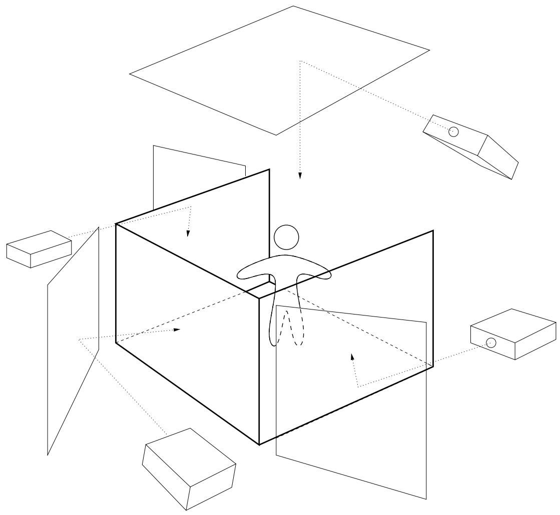
Grafikk-datamaskinen genererer realitetsnær grafikk i sanntid, ofte ved å bruke OpenGL som grafikk-bibliotek. Med en oppdateringsfrekvens på minst 15–20 bilder per sekund, som projiseres på flere lerreter i stereo kvalitet, kreves det høy grafikkytelse.

I tillegg kommer generering av lyder, mottak av signaler fra input devices, beregningstid for selve applikasjonen, kommunikasjon og administrasjon av systemet. På grunn av disse kravene brukes meget kraftige datamaskiner i en CAVE, ofte implementert med flere prosessorer.

Bildene fra videoprosjektørene projiseres ved hjelp av et speilsystem på lerretene for å minske plassbehovet. Prosjektørene må tilby konsistent lysstyrke og kontrast i tillegg til en så liten forvridning som mulig. Det kreves prosjektører med høy lysstyrke for å gi tilfredsstillende fargekvalitet. Ved hjørnene og kantene i en CAVE stilles det spesielle krav, da de projiserte bildene må passe sammen uten skjøt. Den relativ lave oppløsningen av dagens datagrafikk og interrefleksjon mellom veggene kan lage kunstige uønskede effekter eller uskarpe bilder.

[1] Universität Dortmund, email: mueller@ls7.informatik.uni-dortmund.de

Til stereoinntrykket blir vanligvis shutterbriller brukt, der brilleglasset består av styrte LCD krystallflater, der vekselvis det venstre og det høyre glasset blir gjennomsiktig. For å oppnå en god kvalitet må frekvensen være høy, vanligvis 60 Hz.



Tegningen viser oppbygningen av en CAVE med fire vegger

Elektromagnetiske systemer brukes vanligvis for å følge bevegelsene til brukeren (tracking). Disse har ikke områder der signalet er skygget, slik det er med optiske eller ultralydbaserte systemer. Elektromagnetiske systemer har en ulempe i at nøyaktigheten avtar med voksende avstand. I tillegg er nøyaktigheten ikke konstant over tid. Dette kan kompenseres ved å bruke såkalte *long range transmitter* som bygger opp et kraftigere magnetfelt i forhold til en standard transmitter. Metallgjenstander kan forvrenge magnetfeltet i en CAVE. Derfor blir CAVE-rammer ofte laget av tre som byggemateriale. Imidlertid har dette også en ulempe ved at tre lettere kan vri seg, særlig i kantene. Da bruk av metall ikke alltid kan unngås må en CAVE kalibreres slik at de negative effektene minskes.

Som interaksjonsredskap i en CAVE brukes såkalte "free flying joystick" eller "wand", som brukes i hånda med fritt definerbare interaksjonsknapper. Bevegelsen av instrumentet i rommet blir fulgt av tracking-systemet, som beskrevet ovenfor. En annen mulighet for interaksjon gir datahansker som tillater brukeren å gripe etter objekter. Også fast installerte apparater, som kan være del av det virtuelle objektet blir bruk: Et eksempel for dette er at brukeren befinner seg i et ekte cockpit av en racerbil, mens de aktuelle omgivelser projiseres på veggene.

Oppbyggingen av en CAVE (lerret, prosjektører, speil) koster ca. 1.5 millioner kroner, mens datautstyret (avhengig av ytelsen) koster flere millioner kroner.

Anvendelser

Det finnes mange anvendelser for CAVE teknologien. Spesielt i bilindustrien og utviklingen av mekaniske apparater brukes CAVE for *virtuell prototypeutvikling*. Teknikken brukes bl.a. for å visualisere og vurdere karosserideler og for å evaluere konstruksjonsdata. Ved siden av estetiske kriterier kan teknikere også vurdere om delene passer sammen i sluttkonstruksjonen.

Andre muligheter for virtuell prototypeutvikling finnes innen funksjonale analyser og ergonomistudier. I bilteknikken kan f.eks. funksjonaliteten av dashbordet prøves ved at man visualiserer en 3D-modell. Også luftstrømmer i passasjerkabinen kan simuleres ved hjelp av en partikkel-modell og deretter gjenskapes på fingertuppene til en person som befinner seg i en CAVE.

De tekniske kravene er avhengig av applikasjonen, og spenner fra visualisering av forhåndsberegnete modeller til grafisk on-line interaksjon med en applikasjonsrettet simuleringsprosess. For krevende applikasjoner kan det være nødvendig å gjennomføre simuleringen på en egen datamaskin med høy ytelse.

Innen virtuell prototypeutvikling kan flere personer arbeide sammen, både når flere personer befinner seg i en CAVE, og når flere installasjoner er knyttet sammen med et raskt datanett. Det finnes også forsøk på å representere personer i en CAVE, i en 3D videokonferanse. Deltagerne blir tatt opp ved hjelp av et stereo kamera, og opptaket blir projisert som tekstur i en CAVE. Eksterne personer kan også representeres som såkalte avatarer, hvor deltagerne blir fremstilt som modellerte figurer med programmert oppførsel.

Prototypeutvikling er ikke begrenset til det mekaniske området, men kan også brukes innen arkitektur. I en CAVE kan bygninger, landskap og byer besøkes virtuelt, noe som gir et inntrykk av å være på stedet. Denne teknikken ble også brukt for å visualisere Terminal 1 ved flyplassen i Frankfurt. Inntrykket blir sterkere når et radiositet-prinsipp brukes for å fremstille diffus interrefleksjon av lys i rommet.

Andre eksempler for virtuell prototypeutvikling finnes innen design av molekyler og medisin (diagnose, terapi og trening). Virtuelle omgivelser brukes også for å forske på menneskelig atferd og for terapi av atferdsproblemer. Anvendelsesområder finnes også innen underholdning og læring (edutainment). Store bedrifter ser ut til å se muligheter for bruk av CAVE for underholdning. Et eksempel er et virtuelt oseanium, der fiskesvermer blir simulert i sanntid når de beveger seg i stim eller går til angrep. Tilskueren blir guidet av en virtuell skuespiller som forklarer fakta om havet. Andre installasjoner finnes innen kunst, f.eks ved Ars Electronica Center i Linz eller ved NTT i Tokyo.

Alternativer

CAVE er en kostbar high-end løsning for visualisering og manipulering av virtuelle omgivelser. En slik oppbydelse er ikke alltid nødvendig, forsvarlig eller gjennomførbar. Som et rimelig alternativ har vi grafiske presentasjoner og VRML på datamaskiner med vanlige skjermer, hvor tastatur og mus brukes for interaksjon. Neste steg i retning av større immersjon er bruk av 3D briller med separate bilder for hvert øye.

Helmet mounted displays (HMD) er integrert i en hjelm som bæres på hodet. Bildet overføres med en spesiell optikk i hjelmen til øynene, ved hjelp av en liten integrert monitor. Den første HMD ble presentert av Evans & Sutherland allerede i 1965. HMDs blir vanligvis brukt i VE scenarioer som krever at brukeren kan bevege seg fritt. Posisjonen til hjelmen blir sporet på samme måte som i en CAVE, slik at bildegenereringen er i henhold til brukerens bevegelser. Input av data skjer også med 3D interaksjonsredskaper som i en CAVE. Ulemper ved denne teknikken er den manglende komforten for brukeren samt at det er vanskelig å ta hjelmen av. Dessuten må alle brukere ha sin egen hjelm i motsetning til en CAVE, hvor mange brukere kan oppholde seg samtidig i samme projeksjon (selv om bare en bruker ser den korrekte projeksjonen).

BOOM (Binocular Omni Orientation Monitor) er et annet alternativ. Det amerikanske firmaet Fakespace tilbyr denne muligheten for stereosyn i en virtuell verden. To billedrør er montert i en kasse, som henger utbalansert av to vekter på et system av stenger. Rett ved kassen er det taster for interaksjon. Sporing av bevegelsene skjer mekanisk, og er derfor fritt for forstyrrende innflytelser utenifra (magnetfelt). Ulempen med en BOOM er at brukeren ikke kan bevege seg fritt i rommet.

I tillegg finnes det andre omgivelser basert på projeksjon, som er godt nok for noen applikasjoner, f.eks. Responsive Workbench, Infinity Wall og ImmersaDesk. Men CAVE tilbyr bedre romlig visualisering og en høyere grad av immersjon enn disse metodene. CAVE er for tiden den mest omfattende, intuitive og mennesketilpassede metoden for å løse oppgaven til 3D menneskemaskin interaksjon.

Fremtiden

VE systemer kommer til å utvikle seg videre i minst samme tempo som før når det gjelder maskinytelse og kapasitet. Det kan ventes en tydelig forbedring i oppløsning og skarphet for HMD med nye laserprosjektører, som projiserer bildet direkte inn på netthinnen. Innen videoprojeksjon kommer dagens rørbaserte systemer til å bli byttet ut med LCD-baserte prosjektører.

Den ideelle løsningen ville vært tredimensjonal laserholografi. Til tross for intensiv forskning finnes det fortsatt mange problemer som må løses før dette mediet blir ytelsesmessig tilstrekkelig for dagens teknikker.

En ulempe med dagens VE systemer i forhold til den virkelige verdenen er at gjenstander ikke er laget av materie, dvs. at man ikke kan gripe objektene. Det blir utviklet såkalte haptiske displays som kan gi fysisk motstand. Til tross for mange kreative ideer som egner seg for begrensede applikasjoner mangler det en generell løsning. Et eksempel er PHANToM, som tillater en mekanisk kraftkobling ved hjelp av motorer, eller såkalte shape forming devices, som gjenskaper virtuelle gjenstander ved hjelp av stengematriser eller væsker med foranderlig viskositet.

Det ville også vært ønskelig for brukeren med trådløs overføring av sporingsdata, eller videobasert sporing av posisjonen. Ved hjelp av håndbevegelser som gjenkjennes kan flying joysticks bli overflødig i fremtiden. Det finnes også mulighet for et begrenset sett for språkstyring, f.eks innen medisinske applikasjoner.

Grafikkytelsen, som for noen år siden bare fantes på spesielle datamaskiner finnes idag på PCer med tilsvarende 3D-grafikkort. Det tekniske fremskrittet gjør det mulig å basere fremtidige CAVEs på PC teknologi for noen applikasjoner. Det forventes at den idag fortsatt kompliserte CAVE teknologien vil finne enda større utbredelse i fremtiden.

Visualising Ancient Architecture

Maic Masuch, Thomas Strothotte
Otto-von-Guericke University of Magdeburg

We present a visualisation of an medieval palace as an application for a new rendering and animation tool which is capable of rendering images in the style of line drawings. In contrast to other systems, we are able to generate frame-to-frame coherent line drawings in order to render an animation. The application of this presentation illustrates the capabilities of animated line drawings and makes use of different drawing techniques such as simplifying a scene and emphasising important objects.

The use of non-photorealistic images is common in scientific and educational visualisation. The abstraction of a scene can have essential advantages. In contrast to a photo or a photorealistic rendered image, the viewer's focus of attention can be placed on important elements of the picture. Different drawing styles may be applied to objects to encode various levels of importance, i.e. using bright lines for less important objects and strong lines to depict a more important object.

A non-photorealistic image can differ from a photorealistic image in shape, colour, texture, light and shadow and even can leave out details in order to simplify the visual impression. Thus, the resulting image may have different levels of detail, i.e. fine detail in interesting areas and just rough outlines in less important areas. Often realistic settings of light, realistic texture or accurate shapes are not that much important to convey the essential information.

If these advantages hold for still images, they certainly hold for moving images. During an animation, a viewer always has less time to ascertain important information and it is therefore of utmost importance not to distract the viewer's attention with unnecessary details. If we want to meet the special demands summarised above, we have to investigate methods to render non-photorealistic animation.

Non-photorealistic Rendering

In contrast to the field of scientific visualisation, the main scope of current computer graphics research still lies in the aim to generate images that can no longer be distinguished from a photograph. Only a few systems deal with the creation of non-photorealistic images, and even less with the creation of non-photorealistic animation. If we investigate existing techniques for the generation of such pictures, we can distinguish between two basic types of imaging software: 2D animation systems and 3D rendering systems.

2D animation systems evolved from traditional animation. They are designed to create images that look like their examples and we have virtually no re-usability of models, which turns out to be a severe drawback. In addition, 2D systems cannot handle correct perspectives by their very nature, which make them highly unsuitable for the field of architectural presentations.

3D rendering systems like the PIRANESI-system presented in 1995 by SCHOFIELD et al., or techniques introduced by SALESIN et al. concentrate on the creation of single images and depend strongly on user interaction. The use of random line deviations to model the unsteadiness of a hand-drawn line results in visual temporal artifacts, which are described as lack of *frame-to-frame coherence*. The non-determistic generation of lines leads to the effect, that no two successive frames look the same, even if there is no motion at all. The viewer encounters a disturbing waving and pulsing in the appearance of object shapes, which is is an unwanted effect, as it distracts the viewer's attention.

The only known system for the generation of frame-coherent non-photorealistic animations was introduced in 1996 by MEYER. She presented a rendering technique that simulates the process of painting by placing coloured brush strokes on the picture. These images result in an animation which resemble oil paintings. However, this approach seems unsuitable for the generation of animations in the field of architectural visualisation.

Computer Graphics in Architecture

The visualisation of architecture often implies the intention to give the viewer an impression that is as close as possible to reality. However, perception and understanding of a presented image are subjective processes and thus influenced by many factors. Therefore, a photorealistic image does not always meet the requirements of the purpose. Often, in an early design stage, sketches and hand drawings are used, although it would be possible to create a photorealistic image as well. As STROTHOTTE et al. have shown, an image, rendered with the techniques of line drawings, is not taken as final rendition. A sketch or a hand drawing implies the "unfinishedness" of a depicted object and invites discussion.

Despite its advantages, this non-photorealistic type of presentation seemed initially to be unsuitable for the generation of animation, because hand-drawn frames always imply the lack of frame-to-frame coherence, as explained in the previous section. This can be changed by adapting non-photorealistic visualisation techniques to the field of computer-generated animations. Another, rather surprising demand of historic visualisation is the problem how to present details about which experts are uncertain. A blank space in a photorealistic image breaks the continuity of the depicted scene. In a non-photorealistic context, such questions can be left open without interrupting the continuity of the visualisation.

Rendering Line Drawings

In order to achieve frame-coherent output, we replaced the concept of stochastic deviations to model human drawing styles by the complete parameterisation of a line. Our rendering and animation system can be divided basically into two components: the render engine that is responsible for the process of rendering single frames, and the animation engine that generates the input data for the rendering component.

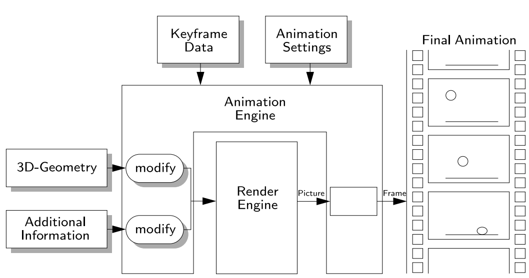


Figure 1: The system overview for the dal!*-system.*

We are using a high-end animation program (3D STUDIO) for the process of modelling and setting up an animation. It is important to benefit from the power of an existing tool, as we concentrate on the visualisation of the depicted objects. We have the following types of input data for our render engine:

- 3D geometry of a scene, including information about cameras and lights,
- additional information stored in a scene hierarchy, which also contains information about object-specific line styles, etc.
- keyframe data that describe all object movements and transformations in the scene, as well as keyframe data about the animation of line attributes and
- general animation data, like the number of frames, output format, resolution etc.

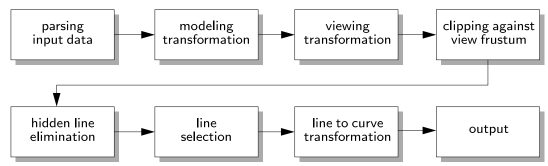


Figure 2: The rendering pipeline.

The steps that have to be performed by the render engine to go from a 3D model to a rendered frame of the animation are as follows: The input data is read and parsed by the animation engine, which provides the render engine with the necessary scene information for each single frame. First, the render engine performs a modelling and a viewing transformation. Then the transformed objects are clipped against the view frustum and a hidden line/hidden surface removal algorithm determines which lines are visible and which are not. The output of the hidden line elimination consists of a set of line segments, characterised by their start and end points. Furthermore, the rendering algorithm generates additional information that is needed in the next step for the transformation from individual line segments to a curve, like edge classification (into contour-, inner- or triangulation edges), neighbourhood information (left and right neighbours of an edge), smoothing information (edge is smooth or not), whether a path is cyclic or not and other valuable information. After that, matching line segments are connected to chains which serve as basis for the drawing path of a line (see next section). This data is passed back to the animation engine which paints the vector oriented representation of the image in an abstract output medium. At this point, the picture is converted into the desired output format, i.e. a vector-oriented format like POSTSCRIPT or a pixel-oriented format li-

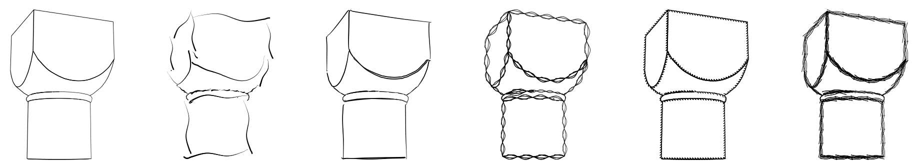


Figure 3: The first capital is rendered with a plain, unmodified line style, which depicts the result of the visible line determination. The following pictures depict the same capital with different line styles applied. In all images the geometry of the object remains unchanged, only the appearances of the lines are altered.

ke GIF. This rendering pipeline is subsequently executed, until all frames are rendered. Finally the successive frames can be encoded in an animation on video or an adequate digital format.

The Generation of Lines with Style

How a line drawing looks like depends on two crucial aspects: The placement and the appearance of the lines depicting an object. The placement of a line results from the determination of visibility that delivers a start and an end point for a given line. The appearance of the line is encoded in a parametric description concerning all characteristics of a line and is called *style*. The style is defined by a number of control vertices, an interpolation method, a parametric description that specifies deviations of the path, the course of line width and line brightness and other attributes. The final appearance of a line is calculated by the superposition of path and style. The line deviations model the inaccuracy of a hand drawing. Figure 3 illustrates the effect of line styles applied to a capital.

Model Requirements

The daLi!-system uses a polygonal representation of geometric objects and additional data as input. A former version of the system was restricted to 3D objects that consist of polygon meshes that are strictly non-intersecting. This demand has turned out to be a rather severe limitation and has been overcome by using a new clipping algorithm for the determination of visible lines and surfaces. The new method is slower, but no longer restricted to non-intersecting models.

Furthermore, we found out that for computer-generated line drawings, the level of detail of a model can be reduced significantly. Line drawings intuitively are associated with preliminary drafts, and therefore – in general – do not require such a high level of detail as models for photorealistic images. But if we want to have direct access to the way how certain objects in a scene are depicted, we have to think about techniques to specify the drawing style of an object.

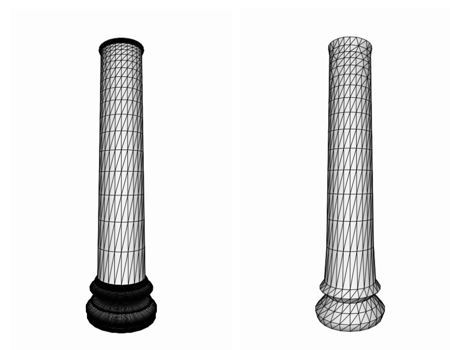


Figure 4: Using a mesh optimisation algorithm on certain objects does not necessarily change their visual quality, even though the second column consists of only 10% of the number of polygons of the first column.

The Enriched Scene Model

In order to determine how certain objects or parts of an object should be rendered, i.e. which line style should be applied to specific lines, we use an enriched scene model. By organising the scene elements in a tree-like structured scene hierarchy, we assign attributes to the geometric description of 3D scene elements. In certain animation techniques it is common to use a hierarchical linking of related objects, e.g. to evaluate *inverse kinematics*. We extended this scene hierarchy to organise attributes of geometric objects. Virtual, i.e. non-existing nodes, can be inserted to built groups of sub-objects in a scene. Node attributes such as movement, line style parameters, or visibility information can be defined and they can be inherited by child objects. This hierarchy can either be constructed explicitly during the process of 3D modelling or it can be generated semi-automatically. Figure 5 illustrates the use of a simple grouping of objects in a scene. The board as a major object inherits the drawing style *bold* to child objects, which are drawn in the same style. The hierarchy is stored in a separate file and is evaluated during the rendering process.

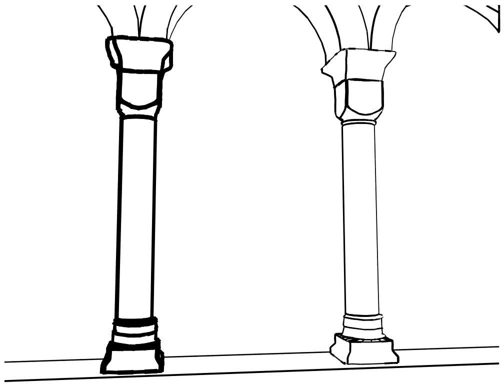


Figure 5: Using the hierarchy to group certain objects of a scene. All parts of the column (shaft, rim, capital etc.) are sub objects of the hierarchy node left column, *which is attributed with a bold line style in order to emphasise the column.*

Animating the Line Style

Due to the separation of style and path of a depicted line it is possible to animate not only the movement of objects but also specific attributes of a line style, e.g. the line width. This unique form of animation allows us to gradually change the appearance of an object without changing its geometry, i.e. applying traditional visualisation techniques like decoding the importance of a depicted object by the variation of the drawing style.

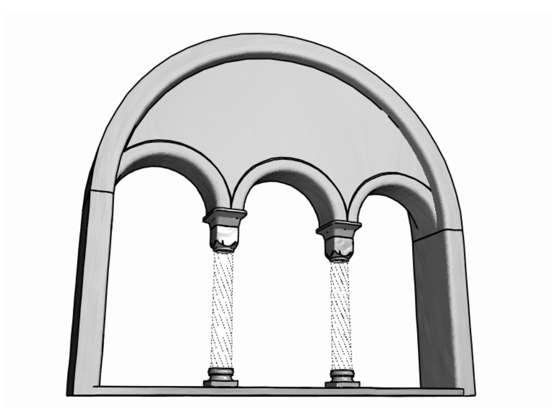


Figure 6: Gradually changing the line width for an object to draw the attention on an important sub object.

Figure 6 shows one frame out of a short sequence that explains the shape of romanesque columns of an arch window. By depicting the columns with bright lines, they become de-accentuated. This de-accentuation can be achieved instantly or gradually over a number of frames; an effect that can be compared to emphasing a text by printing it in bright letters. The parameterisation of the style allows the following line attributes to be animated:

- line width
- line brightness
- line style
- randomness of a line

While the variation of line width and line brightness are adequate techniques to encode importance in the animation, the line style can model hand drawn pencil strokes. The randomness of a line directly controls the frame-to-frame coherence of a depicted line. Certainly, selected discontinuities in the depiction of a line can be used to enhance the impression of movement, particularly if not all objects in a scene are subject to this effect (otherwise we encounter again the disturbing absence of frame-to-frame-coherence). This feature, however, is rarely used in architectural visualisation.

Reconstructing an Ancient Site

A virtual reconstruction allows scientists to share their explorations and discoveries with fellow researchers and the public. In addition, the reconstruction process often leads to completely new insights, because it forces the researcher to think about completely new questions, e.g. how everyday life in an ancient building actually was (Where were the doors? Did they have locks?). Virtual reconstructions of ancient sites are an increasing application of computer graphics and virtual reality.

The Palace of Otto the Great

A first written mention of a settlement named *Magdeburg* was documented in the year 805. In the following two centuries, it developed from a minor trading post at the eastern borderlands to a centre of the German empire. Otto the Great and his wife Editha, daughter of Athelstan, the first king of England, lived in Magdeburg for several years, before Otto himself became King in 936. In the year 962 he was coronated by Pope John XII as Emperor of the Holy Roman Empire. He understood his position as a direct successor of the Roman emperors of the West. Consequently, he ordered to extend the existing palace, to reflect his position as most powerful ruler in the western world. Without regard for expenses the building should compete with the Byzantine emperor's palace. It became an impressive two-storied building, about 30 meters high, with two towers, a huge King's hall and several surrounding buildings. Even tons of marble were brought from Ravenna over the Alps as construction material for columns and tessellated floors. But this glory should not last for long. In the next centuries, the power of the emperors declined and the palace became abandoned. After a disastrous fire in the town of Magdeburg in the year 1207, the ruins of the palace and the surrounding buildings served as quarry for a new cathedral. The once greatest and most impressive secular building of its time vanished.

The Visualisation of the Palace

Archaeological excavations from 1959 until 1968 revealed a large area of foundations near the cathedral of Magdeburg (see Figure 7).

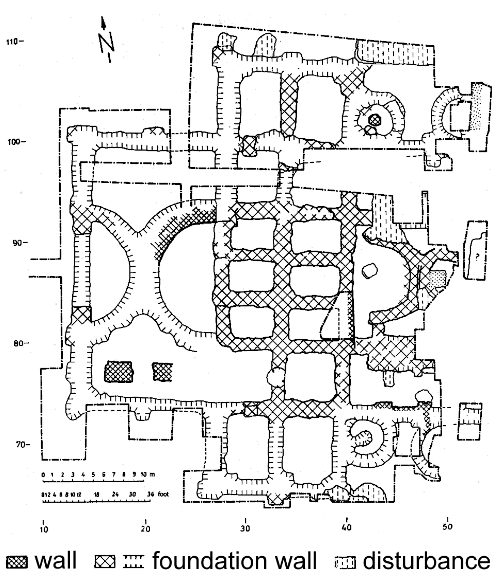


Figure 7: A map of the excavation site which served as basis for the reconstruction.

The artifacts discovered indicated that on this site the long lost palace of the Ottonian kings must have been located. These discoveries served as basis for a three-dimensional reconstruction of the palace. As there exists neither a picture nor a description of the palace, the reconstruction must be considered to be highly speculative by nature. The shapes and measures of the main building can be deduced from other existing Ottonian buildings, mostly churches and cathedrals, but sometimes the reconstruction is not more than an "educated guess". With so little knowledge about the structure and outlook of the building, archaeologists and historians have difficulties with agreeing on details. We found out that at this point, an image rendered in the style of line drawings appears far more adequate than a photorealistic image, which leaves no room for discussion and which suggests a final form, even if there are doubts.

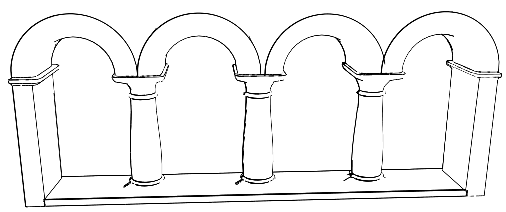


Figure 10: Large window in the great hall.

Using computer-generated line drawings, it is possible to leave open certain details. Besides the absence of texture and colour, the non-photorealistic image itself invites discussion on the subject. An image rendered with the techniques of line drawings is not taken as final. With aid of the daLi! rendering and animation system, we can render images and animations with less detail, using techniques for emphasising and deemphasising, and we can reuse and develop the 3D model. The development of the 3D model is a factor that cannot be overestimated. In the reconstruction process, knowledge about the investigated building is gained. Therefore it is important to be able to update and reuse the computer-generated model in order to improve the accuracy of the reconstruction.

Conclusion

In preparation of our exhibition about Otto the Great, a first reconstruction of the palace of Otto the Great was carried out by students. This 3D model serves as a basis for discussion among experts who will have to come to an agreement on details. As there are even different opinions on the external shapes, it is questionable, whether there will be a settlement on interior colours and textures. Figures 8, 9 (see cover page) and

10 show some frames from a first animation, a short exploration of the palace. Note, that the model consists of about 90,000 polygons, so that the reduction in detail is very significant.
We currently are developing techniques for rendering frame-consistent hatching lines which will improve the 3D impression of objects. The placement of the lines poses a quite difficult problem. Furthermore, it would be interesting to develop techniques for the representation of many objects, e.g. leaves of a tree or clouds. In the past, artists have developed some powerful techniques to depict these kind objects, e.g. painting a tree with characteristic strokes rather than painting every single leaf.

References

Th. Strothotte (1998), Computational Visualization: Graphics, Abstraction and Interactivity, Springer-Verlag, Berlin - Heidelberg - New York

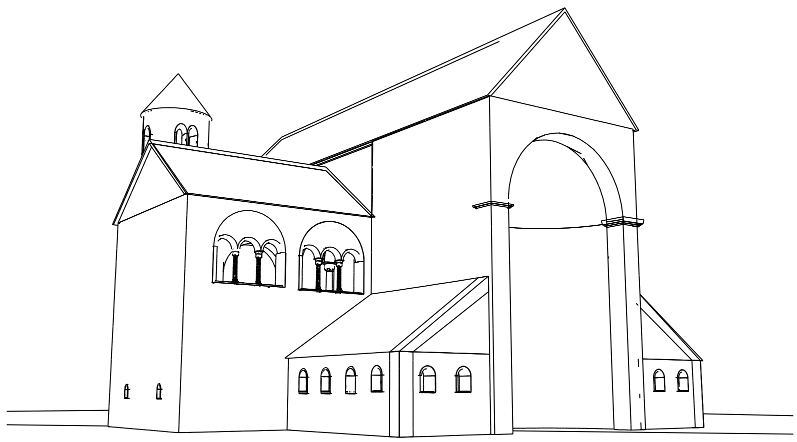


Figure 8: The front of the palace.

Inntrykk fra konferansen SimVis'99

Wolfgang Leister, Norsk Regnesentral

Mars-konferansen i Magdeburg fant sted for tiende gang i år. NORSIGD's fagansvarlig har besøkt konferansen. Under konferansen ble også NORSIGD's prosjekt om grafikk på Web presentert, hvor det fant større interesse enn forventet.

Mars-konferansen i Magdeburg fant sted for tiende gang i år. NORSIGD's fagansvarlig har besøkt konferansen. Under konferansen ble også NORSIGD's prosjekt om grafikk på Web presentert, som ble omfattet med stor interesse.

Totalt fant over 120 personer veien til konferansen. I over tretti foredrag ble det presentert temaer rundt simulering og visualiseringsteknikk. Konferansen ble delt inn i tre parallelle spor, hvorav det ene besto av tutorials og oversiktsforedrag over aktuelle temaer. Her fikk man mulighet til å bli oppdatert innen temaer som man ellers bare følger fra sidelinjen. High-Level-Architecture innen simulering, visualiseringsteknikker og animasjon i brukergrensesnitt var noen av temaene. Simulering på Web var et av hovedtemaene, med noen av de samme problemstillingene som også NORSIGD's prosjekt tok opp. Det ble også diskutert hvordan visuell informasjon bør bli presentert, ikke minst på PDA'er, mobiltelefoner og annet bærbart utstyr. Konferansen ga også muligheten til å diskutere med kolleger og knytte nye kontakter. Til tross for mange temaer i to helt forskjellige områder fremsto konferansen som en helhet. Organisatorene skal ha ros for å ha skapt et miljø for dialog mellom to tilsynelatende forskjellige miljøer. Deltagerne fikk muligheten til å snuse på hverandres arbeidsfelt. Jeg er sikker på at alle deltakerne tok mange nye impulser med seg hjem.

Et konferansebind er utgitt: O. Deussen, V. Hinz, P. Lorenz (Hrsg.): Simulation und Visualisierung'99, SCS–Europe BVBA, Ghent, Belgia, 1999. ISBN 1-56555-131-1.

The CaRo Project

A Robot Guided Camera as an Input Device in Computer Graphics

Alfred Schmitt, Michael Fautz, Peter Oel[2], Universität Karlsruhe

The CaRo project (CaRo = Camera Robot) at the University of Karlsruhe uses a new approach to solve the problem of the acquisition of geometry and surface data of three-dimensional objects. Within the CaRo project a robot arm moves a camera according to the acquisition strategy and points it from various directions towards the object which should be digitized, e.g. a workpiece of mechanical engineering. Using image analysis algorithms the coordinates of surface points of the workpiece can be determined.

Our method has several advantages. The camera can be controlled in an adaptive way by the analysis software. It can easily be switched between global views and detail enlargements. Texture and color as well as the geometry data can be supplied since the surface of the workpiece can be represented using high-resolution color images. The flexibility of the approach becomes clear by looking at the different requests that can be fulfilled. It is even possible to digitize books with cambered pages, which means not getting a three-dimensional representation but getting a textual representation. Furthermore three-dimensional data can be produced. This is needed e.g. in reverse engineering and in advertising and film industry (virtual worlds, computer animations and commercial trick film).

The digitization hardware known so far does not provide the required width and flexibility for object digitization. There is the justified hope that the CaRo approach, the movable camera eye, will become a standard input technique in computer graphics.

In this note we give an overview of the hardware used in the CaRo project and introduce some subprojects. In the next paragraphs the general problem will be described first and a solution will be shown. Then we will focus on two subprojects, object reconstruction and document image analysis.

1 Problem Definition

Whoever is processing graphical data today - especially in the field of 3D - wants to process 3D objects in a flexible way. Concerning the general structure of design systems not only design and manipulation of objects on the graphic display belong to that area, but also import and export functions [9]. By import is meant the recording of the data of objects existing in the real world in order to process them further within the graphics system [2, 7]. For example existing workpieces of mechanical engineering, skeleton bone sections, teeth, plants and leave shapes in biology, and additionally products already existing, such as plates, tools etc. are to be translated in a 3D representation to be used in computer animation and visualization.

The requirements and needs of the users to the imported 3D data are very different. The engineer wants to change the workpiece, thus he must be able to transfer the data into the format, which is needed by his CAD system. The paleontologist wants to measure the pieces of bone exactly, to be able to to draw conclusions from them. With the cooperation of designers and engineers, models are often created at first, whose geometry is then to be transferred to the CAD systems. The computer animator is to produce a commercial video sequence of real products, therefore he needs exact surface textures, to be able to raytrace highly realistic visualizations. While the first case only requires the accurate geometry, the second case needs also information on parameters of the surface as colour, texture, captions etc.

The export function is inverse to the import function, the conversion of 3D data into genuine models, part of which is e.g. the CNC technique and form cutting but other very complex processes as well. Both directions, the input as well as the output of data have always been problematic in graphics processing and are usually solved through specialized processes with a restricted area of application.

In the following we develop a more generalized import function.

[2]email: {aschmitt,fautz,oel}@ira.uka.de

2 Solution

As a technical basis of the CaRo project the following system structure is used: The 3D object to be digitized is placed in the work space of the robot (6 axis, 3 kg load, repeat-accuracy in x-y-z coordinates 0.02 mm, motor: brushless AC servo motor with absolute angle measurement, harmonic drive technique), see figure 1. The robot however is not equipped with a gripping device but guides in place of the gripper a CCD video camera, which supplies a high resolution video signal. The robot can move the camera around the 3D object to view it from all sides, even from above, but of course it cannot view the bottom. The fixed video images thus generated are digitized by a frame-grabber card in full color resolution when needed and are available for the analysis software.

Vision based techniques for object reconstruction are not new. But due to a number of technical problems they have not yet become widely accepted. To be mentioned are image analysis and above all the camera calibration problems, especially determining the exact camera position based only on the picture material. We have already researched the problem how to implement a 3D reconstruction from photos taken from multiple locations at this institute [6] between 1988 and 1991. There were distinct problems with accuracy, especially in deriving the camera locations sufficiently accurate from the image material. It succeeded only roughly, although this is theoretically possible.

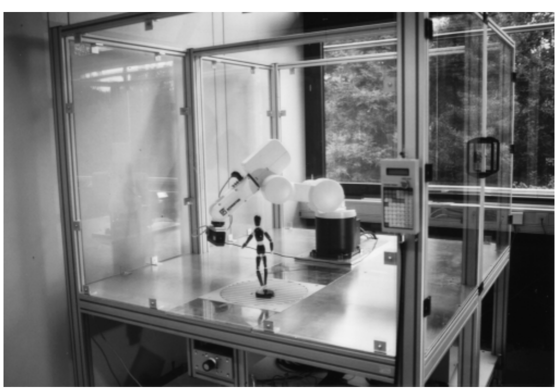


Figure 1: Work space with robot, camera fixed to the gripper of the robot. The model to be digitized, a stylized human puppet, is positioned on the rotation platform, which is sunk into the worktable.

In our approach this problem is solved as the external camera parameters are well-known, because the robot operates in its own fixed coordinate system, so that one knows the camera position as well as the line of sight exactly with each picture taken by the camera.

A further substantial advantage of the CaRo concept arises as a result of the possibility of being able to close up on details of an object with the camera and thus get strongly magnified detail views. With a suitably equipped lens one can almost advance into the microscopic area. Therefore also rather small objects can be scanned. Generally one can say that the resolution of the reconstruction procedure can be given in advance for wide areas and that surfaces can be digitized with exact colouring, which clearly expands the range of application.

With the reconstruction of larger objects the limited work space of the robot has consequences, as the camera cannot view the object from all sides any longer. In such cases the well-known technique of the rotating platform is used. The robot is set up in a suitable distance to the platform. And the object which is to be digitized is fixed on the platform. Now this can be rotated up to an angle of 360 degrees by the controlling software, so that also lateral and rear views are accessible. In particular the silhouette of the object can be obtained.

Worth mentioning as a further advantage is the possibility that the camera can be moved depending on the respective digitization strategy. So we do not have to proceed according to a fixed pattern, but we are able to make use of a variety of strategies depending on the data evaluation and the digitization software [4]. In this way a characteristic point of an object to be digitized can be focused several times and from different views, in order to obtain especially high coordinate accuracy with compensation formulas.

3 Parts of the project

3.1 Technical Environment

As a first part of the project the required hardware was connected and the technical environment (see figure 1) was prepared. In order to develop application software from different workstations, we set up a client-server architecture. Remote computers can get access to the CaRo machinery over the net. Using this architecture we can for example telemanipulate the robot via WWW and the Internet.

3.2 Object reconstruction

One part of the project deals with the reconstruction of threedimensional objects. In contrast to other object scanning techniques as laser range scanners or the structured lighting approach we don't want to get a dense point set of the object to be reconstructed.

We rather want a small data volume and nevertheless a good object approximation. Thereby the CaRo technique is used - the selective use of global views and detail enlargements. Beginning with a simple geometry the object representation gets refined up to the desired quality by adding more images from different views. A small data volume is obtained by this approach as the mesh density of the reconstructed object is correlated with the granularity of the object to be reconstructed. In this way the loss of accuracy by an additional reduction of the measurement data is avoided.

The actual status of the work

In our project up to now we have algorithms implemented for object segmentation from color images and a novel volumetric intersection algorithm which was developed by M. Löhlein in his diploma thesis. As a data structure for 3d object representation a boundary representation (b-rep) is used. In contrast to approaches using a voxel representation for volume intersection one can describe a surface in a geometric way without the loss of accuracy obtained by discretization. As a disadvantage the b-rep intersection algorithm is much more complicated. Actually there are two processing steps to get a coarse approximation of the object to be reconstructed:

In a first step the object to be reconstructed is identified in the color image and its contour is approximated with a polygonal boundary. Holes can also be detected. This silhouette together with the center of projection of the camera builds a generalized cone where the silhouette lines represent its profile and the center of projection the top of the cone.

The second step is to intersect such a cone with the object reconstructed so far. Initially there is used a cube which represents the working area of CaRo. Repeating the two processing steps the cube is intersected with silhouette cones and becomes an approximation of the visual hull [5] of the object. In the following a brief description of the object identifying algorithm and the volumetric intersection method is given.

Object segmentation with color images

The aim is to get polygonal boundaries which represent the contour of the object in the image plane. To get the object and hole regions we assume that every region is enclosed by edges of the image. Therefore an edge detection in the RGB image and in an image resulting from color differences is done. Afterwards the background of the image is identified and colors of the inner of the regions are analyzed and compared to the colors classified from the background. Doing this, object and hole boundaries can be identified. In a finishing step the identified regions are approximated by polygonal boundaries - the silhouette of the object.

The results are amazing. As shown in figure 2 we are able to identify even a black-white checked cup in front of a black-white background.

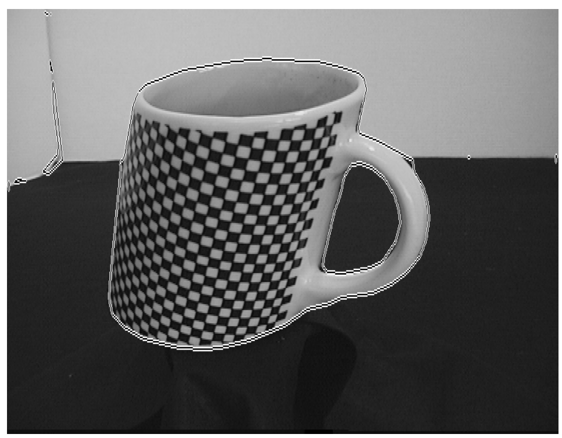


Figure 2: The segmentation result of a black-white coffee cup in front of a black-white background.

Volumetric intersection using a boundary representation

Universal intersection algorithms with b-reps are very difficult to implement. Fortunately we don't have to solve the general b-rep intersection problem. In our case we only intersect a generalized cone with an object given in b-rep which allows us to reduce the 3d intersection problem to a 2d polygon intersection. Two processing steps are needed to solve this problem:

1. Intersection of the 2d silhouette polygons with the projection of the object reconstructed so far to the 2d image plain,

2. closing the gaps that are produced by the intersection step.

Reduction to a 2d intersection problem

To get a 2d problem we first have to project the generalized cone and the current object approximation (the b-rep data structure) to the 2d image plane (see Figure 3). Actually only the object approximation needs to be projected as the cone's contour is already known by the first processing step, the object segmentation.

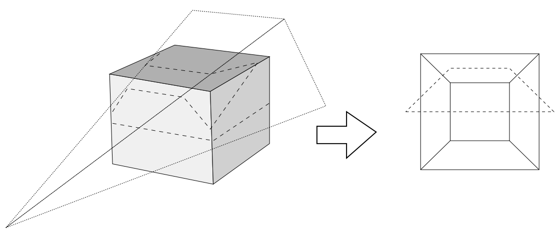


Figure 3: Projection to the 2d plane.

For each polygon which corresponds to a facet of the approximated object an intersection with the cone's contour polygons is done.
Now let O be a polygon of the reconstructed object and C be a cone's polygon; both projected to the image plane of the cone's image. The polygons are oriented so that the inner is located at its right side. This means that O is always oriented CW and C can be oriented CW or CCW (if it's a hole). The intersection of two polygons O, C is done with a well known algorithm [8] as follows:

- Find an edge of O that intersects an edge of C.
- If the O's edge enters the polygon C then beginning with O alternately report the edges of O and the edges of C until the next cut is reached. Otherwise if O's edge leaves the polygon C then start reporting with the edges of C.
- Repeat until all intersection points are processed.

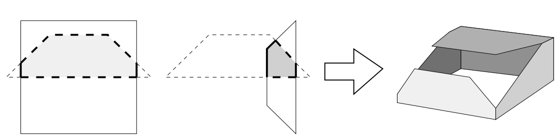


Figure 4: Intersection of the front plane and the right plane of the cube. All resulting polygons are projected back to the object space.

It is possible that no cut appears at all. Then one must check the polygons on inclusion. There are four possible cases:

- there is no inclusion,
- O is included in C,
- C is included in O,
- O and C include one another.

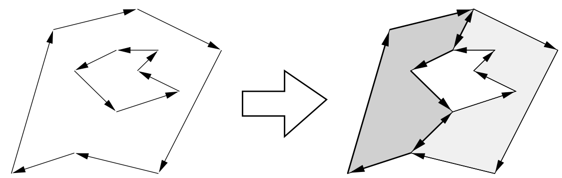


Figure 5: Polygons containing a hole have to been split into two parts.

In the first case the intersection of O and C is empty, in the second and third case the included polygon is the intersection result. The interesting fourth case occurs if C describes a hole and lies inside of O. It is obvious that the result of the intersection must be O without C. To keep the b-rep polygons simple we have to split O and C into two parts and connect each half of O with a half of C so we get two simple polygons as shown in Figure 5.

Now we have to project the resulting polygons of the intersection O, C from the image plane back to the plane defined by O. What we get is a b-rep object with some gaps in it similar to the one shown in Figure 4.

Closing the gaps

Taking a closer look at the intersection result received so far, one can see that the gaps result from the cut by the facets of the generalized cone. Therefore we have to find the parts of each side of the cone which covers the gap. This can be done as follows:

- For each side of the generalized cone collect the edges lying within this side. This can easily be done during the 2d intersection. When we walk along an edge of the cone's polygon then store the corresponding 3d line in a list connected to this side of the cone.

In some cases there are still a few edges missing. These edges are parts of the 3d-cone's edges. In the 2d-projection these edges correspond to the corners of the cone's polygon. In Figure 4 the intersection with the cube's front and top plane shows that case.

- Whenever during the 2d-intersection a corner of the cone's polygon is reached, this corner marks a start or an endpoint of a missing edge. Store all these points sorted by the distance from the center of projection in a list connected to this edge of the cone. The edges can be obtained by taking two successive points.
- Now search for cycles in the edge list connected to each side of the cone. Each cycle forms a polygon which closes one gap in the surface.

Note that some of the cycles found may describe holes. These polygons must be cut from the surrounding polygon as already described in the previous section (see Figure 5).

After this step the intersection of the b-rep and the generalized cone is complete. Figure 6 shows the reconstruction result received by the intersection of the silhouette cones extracted from eight synthesized images.

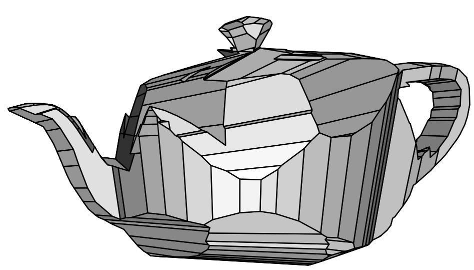


Figure 6: A teapot reconstructed from eight images.

3.3 Document image analysis

Another subproject focuses on using adaptive techniques in document image analysis. The classical approach is divided into four steps:

1. optical scanning and digitization,
2. block segmentation and labeling,
3. processing text and images, if needed: Getting meta information like font styles and layout structure,
4. building a document representation nearly equal to the source document.

Usually step 1, the optical scanning, is done by using a flat scanner. Problems arise when so called "non-flat" pages should be processed, that are documents that can not (e.g. label of a bottle) or must not (e.g. ancient books) lie flat. The usual flat scanner was not designed to scan such documents.

Thinking about a sensible solution the idea to use a robot-guided camera (CaRo) as a scanner was born. Such a camera which has six degrees of freedom has various characteristics: The page of an opened book, which lies usually curved, can be focused from different angles to ensure an orthogonal view of the paper. That of course implies that the final pixel image must be assembled from many overlapping single shootings. The camera may also vary the magnification and thus adapt to the necessary degree of detail depending on the variety of detail of the image content. Summed up, the CaRo hardware is used to do the optical scanning.

We want to use the flexibility provided by CaRo not only in step 1, but also in step 2 to step 4. For example: At a lower resolution level less storage space is needed and algorithms run faster. Currently realized are solutions for step 2 and a first version for step 3.

Block segmentation and classification is done at a very low resolution of about 60 dpi using a combination of Fletcher and Kasturi's connected component analysis [3] with the technique introduced by Wahl, Wong, and Casey [10]. See Figure 7 for some results. The actual work tries to take more advantage of multiresolution methods like the ones discussed by Cinque, Lombardi, and Manzini in [1].

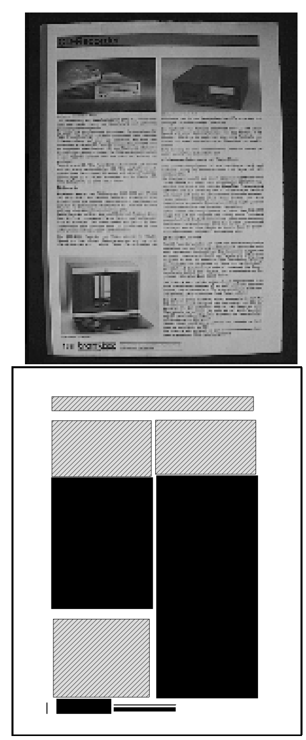


Figure 7: Original document image and the block

segmentation and classification result. The darker blocks represent text regions, the lighter ones images.

Actual work focuses on finding an adaptive solution for the optical character recognition (OCR) problem of step 3. We use an adaptive technique, where the first recognition is done at a lower resolution. Problematic characters are looked at more closely and at a higher resolution. Thus one arrives at a far-reaching analogy to the reading eye, humans fix hardly recognizable items by closing up, too.

Conclusion

Summed up, the approach introduced here has a number of important advantages, which let us hope that we found a flexible solution for the import problem of graphical data processing. The research project however is of a remarkable complexity and demands extensive experimental works as well as the development of new software architectures to obtain reliable statements about suitability in practical work.

Referanser

[1] L. Cinque, L. Lombardi, and G. Manzini. A multiresolution approach for page segmentation. *Pattern Recognition Letters*, 19:217–225, 1998.

[2] O. Faugeras. *Three-Dimensional Computer Vision, a Geometric Viewpoint.* MIT Press, Cambridge, Massachusetts, 1993.

[3] Lloyed Alan Fletcher and Rangachar Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[4] R. A. Jarvis. A perspective on range finding techniques for computer vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 122–139, 1983.

[5] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[6] W. Leister. *Geometrisches Modellieren durch interaktive Rekonstruktionsmethoden.* PhD thesis, Fakultät für Informatik, Universität Karlsruhe, 1991.

[7] B. Schunck R. Jain, R. Kasturi. *Machine vision.* McGraw-Hill, New York, 1995.

[8] A. Schmitt, O. Deussen, and M Kreeb. *Einführung in graphisch-geometrische Algorithmen.* Teubner, Stuttgart, 1995.

[9] A. A. Schmitt. *Dialogsysteme - Kommunikative Schnittstellen, Software-Ergonomie und Systemgestaltung.* BI Wissenschaftsverlag, Mannheim, 1983.

[10] Friedrich M. Wahl, Kwan Y. Wong, and Richard G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20:375–390, 1982.

Hva er NORSIGD?

NORSIGD – Norsk samarbeid innen grafisk databehandling – ble stiftet 10. januar 1974. NORSIGD er en ikke-kommersiell forening med formål *å fremme bruken av, øke interessen for, og øke kunnskapen om grafisk databehandling i Norge*.

Foreningen er åpen for alle enkeltpersoner, bedrifter og institusjoner som har interesse for grafisk datbehandling. NORSIGD har per januar 1998 44 institusjons- og 12 personlige medlemmer. Medlemskontingenten er 1.000 kr per år for institusjoner. Institusjonsmedlemmene er stemmeberettiget på foreningens årsmøte, og kan derigjennom påvirke bruken av foreningens midler.

Personlig medlemskap koster 250 kr per år. Personlige medlemmer får tilsendt medlemsbladet *NORSIGD Info*. Kontingenten er redusert til 150 kr ved samtidig medlemskap i vår europeiske samarbeidsorganisasjon *Eurographics*.

Alle medlemmer får tilsendt medlemsbladet *NORSIGD Info* 3–4 ganger per år.

Interesseområder

NORSIGD er et forum for alle som er opptatt av grafiske brukergrensesnitt og grafisk presentasjon, uavhengig av om basisen er *The X window System*, *Microsoft Windows* eller andre systemer. NORSIGD arrangerer møter og seminarer, formidler informasjon fra internasjonale fora og distribuerer fritt tilgjengelig programvare. I tillegg formidles kontakt mellom brukere og kommersielle programvareleverandører.

NORSIGD har lang tradisjon for å støtte opp om bruk av datagrafikk. Foreningen bidrar til spredning av informasjon ved å arrangere møter, seminarer og kurs for brukere og systemutviklere.

GPGS

GPGS er en 2D- og 3D grafisk subrutinepakke. GPGS er maskin- og utstyrsuavhengig. Det vil si at et program utviklet for et operativsystem med f.eks. bruk av plotter, kan flyttes til en annen maskin hvor plotteren er erstattet av en grafisk skjerm uten endringer i de grafiske rutinekallene. Det er definert grensesnitt for bruk av GPGS fra FORTRAN og C.

Det finnes versjoner av GPGS for en rekke forskjellige maskinplattformer, fra stormaskiner til Unix arbeidsstasjoner og PC. GPGS har drivere for over femti forskjellige typer utsyr (plottere, skjermer o.l.). GPGS støtter mange grafikkstandarder slik som Postscript, HPGL/2 og CGM. GPGS er fortsatt under utvikling og støtter stadig nye standarder.

GPGS eies av NORSIGD, og leies ut til foreningens medlemmer.

Eurographics

NORSIGD samarbeider med Eurographics. Personlige medlemmer i NORSIGD får 20 SFr rabatt på medlemskap i Eurographics, og vi formidler informasjon om aktuelle aktiviteter og arrangementer som avholdes i Eurographics-regi. Tilsvarende får Eurographics medlemmer kr 100 i rabatt på medlemskap i NORSIGD.

Eurographics ble grunnlagt i 1981 og har medlemmer over hele verden. Organisasjonen utgir et av verdens fremste fagtidsskrifter innen grafisk databehandling, *Computer Graphics Forum*. *Forum* sendes medlemmene annen hver måned. Eurographics konferansen arrangeres årlig med seminarer, utstilling, kurs og arbeidgrupper.

NORSIGD
v/ Reidar Rekdal
DNV Software
Postboks 300
1322 HØVIK

Returadresse:
NORSIGD v/ Reidar Rekdal
DNV Software
Postboks 300
1322 HØVIK

Styret i NORSIGD 1998

Funksjon	Adresse	Telefon	email
Leder	Ketil Aamnes ViewTech AS PB 47 Pirsenteret 7005 TRONDHEIM	73 54 61 23 (direkte) 73 54 61 44 (fax)	Ketil.Aamnes @viewtech.no
Fagansvarlig	Wolfgang Leister Norsk Regnesentral Postboks 114 Blindern 0314 OSLO	22 85 25 78 (direkte) 22 85 25 00 (sentralbord) 22 69 76 60 (fax)	leister@online.no
Sekretær	Reidar Rekdal Det Norske Veritas Software Postboks 300 1322 HØVIK	67 57 73 18 (direkte) 67 57 72 50 (sentralbord) 67 57 72 72 (fax)	reidar.rekdal @dnv.com
Styremedlem	Gisle Fiksdal MARINTEK A.S Postboks 4125, Valentinlyst 7002 TRONDHEIM	73 59 59 07 (direkte) 73 59 57 76 (fax)	Gisle.Fiksdal @marintek.sintef.no
Varamedlem	Svein Taksdal Norges Vassdrags- og Energiselskap Hydrologisk Avdeling, Seksjon data Postboks 5091, Majorstua 0301 OSLO	22 95 92 86 (direkte) 22 95 92 01 (fax)	svein.taksdal @nve.no
Varamedlem	Rune Torkildsen Autograph Broadcast Systems AS Postboks 2 5002 BERGEN	55 90 81 40 55 90 80 70 (sentralbord) 55 90 80 90 (fax)	Rune.Torkildsen @tv2.no

Svarkupong

- ◯ Innmelding – institusjonsmedlem
- ◯ Innmelding – personlig medlem
- ◯ Innmelding – Eurographics medlem
- ◯ Ny kontaktperson
- ◯ Adresseforandring

Navn: ....
Firma: ....
Gateadresse: ....
....
Postadresse: ....
....
Postnummer/sted: ....
....
Telefon: ....
Telefaks: ....
email: ....