

# 8

## Zusammenfassung und Ausblick

### 8.1 Zusammenfassung

Im Rahmen dieser Diplomarbeit sollte ein System zur Simulation eines multi-Kamera-basierten Gesichtsscanners entwickelt werden. Ein Vergleich der im vorangehenden Kapitel demonstrierten Resultate mit der Aufgabenstellung dieser Diplomarbeit zeigt, dass die meisten verlangten Ziele erreicht werden konnten:

- Es steht eine objektorientierte C++-Schnittstelle als Scanner-API zur Steuerung des virtuellen Scanners zur Verfügung, die gleichzeitig auch als Schnittstelle zum endgültigen Scanner verwendet werden kann.
- Der Scanner kann aus einer beliebigen Anzahl frei anzuordnender Kameras aufgebaut werden.
- Die intrinsischen Kalibrierungs-Eigenschaften der Kameras sind frei definierbar.
- In einem Scannersystem können gleichzeitig unterschiedliche Kameratypen verwendet werden.
- Beliebig viele, einzeln steuerbare Punktlichtquellen unterschiedlicher Farbe können frei im Scanner positioniert werden.
- Mit einem virtuellen Drehteller kann das aufzunehmende Objekt beliebig um die vertikale Achse gedreht werden.
- Die Bilder des virtuellen Scanners werden mit Alias gerendert und in einem Nachbearbeitungsschritt durch das Einfügen von diversen Kamera-Artefakten möglichst naturgetreu gemacht. Die unterstützten Artefakte sind:
  - Radiale Verzerrung
  - Vignetting
  - Blooming
  - Bayer-Pattern
- Der Aufbau als verteiltes System mit einem Scanner-Client und mehreren Kamera-Servern erlaubt ein effizientes Rendering durch gleichzeitige Ausnutzung der Rechenkapazität mehrere Rechner in einem Verbund.

- Um bereits gerenderte Bilder nicht mehrmals zu berechnen, steht ein Cache-Mechanismus zur Verfügung.

Die Tatsache, dass das entwickelte System in Teilen bereits zum jetzigen Zeitpunkt verwendet wird, unterstützt die Aussage, dass das Resultat den Anforderungen grösstenteils gerecht wird. Nicht vollständig erreicht werden konnte der Aspekt der Systemunabhängigkeit: Bedingt durch die vorhandenen Alias-Lizenzen kann der Simulator ausschliesslich unter IRIX betrieben werden. Zusätzlich beschränkt die Verwendung der Prozessgenerierungsroutine `fork` an mehreren Stellen im Code den Gebrauch des gesamten Systems auf UNIX und seine Derivate, wodurch die gesamte Windows-Welt aussen vor bleibt.

## 8.2 Ausblick

Um mit dem beschriebenen System wirklich realistische Bilder erstellen zu können, müssen die Feinheiten diverser Parameter, welche Alias betreffen, durch Tests evaluiert werden. Eventuell müssen auch weitere der im Kapitel 4.1 beschriebenen Artefakte implementiert werden, um reale Kameras ausreichend zu approximieren. Ausserdem liesse sich an diversen Stellen im Code durch Optimierungen eine bessere Performanz erreichen. Ob sich dieser Aufwand allerdings lohnt, ist wiederum unklar, denn der Zeitaufwand für die Nachbearbeitung der gerenderten Bilder steht in keinem Verhältnis zur von Alias für das Rendering beanspruchten Dauer.

Damit der Scanner-Client auch unter Windows betrieben werden kann, müssen die Aufrufe von `fork` ersetzt werden. Diese werden neben dem Aufruf von Alias verwendet, um den C-Präprozessor zur Ersetzung von Einbindungen und Definitionen in Konfigurationsdateien aufzurufen. Die von diesem Programm genutzte Funktionalität könnte aber mit wenig Aufwand direkt als Teil der Scanner-API implementiert werden. Eine Portierung des Simulators nach Linux oder Windows ist mangels einer entsprechenden Alias-Lizenz nicht möglich, weshalb auch der entsprechende `fork`-Aufruf nicht ersetzt werden muss.

Das Simulations-System wird in einer ersten Phase des Gesichtsscanner-Projektes an der ETH in Zürich (Kapitel 1.2) für das Testen und die Evaluation von Rekonstruktionsalgorithmen verwendet werden. Gleichzeitig können dabei verschiedene Konfigurationen ausprobiert werden, um später beim Bau des realen Gesichtsscanners die optimale Anordnung der Kameras und der Lichtquellen zu kennen. Die Scanner-API wird die Steuerung dieses noch zu bauenden Scanners ohne grössere Anpassungen erlauben.

# Danksagung

Während dieser vielseitigen und damit äusserst interessanten Diplomarbeit konnte ich einerseits Kenntnisse der Programmiersprache C++ und der Programmierung unter UNIX vertiefen, andererseits erlaubten der objektorientierte Ansatz und das entsprechende Design des Software-systems eine Vertiefung in die Objektorientierte Programmierung. Das Erlernen dieses Rüstzeugs der modernen Programmierung wurde für mich zu einer wichtigen Erfahrung und wird mir bestimmt im nun folgenden Berufsleben oft helfen.

Ich möchte an dieser Stelle ganz herzlich meinem Betreuer Tim Weyrich für seine grosse Bereitschaft danken, mir zu jedem Zeitpunkt bei Verständigungs- oder Implementationsproblemen Hilfe zu leisten. Dank seinem umfangreichen Wissen konnten die allermeisten Probleme innert kürzester Zeit erfolgreich gelöst werden. Diese Unterstützung war massgeblich für den Erfolg dieser Diplomarbeit verantwortlich. Zusätzlich konnte ich seine grosse Begeisterung für sein Gesichtsscanner-Projekt als Motivation nutzen, eine für ihn brauchbare Arbeit zu leisten.

Ein zweiter, nicht minder grosser Dank geht an meine Freundin Nicole Tobler. Sie hat während diesen vier Monaten meine Launen ertragen müssen und mir viel mentale Unterstützung gegeben. Gar nicht selbstverständlich ist auch, dass sie sich während ihrer Diplomprüfungsvorbereitung die Zeit nahm, mich liebevoll durch eine Grippe zu pflegen. Viel Erfolg bei Deiner Diplomarbeit in den nächsten Monaten.

Nicht vergessen möchte ich meine Eltern Astrid und Werner Nef, die mein Studium durch ihre Unterstützung erst möglich gemacht haben. Vielen Dank für alles, was Ihr mir gegeben habt.

Marco Nef, 13. März 2002

```
<mnef@vermeer:~> cd DA
<mnef@vermeer:DA> find . \( -name '*.cpp' -o -name '*.c' -o -name '*.h' \) | xargs wc
      16225      46763      608146 total
<mnef@vermeer:DA> yahoo
```

