

Anja Schäfer – Heike Leitte – Hans Georg Bock

Ein automatisches und virtuelles 3D-Puzzle von Khmer-Tempeln im Angkorstil

Zusammenfassung

In Banteay Chhmar, Kambodscha, befindet sich eine der größten Tempelanlagen weltweit. Jedoch sind von vielen Mauern heute nur noch Steinhäufen vorhanden, viele scheinbar intakte Mauern benötigen dringend ein neues Fundament. Ein neu entwickeltes Verfahren zur automatisierten und virtuellen Rekonstruktion einer Mauer auf der Grundlage von 3D-Scans gewährleistet, dass solche annähernd quaderförmigen Steine beim Wiederaufbau möglichst wenig bewegt werden müssen. Im ersten Schritt wird jedes hochauflösende 3D-Steinmodell auf die wesentlichen Merkmale, die Ecken, Kanten und Seitenflächen reduziert. Das Verfahren ermittelt auf Grundlage des Winkels zwischen benachbarten Flächen Kombinationsmöglichkeiten von Mauersteinen. Schließlich erfolgt im dritten Arbeitsschritt die Validierung dieser Vorschläge durch den Experten.

Keywords: Virtuelles 3D-Puzzle; Angkorzeit; *linear least squares*; *minimal volume box*; Krümmungsmodell; Kambodscha; 3D-Modell.

One of the largest temple complexes in the world is located in Banteay Chhmar, Cambodia. However, many of the temple walls are currently only piles of stone blocks and many of the seemingly intact walls are in urgent need of new foundations. A new method was developed for the automated virtual reconstruction of a stone wall on the basis of 3D scans allowing to limit the moving of the nearly cube-shaped stones to a minimum during reconstruction work. In a first step, each high-resolution 3D stone model is reduced to its essential features, namely vertices, edges and faces. The method then suggests pairs of fitting stone blocks on the basis of the angle between adjacent surfaces. Finally, the expert has to validate the suggested fitting in a third step.

Keywords: Virtual 3D puzzle; Angkor period; linear least squares; minimal volume box; curvature model; Cambodia; 3D model.

Dank gebührt Dr. Pheakdey Nguonphan für seine Hilfe und Unterstützung während unseres Aufenthaltes in Kambodscha. Darüberhinaus gilt er dem Global Heritage Fund und dem kambodschanischen Ministerium für Kultur und Kunst. Diese Arbeit ist Teil der IWR Pioneering Projects und wird gefördert durch die HGS MathComp – DFG Graduate School 220.

1 Einleitung

In Banteay Chhmar, Kambodscha, befindet sich eine der weltweit größten Tempelanlagen mit einem Areal von ca. 9 km². Erbaut wurde sie während der Angkorzeit unter König Jayavarman VII (ca. 1180–1240 n. Chr.). Die Anlage besteht aus einem Haupttempel und acht Satellitentempeln.¹ Aufgrund seiner abgeschiedenen Lage (s. Abb. 1) ist der Tempel durch Kultur- und Steinräuber sowie durch starken Zerfall bedroht. Zwar wurden die Steine des Tempels vor Ort nicht als neues Baumaterial genutzt, da die Steine jedoch nicht mit Mörtel verbunden sind, können heute Pflanzen in die Zwischenräume vordringen. Somit liegen weite Teile des Komplexes nur noch als Trümmerhaufen vor (s. Abb. 2).

Gemeinsam mit dem Global Heritage Fund waren wir 2010 in einem mehrmonatigen Projekt vor Ort, um verschiedene Teile dieser Anlage mit Hilfe unseres 3D-Scanners² digital zu erfassen.³ Unter anderem waren dies die bekannte Avalokeshvara⁴ (Abb. 2 und Abb. 3) sowie einzelne Steine einer noch intakten Mauer (Abb. 4). Eben diese Mauer benötigte neue Fundamente, um sie vor weiterem Verfall zu bewahren, weshalb sie zerlegt werden musste. Dies ermöglichte es uns, 135 einzelne Steine einzuscannen und 3D-Modelle jedes einzelnen Steins zu erzeugen (vgl. Abb. 5).

Eine virtuelle Rekonstruktion von Khmer-Tempeln im Angkorstil, bei denen die einzelnen Steine nur als Trümmerhaufen vorliegen (vgl. Abb. 2), ist mit einem virtuellen 3D-Puzzle zu vergleichen.⁵ Aufbauend auf dem in Banteay Chhmar gewonnenen Datensatz kann nun ein computergestützter Algorithmus zur Lösung dieses Puzzles entwickelt werden. Die Tatsache, dass die korrekte Anordnung der Steine im vorliegenden Fall bekannt ist, hilft dabei, die Ergebnisse dieses Algorithmus zu verifizieren.

1 Higham 2004, s. v. Banteay Chhmar, sowie Chandler 2008 und Groslier 1937.

2 Ein Breuckmann SMART-Scan HE inkl. Software. Der Scanner basiert auf zwei Methoden: Streifenlicht wird mit *Stereo Vision* gekoppelt und die mitgelieferte Software kann jeden Scan direkt als trianguliertes *Mesh* darstellen.

3 Freudenreich u. a. 2011; Schäfer u. a. 2011.

4 Auch unter anderen Schreibweisen bekannt, zum Beispiel. Avalokiteshvara, Lokeshvara, Lokiteshvara.

5 Auch bei einem Puzzle liegen die Ausgangsdaten (zum Beispiel Puzzleteile) ungeordnet vor und die richtig zusammenpassende Lösung ist gesucht.

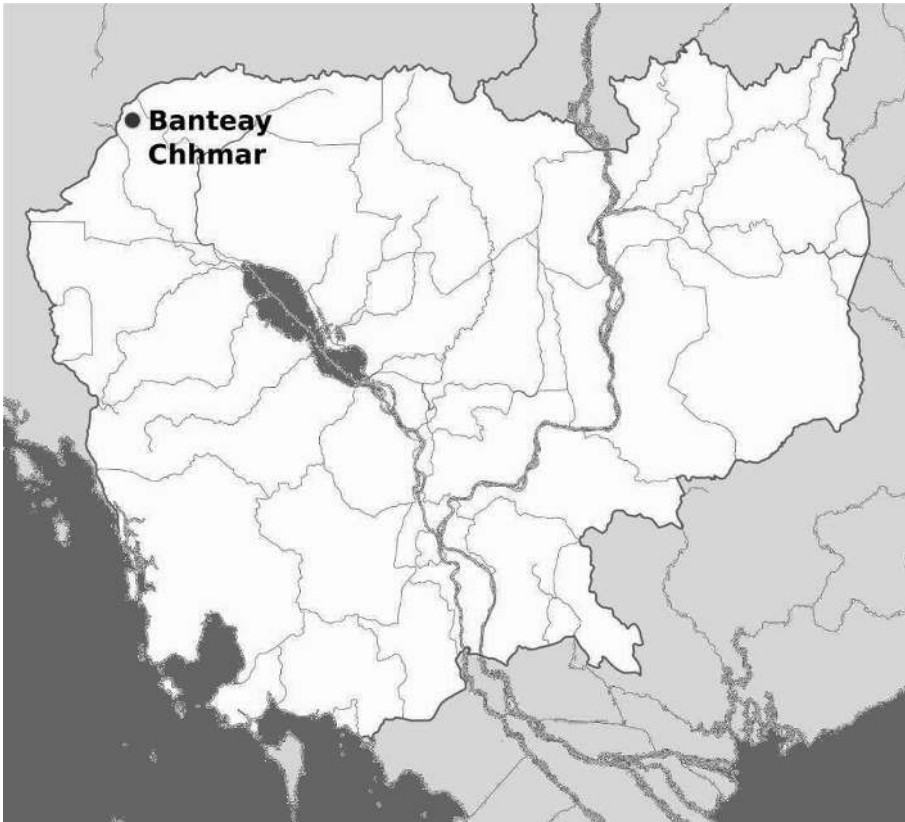


Abb. 1 Eine der bekanntesten Tempelanlagen, erbaut durch König Jayavarman VII, befindet sich in Banteay Chhmar, Kambodscha. Die Karte Kambodschas zeigt die Lage innerhalb des Landes.

Der vorliegende Beitrag diskutiert die Vorzüge einer virtuellen Rekonstruktion und stellt manuelle, am Computer erzeugte Lösungen vor. Danach stellen wir dar, welche Herausforderungen bei 3D-Puzzles zu bewältigen sind und erläutern die Grundzüge des von uns entwickelten computergestützten Algorithmus.

2 Rekonstruktion

2.1 Manueller Wiederaufbau

Warum ist virtueller Wiederaufbau sinnvoll? Die in Banteay Chhmar benutzten Steine wiegen je zwischen 100 kg und 1000 kg und sind daher nur schwer zu bewegen. Dementsprechend muss ein Stein immer von mehreren Arbeitern gleichzeitig mühe-



Abb. 2 Aktueller, zerfallener Zustand der von Jayavarman VII erbauten Tempelanlage.

voll bewegt werden, wodurch sich die Gefahr einer weiteren Beschädigung signifikant erhöht. Darüber hinaus ist die Sicherheit der einzelnen Arbeiter bei dieser schweren körperlichen Arbeit nicht gewährleistet. Hinzu kommt, dass die örtlichen Gegebenheiten (Zustand der Zufahrtsstraßen, Tierwelt, Klimabedingungen, Anfahrtszeit) den Zugang zum Tempel häufig erschweren.

Bisher wird die Rekonstruktion des Tempels manuell durchgeführt, indem ein Expertenteam jeden Stein analysiert (zum Beispiel Relief, Größe, Form) und einen passenden Platz an der wiederaufzubauenden Stelle bestimmt. Die Überlegungen hierzu basieren auf Erfahrungswerten und dem Wissen über die Kultur der Khmer. Dann werden die Steine per Hand und Kran an ihren Platz befördert, wodurch sich jedoch der Zerfall der Steine beschleunigen kann.

2.2 Virtueller Wiederaufbau

Demgegenüber hat der virtuelle Wiederaufbau wesentliche Vorteile. Die Steine müssen vor Ort nur noch zweimal bewegt werden: einmal zum Digitalisieren und einmal für den Wiederaufbau. Am Computer können sie zwischenzeitlich so oft wie nötig verscho-



Abb. 3 Die berühmteste der ehemals acht Avalokeshvaras.



Abb. 4 Die verbleibenden zwei noch stehenden Avalokeshvaras in Banteay Chhmar.



Abb. 5 Die östliche Mauer in ihrem ursprünglichen Zustand. Die Steine, im Bild durch die roten Linien begrenzt, wurden digital vermessen.

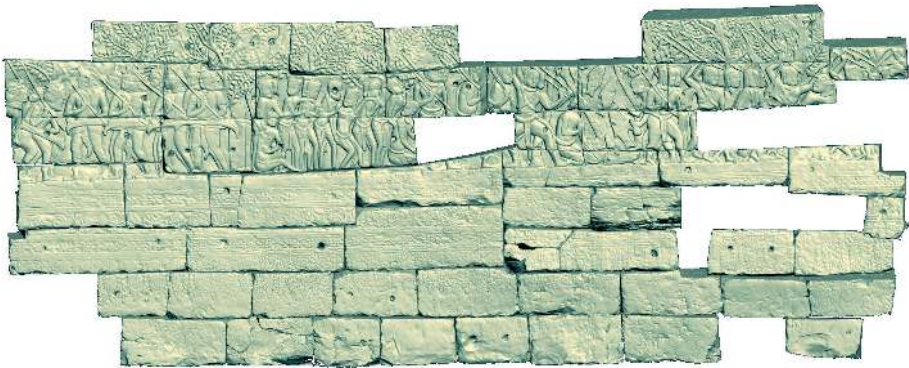


Abb. 6 Erster manueller virtueller Wiederaufbau.

ben oder gedreht werden. Zusätzlich bietet der computergestützte Dokumentation der Steine die Grundlage dafür, Algorithmen zu entwickeln, die die Passgenauigkeit jeder betrachteten Lösung wesentlich genauer und schneller berechnen können. Damit kann ein Puzzle zusammengesetzt werden, ohne die Menschen vor Ort in Gefahr zu bringen, während gleichzeitig das Kulturgut geschützt bleibt.

Die hier vorgestellte Arbeit soll auch dabei helfen, andere Tempel aus dieser Zeit zu rekonstruieren. Denn Tempel der Angkorperiode haben aufgrund ähnlicher Bauepochen oder gleicher Bauherren viele strukturelle Gemeinsamkeiten. Das bedeutet, dass mit unserer Hilfe Archäologen in der Lage sein werden, sowohl einen virtuellen als auch einen realen Wiederaufbau solcher Tempel durchzuführen.

Zunächst soll durch einen manuellen virtuellen Wiederaufbau herausgefunden werden, wo die Grenzen von Mensch und Maschine liegen. Dinge, die ein Mensch intuitiv erkennt, müssen einem Computer erst beigebracht werden, dagegen führt der Computer die Berechnungen deutlich schneller durch. Anhand der beiden manuellen Er-

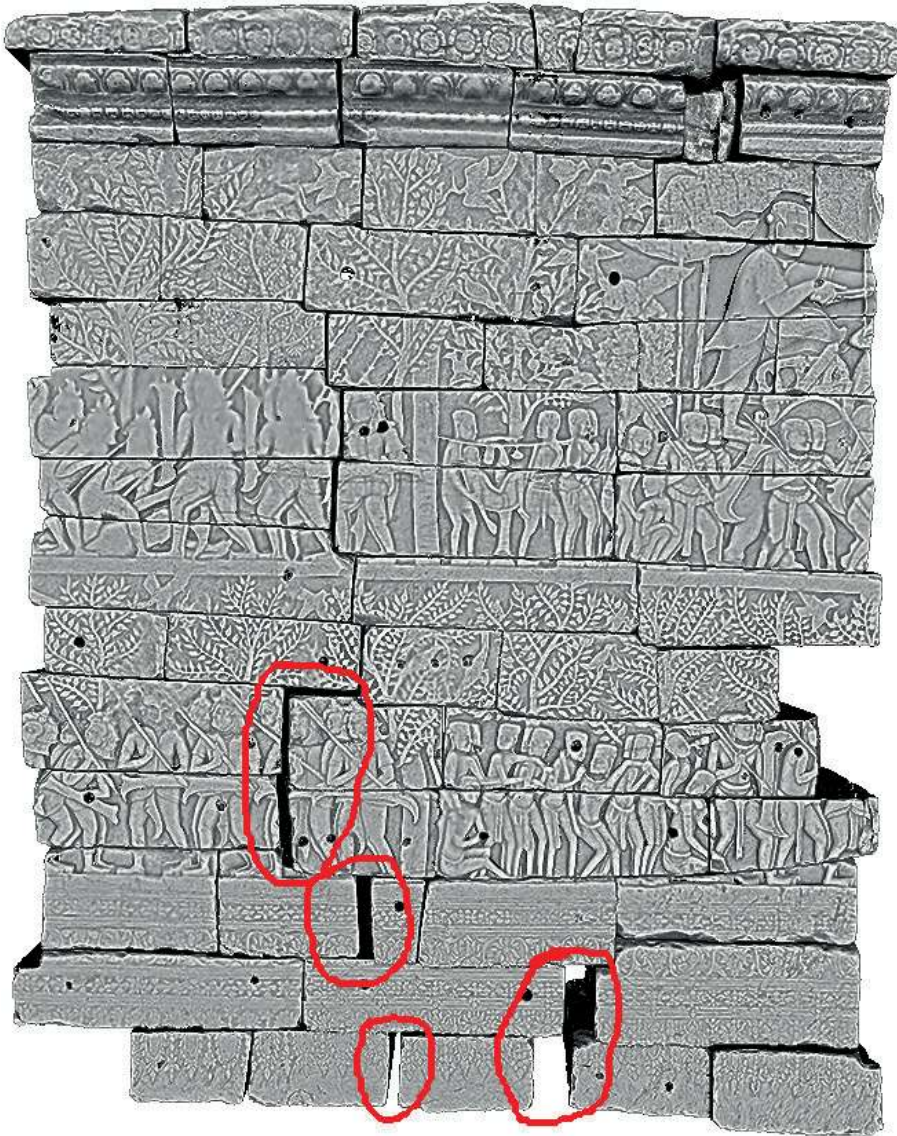


Abb. 7 Virtueller Wiederaufbau mit 65 Steinen. In rot umrandete Lücken zeigen sogenannte Fortsetzungsfehler.

gebnisse in Abb. 6 und Abb. 7 soll erläutert werden, worauf bei einem automatischen 3D-Puzzle zu achten ist. Das in Abb. 6 gezeigte Modell mit 51 von Hand am Computer gepuzzelten Steinen hat eine Dateigröße von ca. 2 GiB (Ausschnitt des in Abb. 5 gezeigten Mauerbereichs, der gescannt wurde).⁶ Dies erschwert das Laden, Verändern oder Hinzufügen weiterer Steine massiv. Zunächst ist daher zu berücksichtigen, dass auch bei einer großen Anzahl von 3D-Objekten die Dateigröße keine Einschränkung darstellt. Des Weiteren sind bei einem manuellen Aufbau deutlich ‚Löcher‘ zu sehen, das heißt Stellen, an denen Steine fehlen.⁷ Somit stellt sich in einem zweiten Schritt die Frage, wie mit den Lücken zu verfahren ist.

Abb. 7 zeigt einen von Hand am Computer in ca. 40 Arbeitsstunden erstellten virtuellen Aufbau von 65 Steinen. Auch hier beträgt die Dateigröße ca. 2 GiB, wodurch die oben beschriebenen Schwierigkeiten auftreten. Das Modell in Abb. 7 ist außerdem ein Beispiel für eine weitere Fehlerquelle, den sog. Fortpflanzungsfehler. Kleinste Unstimmigkeiten beim Zusammensetzen im oberen Teil, die mit bloßem Auge nicht zu erkennen waren, führten in der Summe zu den Fehlern bei der Zusammenpassung der untersten Steine. Die rot umrandeten Bereiche zeigen Lücken auf, die im ursprünglichen Zustand nicht vorkamen.

Beide Modelle wurden mit Hilfe einer Musterlösung erstellt, doch der gesuchte, computergestützte Algorithmus soll die richtige Zusammenstellung ohne Vorkenntnisse finden. Eine zusätzliche Anforderung betrifft die Effizienz, das heißt der Algorithmus soll einen (wesentlichen) zeitlichen Vorteil gegenüber dem manuellen Puzzeln an Ort und Stelle haben.

3 Herausforderungen

3D-Puzzle sind seit mittlerweile über zehn Jahren Gegenstand intensiver Forschung und ständiger Neuerungen. Hervorzuheben sind auf diesem Gebiet die Arbeiten der Wiener Arbeitsgruppe unter Helmut Pottmann.⁸ Ausgehend von einem zerbrochenen Objekt, klassifiziert ihr Ansatz die Oberflächen der Bruchstücke als ‚Bruchfläche‘ und ‚Nichtbruchfläche‘ auf Basis von sog. Integralinvarianten.⁹ Reuter u. a. stellen einen semiautomatischen Ansatz vor: Um zwei vorgegebene passende Teile möglichst nahe zusammen

6 GiB ist die Abkürzung für Gibibyte, wobei 1 GiB = 2^{30} Byte, im Gegensatz zu einem Gigabyte 1 GB = 10^9 Byte.

7 Die Gründe dafür reichen vom tatsächlichen Fehlen des Steins (z. B. zu starker Zerfall oder Raub) bis hin zum Datenverlust.

8 Huang u. a. 2006.

9 Bezugsgrößen, die sich auch bei Integration nicht ändern. Integrale werden zur Flächenberechnung eingesetzt, wobei Invarianten von der Integration unabhängige Aussagen über die Fläche treffen können. Vgl. hierzu: Pottman u. a. 2009.

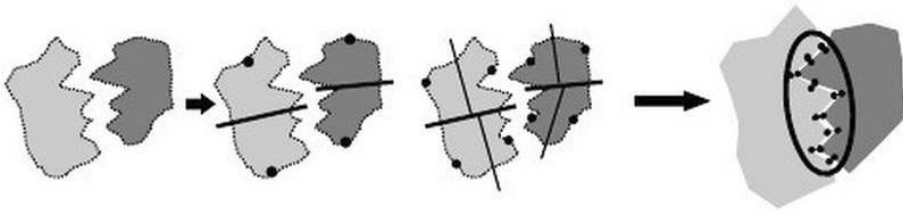


Abb. 8 Exemplarische vereinfachte Darstellung zweier zusammenpassender Bruchstücke. Jede Kontaktfläche hat ein nahezu perfekt passendes Gegenstück.

zu setzen, kommt der *Iterative Closest Point Algorithmus*¹⁰ zum Einsatz.¹¹ Eine weitere Alternative nutzt *cluster trees*,¹² wobei eine Baumstruktur die verschiedenen Möglichkeiten Bruchstücke zu puzzeln speichert. Der Algorithmus arbeitet sich in die Tiefe des Baumes, prüft dabei iterativ die Qualität der Passgenauigkeit und geht entweder in die gleiche Richtung weiter vor oder wählt eine andere.

Alle Ansätze gehen davon aus, dass ein zerbrochenes, ehemals als Ganzes vorliegendes Objekt¹³ wieder zusammengesetzt werden soll. Dabei wird die Unterscheidung zwischen Bruch- und Oberflächenkanten ausgenutzt.¹⁴ Jede Bruchkante besitzt eine Art Gegenstück, so dass beide Teile nahezu perfekt zueinander passen, wie in Abb. 8 deutlich zu sehen ist. Auch für Teilstücke, deren Oberflächenstruktur beispielsweise durch Erosion teilweise verloren ging, können aufgrund dieser Eigenschaft noch passende Gegenstücke gefunden werden.

Im vorliegenden Fall ist die Ausgangssituation jedoch anders, denn die in einem Steinbruch gewonnenen Steine mit nachbearbeiteten Oberflächen weisen meist nur Oberflächenkanten auf. Falls Bruchkanten auftreten, ist der Stein beschädigt. Die spezielle Verarbeitung direkt beim Einbau (s. Abschnitt 4 „Modellsimplifikation durch minimale Volumina“) sorgt dafür, dass auf jeder möglichen Kontaktfläche Vertiefungen entstehen, die nicht durch Ausbuchtungen im entsprechenden Gegenstück gefüllt werden (Abb. 9). Somit wäre eine Anpassung allein aufgrund der Beschreibung der Oberflächen in unserem Fall nicht zielführend. Hinzu kommt, dass bisherige Puzzleansätze mit einer geringen Anzahl (weniger als 50) von Bruchstücken arbeiten, so dass ein sogenannter *brute-force-Algorithmus* einsetzbar ist, der so lange mögliche Kombinationen

10 Hierbei werden zwei Flächen in Verbindung gebracht, indem zueinandergehörige Punkte gesucht und miteinander identifiziert werden. Ursprünglich von Besl und McKay (s. Besl und McKay 1996) entwickelt, mittlerweile Standard im Bereich des surface matchings.

11 Reuter u. a. 2007.

12 Ein *cluster tree* ist ein Baum (Graph), dessen Knoten aus *clustern* (Menge von Objekten mit ähnlichen Ei-

genschaften) bestehen. Für weitere Informationen seien Standardwerke zur Graphentheorie empfohlen, zum Beispiel Diestel 2010; Winkelbach und Wahl 2008.

13 Vgl. Huang u. a. 2006, Abb. 1 unten.

14 Hierbei wird die Rauigkeit der Oberfläche als Indikator genommen, um zwischen Bruchkanten und Oberflächenkanten zu unterscheiden.

von Teilstücken vergleicht, bis die richtige gefunden ist. Da die Anzahl der Möglichkeiten quadratisch mit der Menge an Objekten steigt, ist dieses Verfahren für größere Datenmengen praktisch nicht mehr anwendbar.

Anschaulich ist das am Banteay-Chhmar-Datensatz zu sehen mit seinen 135 einzelnen Steinen. Angenommen, jeder Stein hat eine quaderähnliche Form ohne Einkerbungen bei dem nur die Seitenflächen, nicht aber die Vorder- und Rückseite, mit anderen Steinen in Kontakt sein können. Das bedeutet, dass jeder Stein mindestens vier mögliche Kontaktflächen besitzt und somit ein *brute-force*-Algorithmus

$$2(n^2 - n) = 2(135^2 - 135) = 36180$$

paarweise Kombinationsmöglichkeiten überprüfen muss.¹⁵ Im Abschnitt 4 wird aufgezeigt, dass es jedoch Steine gibt, die mehr als nur vier mögliche Kontaktflächen besitzen, wodurch sich die Anzahl der Kombinationen deutlich erhöht. Darüber hinaus sollen in einem zukünftigen Projekt weitere Steine eingescannt und automatisch gepuzzelt werden, deren Position unbekannt ist. Dafür ist ein intelligenter Suchalgorithmus notwendig.

Ausgehend von den in den vorherigen Abschnitten diskutierten Anforderungen entwickelten wir den folgenden Arbeitsablauf, der auf jedes 3D-Modell eines Steins angewendet wird:

- Simplifikation, um schnelle Zugriffe und Berechnungen zu ermöglichen,
- Klassifikation, um genauere Übereinstimmungen zu finden,
- Identifikation, um eine richtige Lösung unter vielen möglichen zu finden.

Der folgende Abschnitt geht genauer auf die ersten beiden Punkte – Simplifikation und Klassifikation – ein, während Abschnitt „Puzzlealgorithmus“ den Arbeitsablauf insgesamt erläutert und den Aspekt der Identifikation näher beleuchtet.

4 Modellsimplifikation durch *minimale Volumina*

Eine Voraussetzung für einen computergestützten Algorithmus zur Rekonstruktion ist die Auseinandersetzung mit der Architektur von Khmer-tempeln: Im Steinbruch wurden die Steine in Blockform aus dem Fels gehauen und anschließend weiter bearbeitet.

15 An dieser Stelle ist zu beachten, dass der von uns entwickelte Algorithmus nur paarweise Kombinationen prüft und dadurch der eigentliche Aufwand von mindestens $n!$ auf die oben angegebene Formel reduziert werden kann. Das heißt es wird nicht ge-

prüft, wie ein Stein mit allen anderen, 2 Steine mit allen anderen, 3 Steine mit allen anderen, usw., passen, sondern nur geprüft, wie jeweils zwei Steine zueinander passen. Das ergibt $n \cdot (n-1) / 2 \cdot 4 = 2(n^2 - n)$ Möglichkeiten.

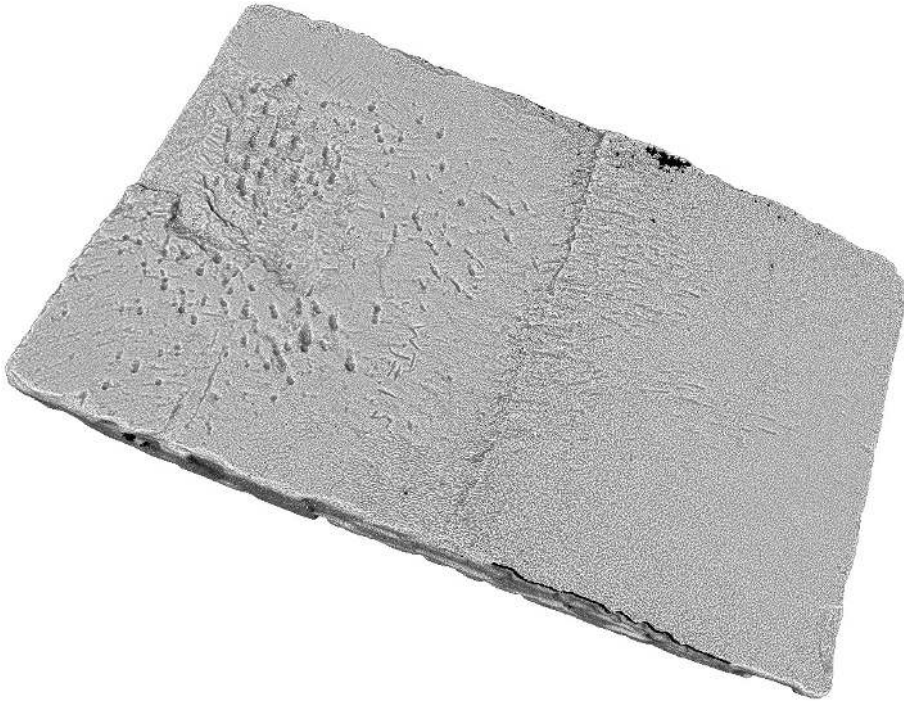


Abb. 9 Die Oberfläche eines Steines. Deutlich zu erkennen sind die Einkerbungen, jedoch existieren keine ‚Hervorhebungen‘.

Über die Weiterverarbeitung gibt es in der Forschung zwei gegenläufige Ansichten. Eine Meinung besagt, dass die Steine unbehauen an ihre Position gebracht und dann durch Hin- und Herschieben so geschliffen wurden, dass sie ihre endgültige präzise Passform erreichten. Die Gegenrichtung vermutet, dass die Steine in Form gehauen wurden.¹⁶

Da beim Tempelbau die Größe der Steine in der Regel unverändert blieb, haben die Erbauer versucht, jede neue Reihe von Steinen auf einer Höhe zu halten. Steine, die an einer Stelle nicht ganz passten, erhielten Ausklinkungen.¹⁷ Der Puzzlealgorithmus verwendet diese Einkerbungen und die hervorragende Passform als eindeutig identifizierende Eigenschaften auf der Suche nach passenden Steinpartnern. Dies führte zu der Idee, sog. *minimal volume boxes* (MVB)¹⁸ zu konstruieren, das heißt Polyeder, welche die

16 Dumarca y 1973; Dumarca y und Royère 2001, 16.

17 Eine Ausklinkung ist eine annähernd rechtwinklige Ausparung, wie sie bei einigen Steinen/Steinmodellen in Abb. 6 (bspw. Mitte rechts) zu beobachten ist. Ein anderes Wort ist auch Einkerbung.

18 Die Wortschöpfung einer *minimal volume box* (MVB) wurde in Anlehnung an *minimal bounding boxes* (MBB) geschaffen. Eine MBB ist ein nicht achsenparalleler Quader minimalen Volumens, der ein beliebiges Objekt vollständig umschließt. Eine MVB hingegen ist nur quaderähnlich und kann auch vom Objekt durchdrungen werden.

Kanten des Steins mittels eines *linear least squares* Verfahrens approximieren. Das Ziel bei den durchaus aufwändigen Berechnungen zur Ermittlung stark vereinfachter Steinmodelle auf Grundlage der hoch aufgelösten Scandaten ist natürlich, spätere Berechnungen schneller durchführen zu können.

Nach dem Scanvorgang liegt das 3D-Modell eines jeden Steins als Punktwolke vor. Um nun solch eine MVB zu berechnen, ist es nötig, die Kanten von den Flächen unterscheiden zu können. Mathematisch gesprochen sind Kanten Bereiche, die eine starke Oberflächenkrümmung aufweisen. Das bedeutet, es muss zuerst ein Krümmungsmodell erstellt werden. Unter den verschiedenen existierenden Möglichkeiten entschieden wir uns für einen *algebraic point set surface* (APSS)¹⁹ Ansatz, da dieser sich schnell und unkompliziert auf unsere Daten anwenden ließ. Mit Hilfe des so erhaltenen Krümmungsmodells werden alle Bereiche, die eine hohe Krümmung aufweisen (in Abb. 10B sind dies die hellgrünen Bereiche), extrahiert. Der Puzzlealgorithmus benötigt Vorwissen über das Vorhandensein und die Position (links oder rechts) einer Ausklinkung in dem Kantenmodell. Mit diesem Vorwissen werden die MVB klassifiziert, um im späteren Puzzlealgorithmus feststellen zu können, welche Kombinationen zweier MVB möglich und sinnvoll sind.

Verschiedene Ansätze, diese Einkerbung automatisch zu detektieren, waren:

- Punktglättung²⁰, um Ausreißer zu eliminieren. Dies führt jedoch dazu, dass die gesuchte Einkerbung sehr verwaschen wird und letztlich nicht mehr vom Rest der Daten zu unterscheiden ist.
- Minimaler Spannbaum²¹: Der so gefundene, kürzeste Weg hat nur wenige Abzweigungen und die Überlegung war, dass solche Abzweigungen idealerweise nur an den Ecken des Modells entstehen. Jedoch traten die Abzweigungen vornehmlich an Stellen auf, die von den Ecken weit entfernt lagen.

Die eben beschriebenen Schwierigkeiten führten auf den von uns entwickelten Zylindertest: Für den Zylindertest wird zuerst die *bounding box* (dreidimensionale Begrenzung

19 APSS sind Oberflächenbeschreibungen basierend auf Punktwolken in der Computergraphik und Visualisierung. Hierbei werden für kleine Nachbarschaften jedes Punktes Kugeln eingepasst. Eine Aussage über die Krümmung der Oberfläche kann nun in Abhängigkeit vom Kugelradius getroffen werden. Siehe auch: Guennebaud und Gross 2007.

20 Das Verfahren der Punktglättung fasst mehrere, in einer Nachbarschaft liegende, Punkte zusammen und untersucht mittels Kovarianzanalyse die Hauptrichtung. Diese wird als Richtungsvektor einer Ge-

raden aufgefasst, auf welche die einzelnen Punkte projiziert werden.

21 Der Minimale Spannbaum ist ein Konzept aus der Graphentheorie. Gegeben sei eine Punktmenge, wobei die Punkte untereinander durch sog. Kanten miteinander verbunden sind, die ihrerseits Gewichte (z. B. Länge, Priorität, ...) haben. Der Minimale Spannbaum verbindet nun alle Punkte durch ausgewählte Kanten so, dass die Summe der Gewichte dieser Kanten möglichst niedrig ist und von jedem Punkt aus jeder andere zu erreichen ist.

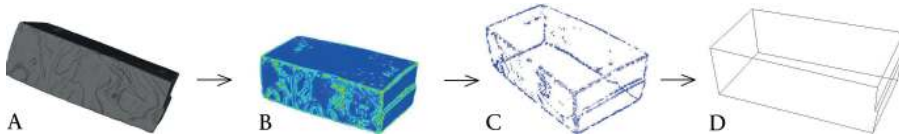


Abb. 10 Vom Stein zur *minimal volume box*. Zuerst die Identifikation: A: 3D-Modell eines Steins, dessen Krümmungen berechnet werden, B: Ermittlung der Punkte mit besonders großer Krümmung, C: Zylindertest, D: fertiges Kantenmodell.

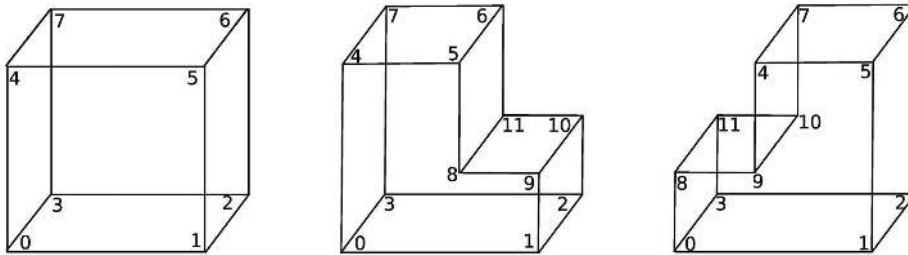


Abb. 11 Vom Stein zur *minimal volume box*. Nach der Identifikation die Klassifikation: „In welche der drei dargestellten Kategorien, gehört der Stein?“

eines digitalen 3D-Objektes) des Modells berechnet. Ausklinkungen können von der *bounding box* nicht erkannt werden und müssen zusätzlich manuell selektiert werden. Danach wird um jeweils zwei benachbarte Eckpunkte ein Zylinder gelegt, dessen Höhe der Abstand der Punkte ist und für dessen Radius sich ein Wert von 15,0 cm in der Praxis als passend erwiesen hat. Für alle Modellpunkte, die sich innerhalb dieses Zylinders befinden, wird danach mittels *linear least squares* Verfahren eine Ausgleichsgerade berechnet. Sobald dies für alle Kanten durchgeführt wurde, wird in den jeweiligen Ecken der Punkt ermittelt, der allen drei Geraden am nächsten ist und dieser als endgültiger Eckpunkt festgelegt. Dieses Verfahren ist nötig, da die so entstehende vereinfachte Fläche, die nur durch Eckpunkte beschrieben wird, die Fläche des Steins möglichst gut approximieren soll. Hierbei besteht die Möglichkeit, dass einer der Eckpunkte nicht mehr auf dem Modell liegt, sondern „im Leeren“ (beispielsweise, falls ein Teil ausgebrochen ist). Da die existierenden 3D-Softwarelösungen solche Fälle nicht unterstützen, müssen diese auf die eben beschriebene Art und Weise berechnet werden.

Abschließend wird die so errechnete *minimal volume box* einer der folgenden Kategorien zugeordnet: keine Ausklinkung, eine Ausklinkung (rechts oder links, wobei die räumliche Orientierung²² des 3D-Modells eines Steins bekannt ist) und zwei Ausklinkungen. Abb. 10 und Abb. 11 veranschaulichen dies. Die Simplifikation auf die MVB wird in Abb. 10 dargestellt, zwei der drei Kategorien in Abb. 11.



Abb. 12 Zwei Steine mit sehr kleinem Schnittwinkel, die anhand ihrer MVB korrekt angeordnet wurden.

5 Puzzlealgorithmus

Aus den beschriebenen Vorgehensweisen wurde ein semiautomatischer Puzzlealgorithmus entwickelt. Dieser berechnet im ersten Schritt für jeden Stein seine MVB-Eckpunkte und die zugehörigen MVB-Flächen, also das Kantenmodell. Für die möglichst korrekte Lösung des Puzzles sind nun verschiedene Kriterien (sog. Puzzlekriterien) nötig. Als erstes Puzzlekriterium werden im zweiten Schritt jeweils zwei Steine an einer der möglichen Kontaktflächen miteinander verbunden und der Schnittwinkel der zugehörigen MVB-Flächen in dieser Position berechnet. Dies wird für alle zulässigen und sinnvollen²³ Kombinationen von Flächen und Steinen durchgeführt und die Ergebnisse nach Größe des Schnittwinkels sortiert. Der letzte Schritt zeigt – beginnend mit dem kleinsten Winkel – dem Benutzer jede der Möglichkeiten am Monitor an. Dieser entscheidet dann, ob die dargestellte Kombination korrekt ist. Abb. 12 zeigt ein zueinander passendes Steinpaar, das auf diese Weise gefunden wurde. Im Idealfall beträgt der Schnittwinkel 0° , doch aufgrund der Simplifikation der Oberflächen wird dies nicht in jedem Fall erreicht.

22 Hierbei ist gemeint, dass „oben“, „unten“, „rechts“, „links“, „vorne“ und „hinten“ bekannt sind. Da es kein uns bekanntes Verfahren gibt, diese räumliche Orientierung automatisch zu bestimmen, wird dies beim Scanvorgang von Hand dem 3D-Modell als Zusatzinformation hinzugefügt.

23 Die Mauer hat eine Dicke von genau einem Stein, das heißt die Vorder- und Rückseiten der Steine haben keine ‚Partner‘. Daraus ergibt sich, dass Kombinationen von Vorderseite auf Vorderseite oder Rückseite auf Rückseite nicht zulässig sind. Eine nicht sinnvolle Kombination wäre, Steine der Kategorien, die in der Abb. 11 *mittig* und *rechts* dargestellt sind,

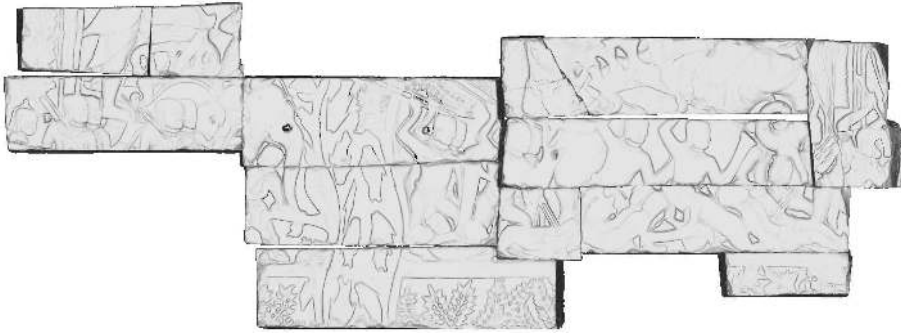


Abb. 13 Ergebnis des ersten semiautomatischen Puzzlealgorithmus.

Der Ebenenschnittwinkel als alleiniges Kriterium für den letzten Schritt der Identifikation ist jedoch vor allem bei größeren Datensätzen unzureichend, da auch falsche Paarungen einen sehr kleinen Schnittwinkel besitzen können. Dies ist auch der Grund für die semiautomatische Implementierung: Um eine Häufung von falschen Kombinationen auszuschließen, wird der Benutzer stark eingebunden. Eine Lösung des Problems der unzureichenden Ebenenschnittwinkel führte zu der Idee eines Ähnlichkeitsmaßes für zwei Kontaktoberflächen, das im Ausblick näher erläutert wird.

6 Ergebnisse

Der im vorangegangenen Abschnitt vorgestellte Algorithmus konnte bereits erfolgreich getestet werden. Abb. 13 zeigt 13 Steine, die auf diese Weise zusammengesetzt wurden. Der zeitliche Aufwand dafür betrug ca. zehn Minuten, wobei knapp die Hälfte der Zeit auf die Berechnungen der Oberflächenkrümmung und der Ebenenschnittwinkel entfallen ist und der Rest benötigt wird, um von einem Nutzer entscheiden zu lassen, ob die jeweils angezeigten Möglichkeiten tatsächlich korrekt sind. Eine einfache Extrapolation dieser Zeit auf die in Abschnitt „Rekonstruktion“ vorgestellten manuell zusammengeführten Mauerteile mit 65 einzelnen Steinen ergibt eine Zeit von knapp zwei Stunden. Somit ist eine deutliche Zeitersparnis zu erwarten im Vergleich zu der manuellen Puzzlearbeit vor Ort, selbst wenn sich die Rechenzeit aufgrund der Verwendung des Ähnlichkeitsmaßes und der größeren Steinanzahl erhöht.

an ihren Seiten miteinander in Kontakt zu bringen,

da der entstehende Zwischenraum zu klein für einen weiteren Stein wäre.

7 Ausblick

In Zukunft soll der bisherige Algorithmus durch ein Ähnlichkeitsmaß unterstützt werden. Dabei wird untersucht, wie sehr die Oberflächen zweier Steine zueinanderpassen. Dies ist wichtig, um die Benutzerinteraktion möglichst gering zu halten und trotzdem bei einer großen Anzahl an möglichen Übereinstimmungen diejenigen herausfiltern zu können, die tatsächlich zueinander gehören.

Darüberhinaus wurde bereits eine Datenstruktur basierend auf der Halbkantendatenstruktur²⁴ entwickelt, die auch in den Algorithmus miteinfließen wird. Damit ist es möglich, schon gefundene Steinpaare von solchen abzugrenzen, die noch zu finden sind.

24 Die Halbkantendatenstruktur (auch doppelt verkettete Kantenliste genannt) erlaubt die effiziente Speicherung und das Wiederauffinden der Anpassungen und ist im vorliegenden Fall sinnvoll, da die

einzelnen Modelle durch ihre Kanten beschrieben werden. Jeder Kante eines Steins wird die zugehörige Kante des angepassten Steins zugeordnet.

Bibliographie

Besl und McKay 1996

Paul J. Besl und Neil D. McKay. „A Method for Registration of 3d-Shapes“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1996), 239–258.

Chandler 2008

David Chandler. *A History of Cambodia*. Boulder/Colorado: Westview Press, 2008.

Diestel 2010

Reinhard Diestel. *Graphentheorie*. Heidelberg: Springer, 2010.

Dumarcay 1973

Jacques Dumarcay. „Histoire architecturale du temple“. In *Bayon*. Bd. III of Mémoires archéologiques. Paris: Publications de l'école française d'extrême-orient, 1973.

Dumarcay und Royère 2001

Jacques Dumarcay und Pascal Royère. „Cambodian Architecture. Eighth to Thirteenth Centuries“. In *Handbook of Oriental Studies*. Hrsg. von B. Arps, M. C. Ricklefs und D. K. Wyatt. Bd. 12(3): South-East Asia. Leiden: Brill, 2001.

Freudenreich u. a. 2011

Julia Freudenreich, Anja Schäfer, Pheakdey Nguonphan, Michael Winckler und Hans Georg Bock. „Close Range Architecture Documentation Of Angkor Style Temples“. In *Proceedings of DMACH 2011 (Conference, Amman, Jordan, March 13–15th, 2011)*. 2011, 237–247.

Groslier 1937

George Groslier. „Une merveilleuse cité khmère. Banteai Chhma, ville ancienne du Cambodge“. *L'Illustration* 3 (1937). 4909, 352–357.

Guennebaud und Gross 2007

Gaelu Guennebaud und Markus Gross. „Algebraic Point Set Surfaces“. *ACM Transactions on Graphics (SIGGRAPH 2007 Proceedings)* 26.3 (2007). Art. 23. doi: 10.1145/1276377.1276406.

Higham 2004

Charles Higham. *Encyclopedia of Ancient Asian Civilizations*. New York: Facts on File, 2004.

Huang u. a. 2006

Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer und Helmut Pottmann. „Reassembling Fractured Objects by Geometric Matching“. *Proceedings of ACM Siggraph* 25 (2006), 569–578.

Pottman u. a. 2009

Helmut Pottman, Johannes Wallner, Qi-Xingand Huang und Yong-Liang Yang. „Integral Invariants for Robust Geometry Processing“. *Computer Aided Geometric Design* 26.1 (2009), 37–60.

Reuter u. a. 2007

P. Reuter, G. Rivière, N. Sorraing, L. Espinasse und R. Vergnieux. „Archeo TUI – A Tangible User Interface for the Virtual Reassembly of Fractured Archeological Objects“. In *VAST*. 2007.

Schäfer u. a. 2011

Anja Schäfer, Hubert Mara, Julia Freudenreich, Bernd Breuckmann, Christiane Düffort und Hans Georg Bock. „Large Scale Angkor Style Reliefs: High Definition 3D Acquisition and Improved Visualization Using Local Feature Estimation“. In *Revive the Past. CAA 2011 – Proceedings of the 39th Conference*. Hrsg. von Mingquan Zho, Iza Romanowska, Zhongke Wu, Pengfei Xu und Philip Verhagen. Amsterdam: Pallas Publications, 2011, 318–328.

Winkelbach und Wahl 2008

Simon Winkelbach und Friedrich Wahl. „Pairwise Matching of 3d Fragments Using Cluster Trees“. *International Journal of Computer Vision* 78 (2008), 1–13.

Abbildungsnachweis

1 Pheakdey Nguonphan. 2 Pheakdey Nguonphan. 3 Pheakdey Nguonphan. 4 Pheakdey Nguonphan. 5 Pheakdey Nguonphan. 6 Anja Schäfer, Heike Leitte und Hans Georg Bock. 7 Anja Schäfer, Heike Leitte und Hans Georg Bock. 8 Winkelbach und Wahl 2008, 5. 9 Anja

Schäfer, Heike Leitte und Hans Georg Bock. 10 Anja Schäfer, Heike Leitte und Hans Georg Bock. 11 Anja Schäfer, Heike Leitte und Hans Georg Bock. 12 Anja Schäfer, Heike Leitte und Hans Georg Bock. 13 Anja Schäfer, Heike Leitte und Hans Georg Bock.

ANJA SCHÄFER

Dipl.-Math. (Heidelberg 2010), war wissenschaftliche Mitarbeiterin beim Interdisciplinary Center for Scientific Computing (IWR) an der Universität Heidelberg, wo sie 2015 promoviert wurde. Ihr wissenschaftliches Interesse liegt in sequentia quadratic programming (SQP)-Methoden, nonlinear programs (NLP), der Automation und Optimierung von 3D-Puzzels und der Erstellung von 3D-Scans archäologischer Artefakte.

Anja Schäfer
Interdisciplinary Center for Scientific Computing (IWR)
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg, Deutschland
E-mail: Anja.Schaefer@iwr.uni-heidelberg.de

HEIKE LEITTE

M.Sc. (Leipzig 2006), Promotion (Leipzig 2009), ist Juniorprofessorin für „Computergraphik und Visualisierung (CoVis)“ am IWR (Interdisciplinary Center for Scientific Computing) der Universität Heidelberg. Ihre Forschungsschwerpunkte sind Analysemethoden zeitabhängiger, multivariabler Daten, Informationstheorie zur Visualisation, Fließsimulationen, Klimatologie und Biologie.

Prof. Dr. Heike Leitte
Interdisciplinary Center for Scientific Computing (IWR)
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg, Deutschland
E-mail: heike.leitte@iwr.uni-heidelberg.de

HANS GEORG BOCK

Dipl.-Ing. (Köln 1975), Promotion (Bonn 1986), ist Professor am Lehrstuhl für Softwareinformatik und Optimierung und geschäftsführender Direktor des IWR (Interdisciplinary Center for Scientific Computing) der Universität Heidelberg. Er beschäftigt sich neben mathematischen und statistischen Schwerpunktthemen u.a. mit computergestützten Methoden in der Denkmalpflege und ist neben vielen weiteren Aufgaben Initiator und Leiter des Komitees für Mathematische Modellierung, Simulation und Optimierung (KoMSO) am Vietnamesischen Ministerium für Ausbildung und Erziehung.

Prof. Dr. Dr. h. c. mult. Hans Georg Bock
Interdisciplinary Center for Scientific Computing (IWR)
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg, Deutschland
E-mail: SciCom@iwr.uni-heidelberg.de