

Migration des 3D Stadtmodells Hamburg mit CityGRID® und FME



UVM Systems

Geschäftsbereiche:

- Dienstleistung, Consulting und Softwareentwicklung mit Schwerpunkt 3D
- Entwicklung der City**GRID**[®] Software
- Echtzeitvisualisierung
- 3d Stadtmodellierung

Team:

- 8 Mitarbeiter aus den Bereichen Softwareentwicklung, Geoinformatik und Konstruktion
- Langjährige Erfahrung in Softwareentwicklung
- Langjährige Erfahrung in Echtzeitvisualisierung und Stadtmodellierung
- Firmensitz in Klosterneuburg bei Wien

CityGRID

- Software für Erstellung, Administration und Fortführung von 3D Stadtmodellen
- Datenbankgestützte Verwaltung
- Linienbasierte Datenhaltung
- Auf Gebäudemodelle optimierter Flächenbildungsalgorithmus
- Aufbau einer Gebäudesemantik durch hierarchische Strukturierung
- Schnittstelle zu FME mittels City**GRID**® Reader/Writers

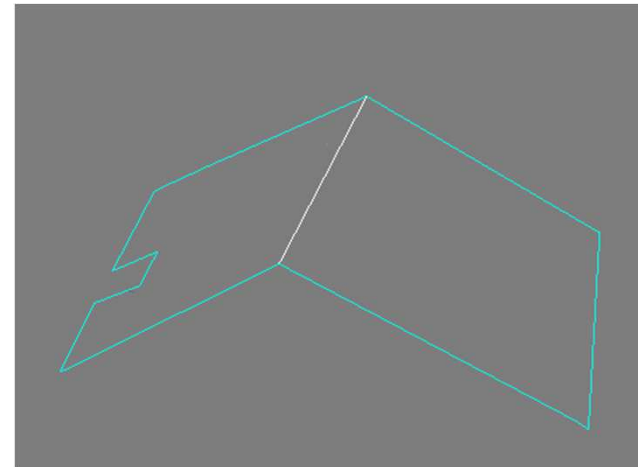


Begriffsdefinition


- Was verstehen wir unter 3D
Geometriedaten mit drei Koordinaten, die die senkrechte oder überhängende Abschnitte enthalten.



- Was ist ein 3D Polygon?
Ein in sich zurücklaufender, geschlossener Linienzug der eine oder mehrere Ebenen beschreibt.



Aufgabenstellung

- Auftraggeber: LGV Hamburg 
- Umgestaltung des Hamburger Stadtmodells (ca. 130 000 Gebäude) in eine objektstrukturierte Datenhaltung.
- Vorbereitung für Migration in eine Datenbank.
- Neubildung von Gebäudeobjekten auf Basis von Grundrissen aus amtlichem Liegenschaftskataster.
- Allfällige Korrektur von Geometrie der vorhandenen Einzel-Dachflächen
- Integration von zusätzlichen Dachdetails (Ziergiebel) sowie Durchfahrten.
- Optionale Bildung eines Dachüberstandes auf Basis der zugewiesenen Grundrisse.

Projekttablauf

- Projektstart: Herbst 2008
- Konzeption und Entwicklung einer Prozesskette durch UVM Systems
- 5 Arbeitsschritte mit insgesamt 8 Workbenches zur Anwendung beim Kunden erstellt.
- Funktionelle Erweiterungen in FME
- Bei LGV in Betrieb seit Sommer 2009
- Laufende Erweiterungen und Adaptierungen

Eingesetzte Software



FME

- Arbeitsumgebung automatisierte Prozesse
- Entwicklung ohne Programmierer
- Batchmodus



CityGRID®

- Flächenbildung (Triangulierung)
- Datenhaltung (XML und Datenbank)
- Gebäudesemantik
- Interaktive Modellierung von Problemsituationen

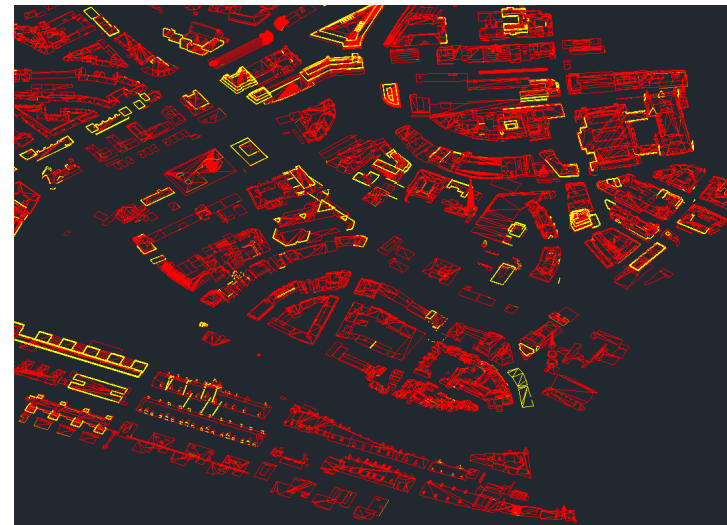


CityGRID®



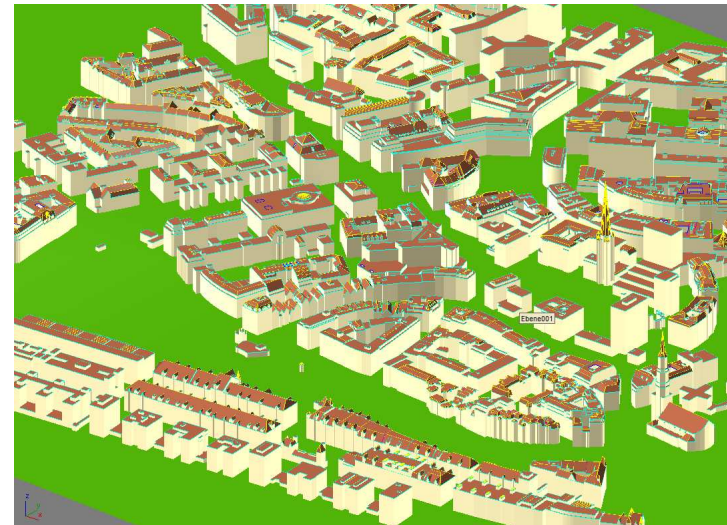
Schritt 1: Dachgruppenbildung

- Übernahme bestehender Dachdaten (Flächenmodell) und Kantenextraktion
- Vereinigung von Dachflächen über gemeinsame Kanten (Schwellwert).
- Beseitigung von überstehenden, sich durchdringenden Flächen im Dachbereich.
- Lückenschluss und Überlappungskorrektur durch 3D Snapping.
- Detektion von Dachdetails und Strukturierung ins LoD Konzept.
- CityGRID® Linienmodell und Flächenbildung mittels CityGRID® FME Writer



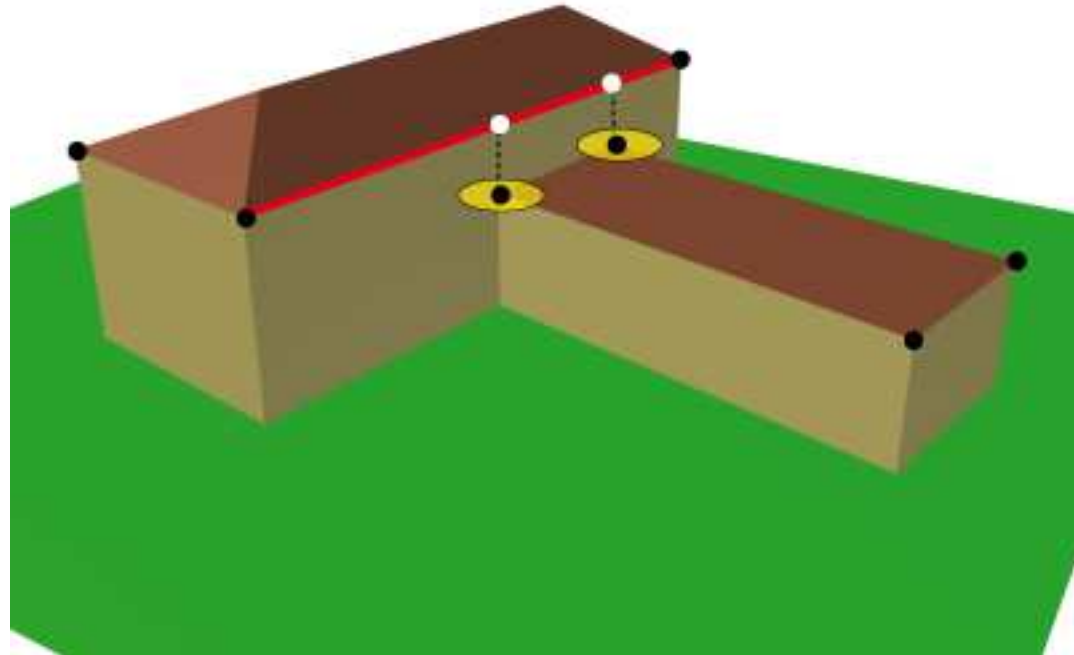
Schritt 1: Dachgruppenbildung

- Übernahme bestehender Dachdaten (Flächenmodell) und Kantenextraktion
- Vereinigung von Dachflächen über gemeinsame Kanten (Schwellwert).
- Beseitigung von überstehenden, sich durchdringenden Flächen im Dachbereich.
- Lückenschluss und Überlappungskorrektur durch 3D Snapping.
- Detektion von Dachdetails und Strukturierung ins LoD Konzept.
- CityGRID[®] Linienmodell und Flächenbildung mittels CityGRID[®] FME Writer



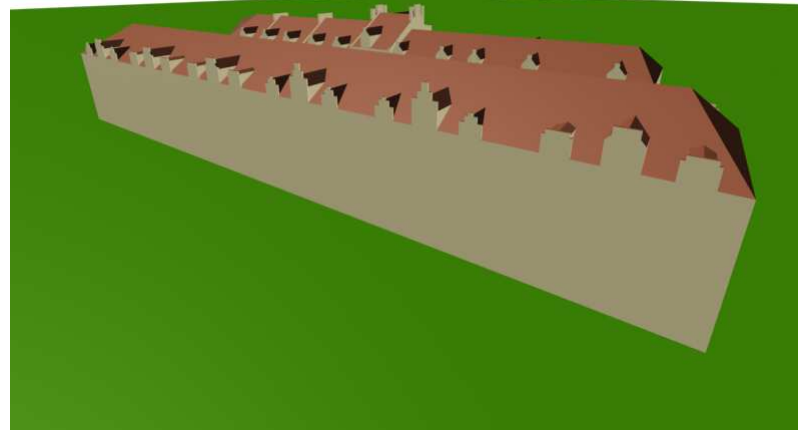
Transformer CGPolygonVertexSynchronizer und Snapper3D

- Fügt Punkt in Polygon ein wenn ein Punkt in der Suchtoleranz gefunden wird.
- 2.5D Untersuchungen an beteiligter Geometrie
- Schließen von Klaffungen/Überschneidungen durch XY Verschiebung
- Erhaltung des Z Wertes.
- Berücksichtigung von elliptischen Strukturen (Kuppeln)



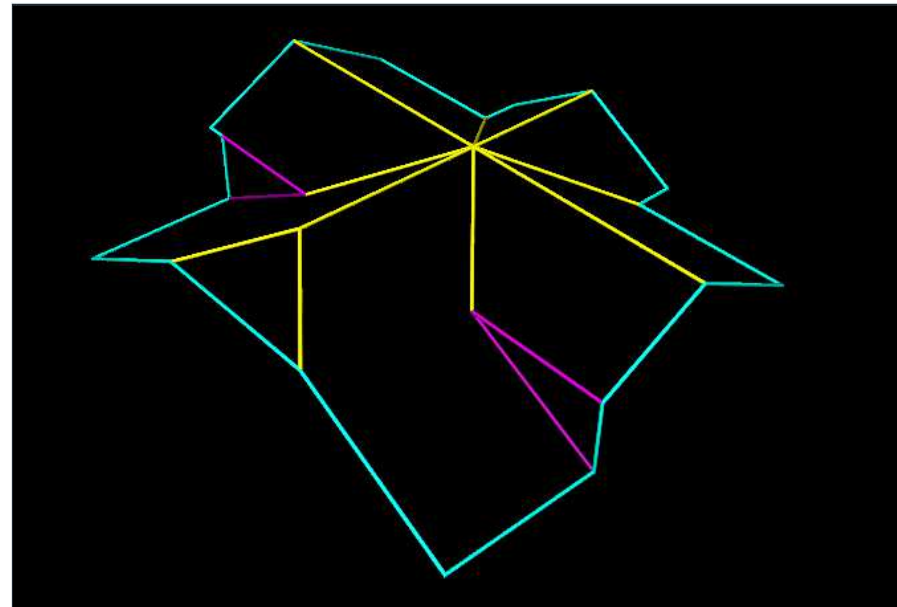
Schritt 2: Integration Dachdetails

- Kombination von Gebäudemodellen (Schritt1) mit externen Dachdetail-Daten
- Erkennung von vorhandenen Bodenflächen (nicht einheitliche Flächennormale) und Entfernung bei vollständiger Lage im Hauptdach
- Vereinigung von aneinander gebauten Dachdetails (getreppte Ziergiebel)
- Semantisch korrekte Eingliederung in Gebäudemodell



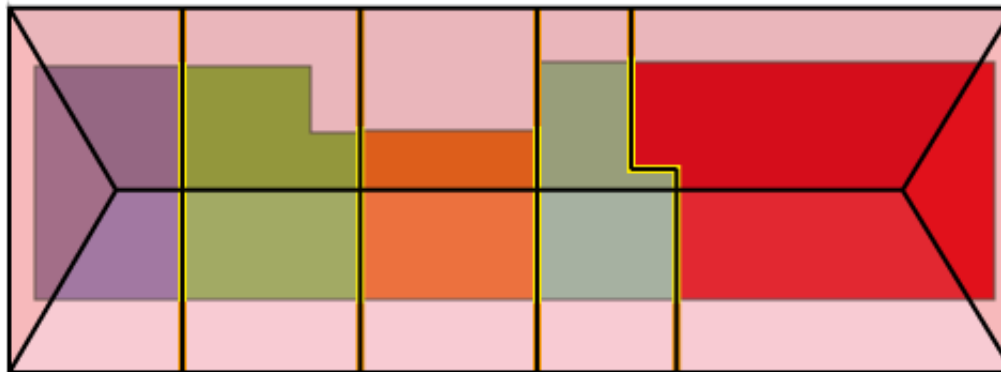
Transformer CGPolygonJoiner

- 3D Dissolver
- Kombination von Flächen mit gemeinsamen Kanten
- Koplanare Flächen vereinen
- Minimale Kantenlänge zum Vereinen von Flächen
- Senkrechte Abschnitte im vereinigten Polygon eingefügt



Schritt 3: Neubildung Gebäudeobjekte

- Kombination Gebäudemodelle mit Grundrissen
- Transformation Gebäudeblöcke über zugehörige Grundrissblöcke
- Zuweisung Grundrisse zu Dachflächen über Schwellwert
- Teilung von Reihenhaussituationen (1 Dach – mehrere Grundrisse)
- Attributübertragung (Gebäudecode und UUID Grundriss)

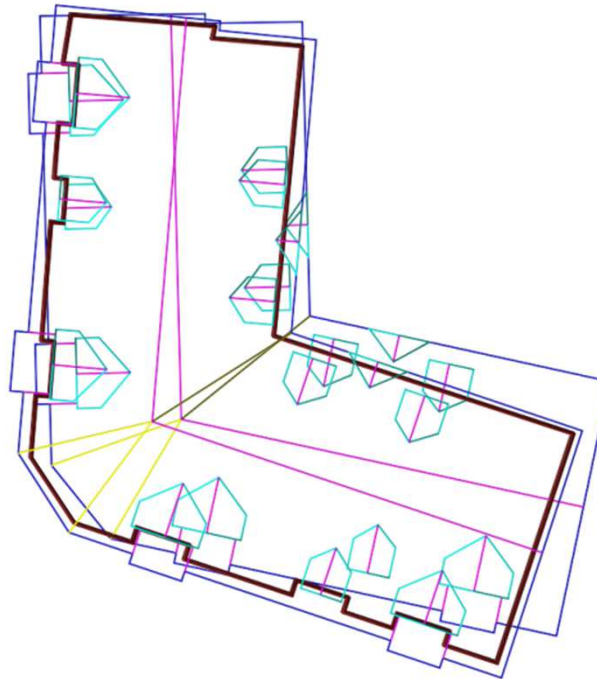


Transformer CGOrientationAnalyzer & CGOffsetAnalyzer

Ermittlung von Verdrehungs- und Verschiebungswerten bei Polygonen gegen Referenzpolygone

Anwendung bei signifikanter Überschreitung von Schwellwerten

Maximaler Rotations- und Offsetwert



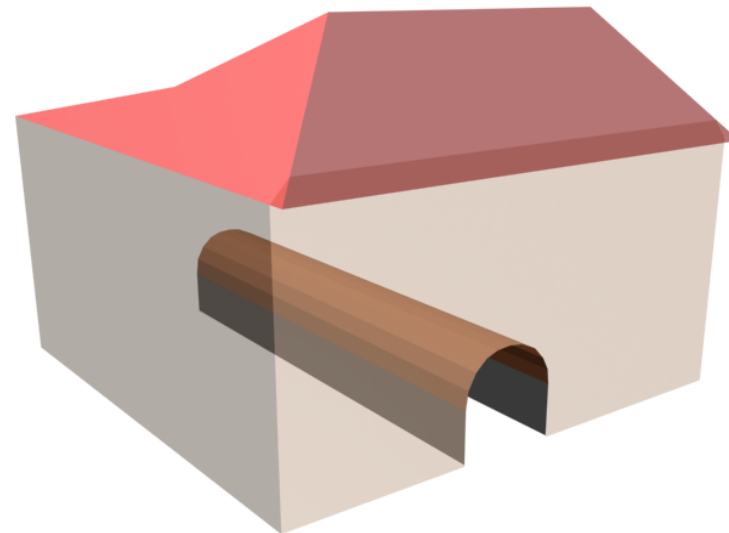
Schritt 4: Integration Durchfahrten

FME

- Durchfahrten, Arkadengänge, Galerien als Abzugskörper
- Kombination Gebäudelinien mit Liniengerüst von Abzugskörpern in FME

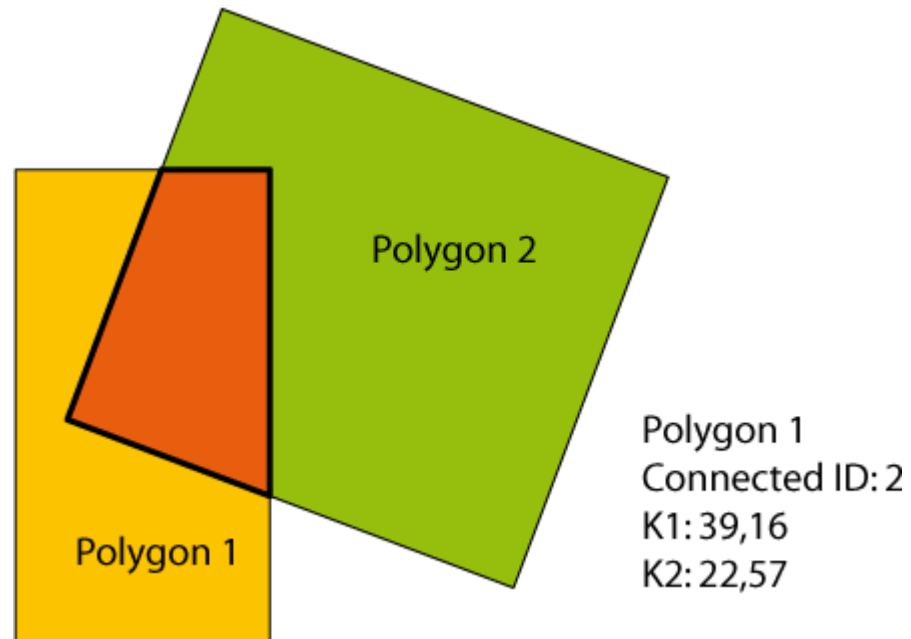
CityGRID®

- Entfernung von Fassadenteilen und Flächenbildung im Durchgangsbereich
- Semantische Eingliederung in Gebäudestruktur



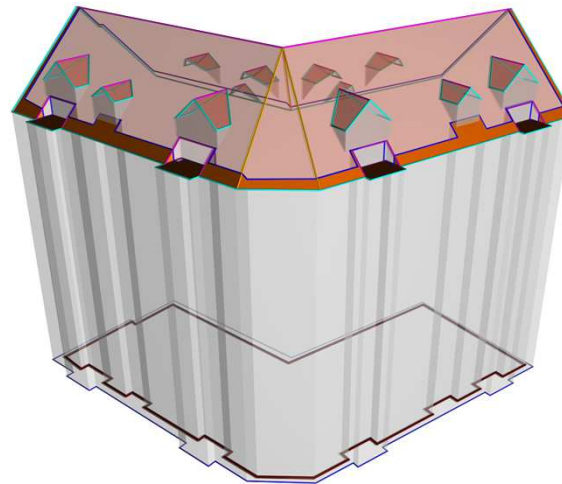
Transformer CGOverlapAnalyzer

- Berechnung von Überlappungen bei 2D Polygonflächen
- Gegenseitige Überdeckungswerte in Prozent
- Listenattribut mit allen Überdeckungswerten
- Polygonflächen mit maximaler Überdeckung finden



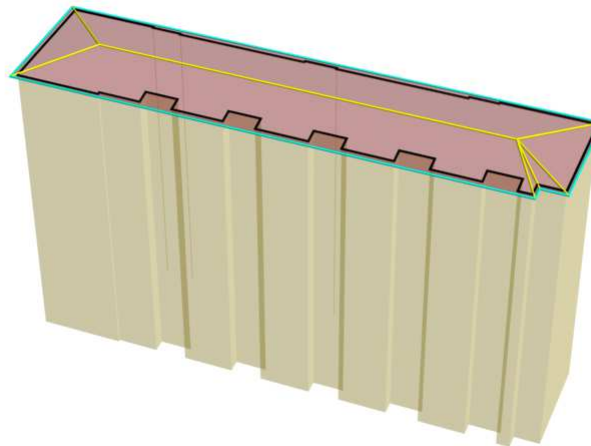
Schritt 5: Generierung Dachüberstand

- Gebäudemodelle mit Grundrissen aus amtlichem Liegenschaftskataster
- Projektion von 2D Grundriss in 3D Dachfläche
- Hochgerechneter Grundriss als Fassadenoberkante
- Fläche zwischen Fassadenoberkante und Traufe (wasserdicht)



Transformer CGPolygon2Mesh

- Projektion von Polygonen in eine 3D Fläche
- Verwendung von bestehender Fläche
- Berücksichtigung von vertikalen Sprüngen
- Offset an projizierte Polygone anbringen



Fazit

- Erfolgreiche Projektentwicklung und Durchführung durch Kombination von FME und City**GRID**®
- Workbencherstellung ohne Programmierkenntnisse
- Funktionalitätserweiterung von FME durch unser eigenes Entwicklerteam
- Universelle Einsetzbarkeit durch Verbreitung von FME
- Gute Integration und Kapselung von selbst entwickelten Funktionen in FME
- Flexible Anwendungsmöglichkeiten einzelner Bausteine

DANKE!