

Zwischenpräsentation
Forschungsprojekt

Swarm Intelligence

Juni 2004

Andrea Lahn, Christian Nitschke,
Onno Haak, Nicolai Marquardt

Projektleitung: Hendrik Wendler, Prof. Fröhlich
Bauhaus-Universität Weimar

Inhalt

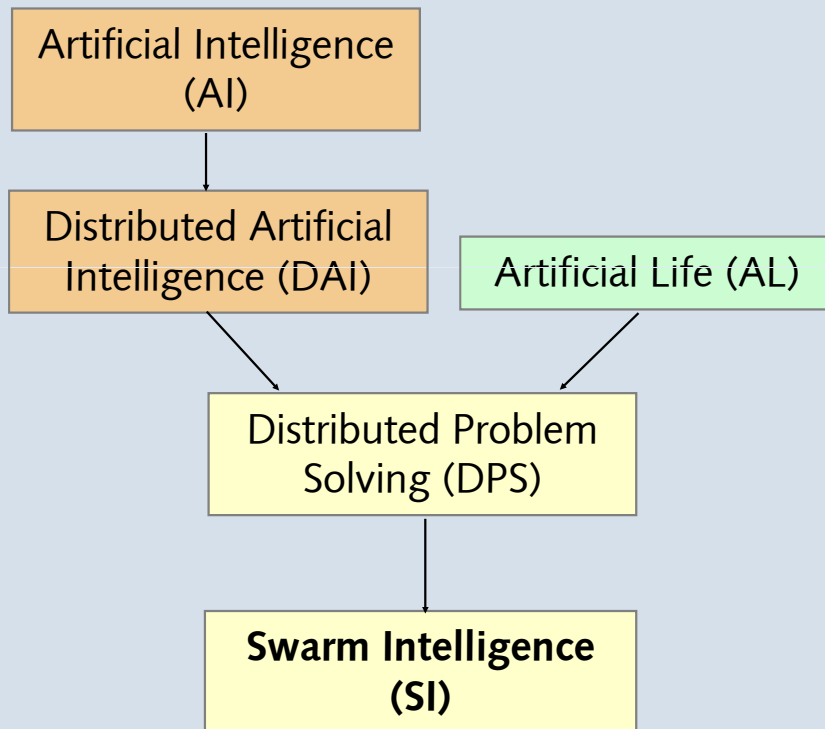
1. Grundlagen: Swarm Intelligence

2. Implementierung, Algorithmen, Prototypen

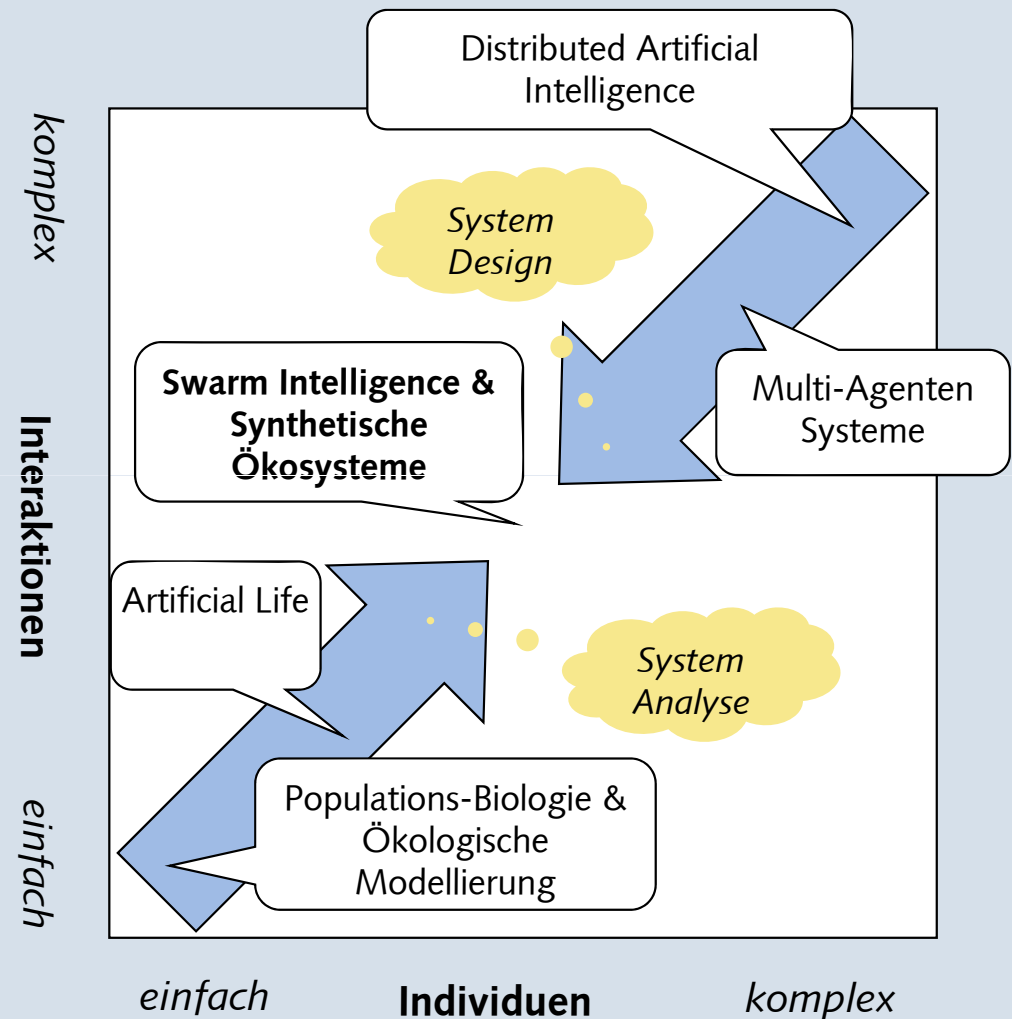
3. Ausblick, weitere Forschung

1. Grundlagen: Swarm Intelligence

1. Grundlagen: Einordnung Swarm Intelligence



Einordnung in Artificial Systems

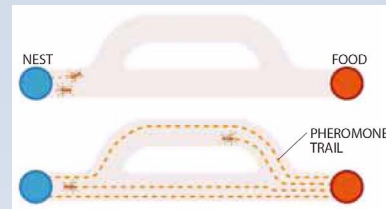


Einordnung zwischen System-Design und -Analyse

1. Grundlagen: Anwendungen Swarm Intelligence

- **Ant Colony Optimization (ACO)**

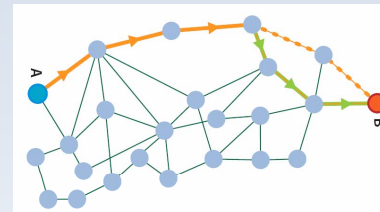
- Math. Optimierungsproblem
- TSP, QAP, JSP, u.a.



© E. Bonabeau

- **Ant-based Control (ABC)**

- Verteilung von Produktionslast
- Ant-based Routing



© E. Bonabeau



© P. Flannery

- **Robotik**

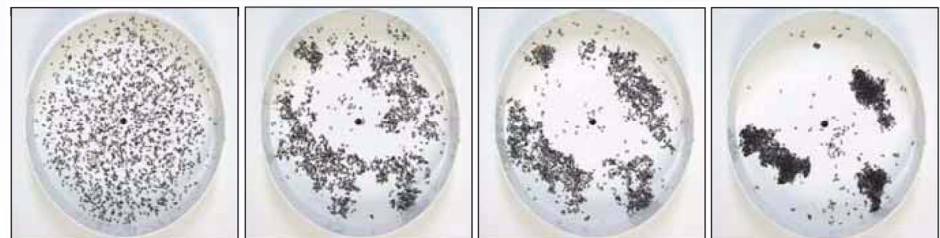
- Dynamische Arbeitsteilung
- Skalierung, Team-Work



- **Dezentralisierung**

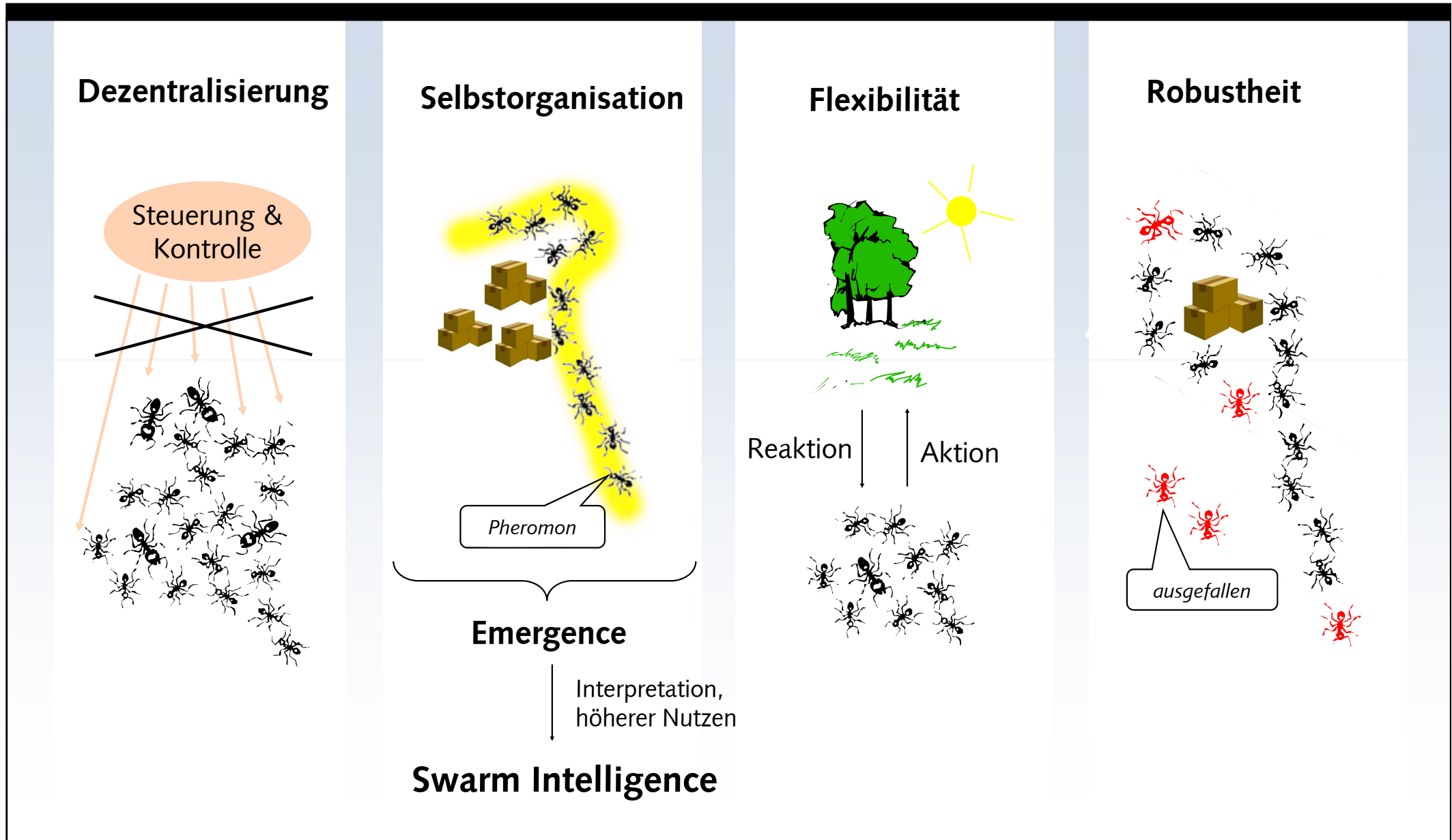
- Strukturen in Firmen
- Ubiquitous Computing

- **Ant-based Clustering / Sorting**



ERIC BONABEAU AND GUY THERAULAZ

1. Grundlagen: Prinzipien Swarm Intelligence

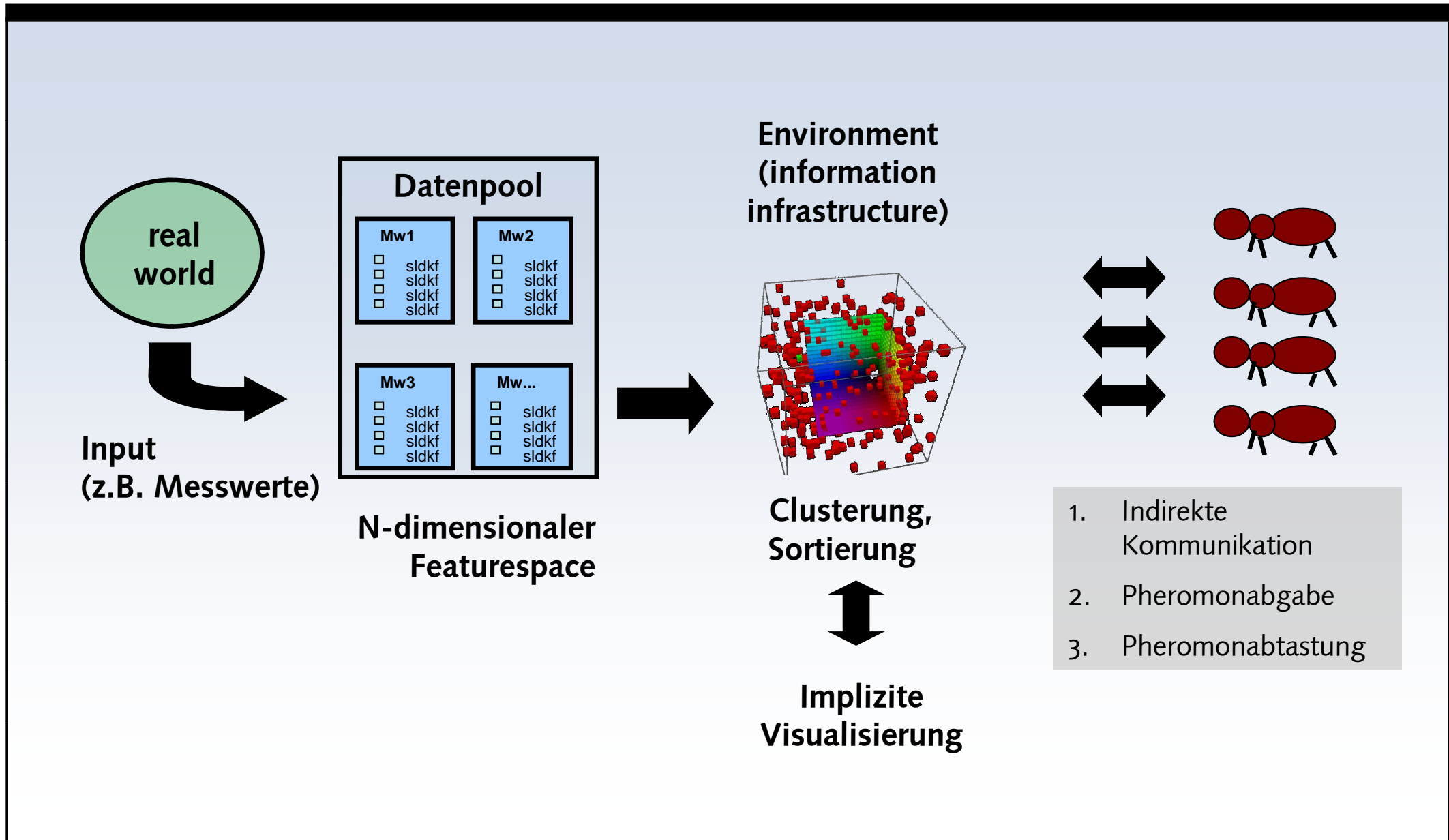


1. Grundlagen: Swarm Intelligence

Prinzipien

- Ameisen (Agenten) sind simple autonome Entitäten
 - nur lokale Sicht
 - keine oder nur geringe Spezialisierung
 - ausschließlich indirekte Kommunikation
(Vermeidung ‚communication overhead‘ und ‚deadlocks‘)
- kein Erfolgsmaß/Eigennutzen, ausschliesslich Verhaltensregeln
- Pheromonlandkarte
 - Autokatalyse → Kettenreaktion

1. Grundlagen: Swarm Intelligence



2. Implementierung, Algorithmen, Prototypen

2. Implementierung, Algorithmen, Prototypen

Frameworks, Implementierung

- Anforderungen an System:
 - Darstellungen in 2D und 3D
 - Algorithmus aufbauend auf Acluster (Ramos) und Erweiterungen
 - Einlesen beliebiger Daten und Abbilden in Parameter-Raum
 - Variable Einstellung alle Parameter (Grenzwerte, u.a.)
 - Beliebige Ergänzungen der Module (Agenten, Visualisierung)
- C++, OpenGL, GLUT, GLUI
- OO-Aufbau: Klassen-Struktur:
Ant, Swarm, Object, World, Scheduler, Display (OpenGL)

2. Implementierung, Algorithmen, Prototypen

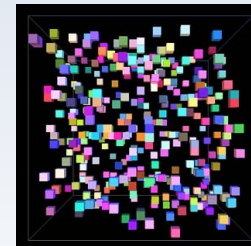
Simulation: Initialisierung



- Parameter: Dimensionen (2D, 3D), Konstanten
- Rand-Bedingungen (umlaufend, spiegelnd, begrenzt)

- Daten importieren
- Attribute in Parameter-Raum abbilden (n-Dim)
- Objekt-Eigenschaften ggf. auf Farben abbilden
- Objekten zufällige Positionen zuweisen

- Anzahl Agenten bestimmen (abh. von Anzahl Objekten, Agenten, Dimension)
- Agenten-Verhalten definieren (aufheben, ablegen, Pheromone)
- Zufällige Positionen und Richtungsvekt.



2. Implementierung, Algorithmen, Prototypen

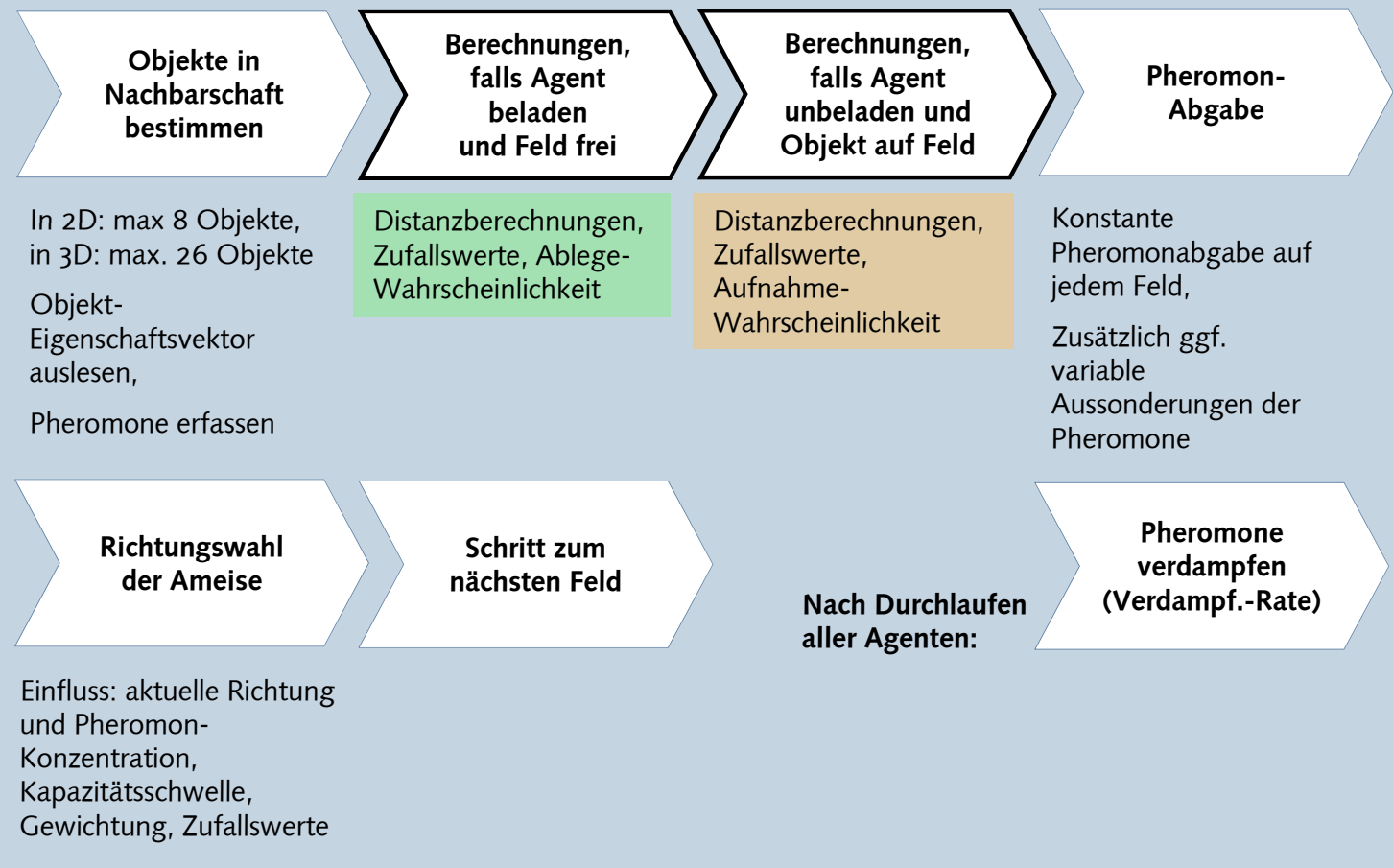
Simulation:

Iteration



Nach Acluster-Algorithmus von V. Ramos [2] [3]

Iterations-Schritte 1 bis MAX_ITERATION, für jeden Agenten durchlaufen



2. Implementierung, Algorithmen, Prototypen

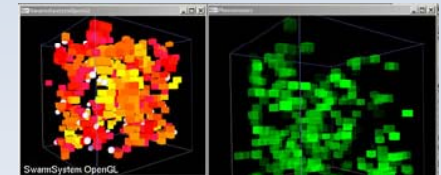
Simulation:

Darstellung, Auswertung



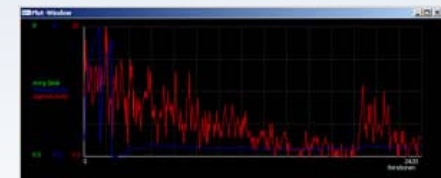
Darstellung, Visualisierung:

- OpenGL
- Objekt-Raum und Pheromon-Feld
- Rotation, Translation, Zoom, Ebenen, Transparenz
- Zuordnen der Farben zu Objekt-Eigenschaften



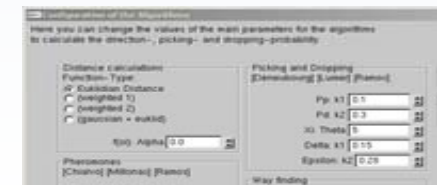
Graphen zur Messung:

- Aktivität der Agenten (Beladen, Ablegen)
- Pheromonfeld (Mittelwert, Max-Werte)
- Entropie (mittlere Übereinstimmung, Dichte des Objekt-Feldes, Cluster-Distanzen, Cluster-Anzahl)

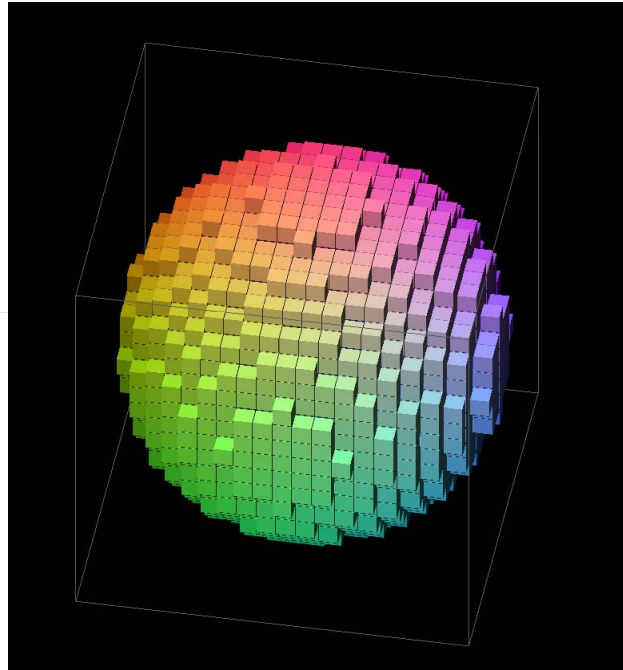


Parameter-Konfiguration:

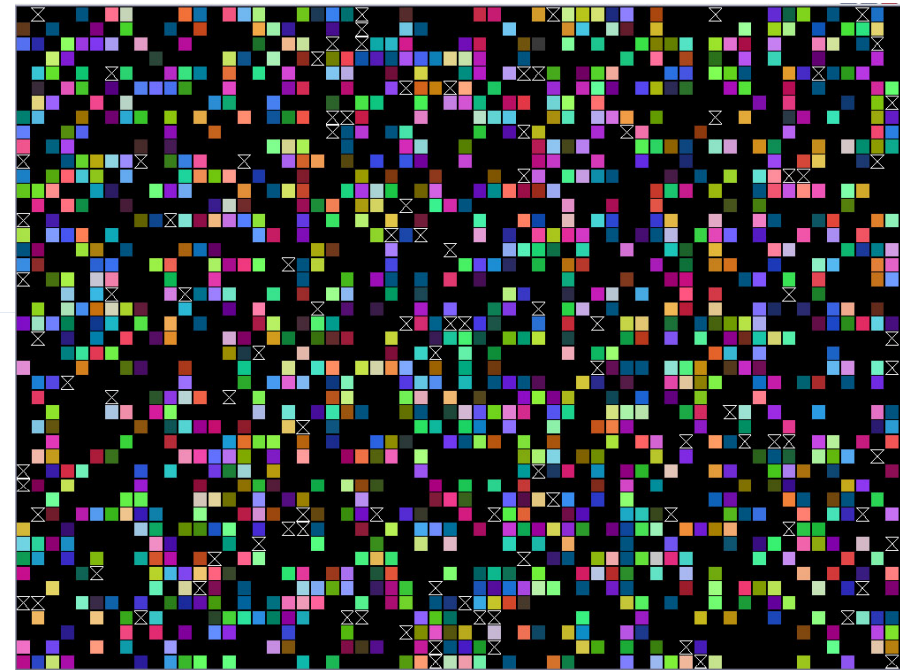
- Während Laufzeit modifizieren
- Evaluieren der veränderten Parameter



2. Prototypen

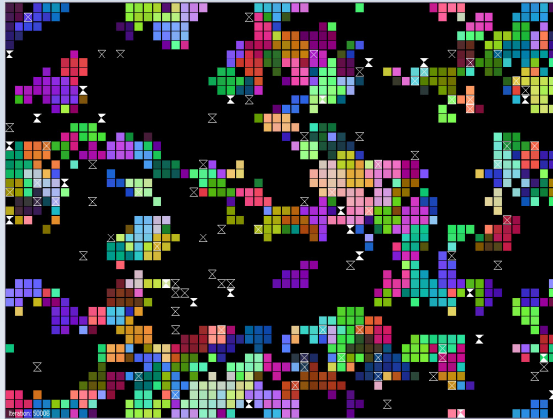


Kugel im 3D-Raum,
mit den Koordinaten der Voxel
als Eigenschaften der Vektoren

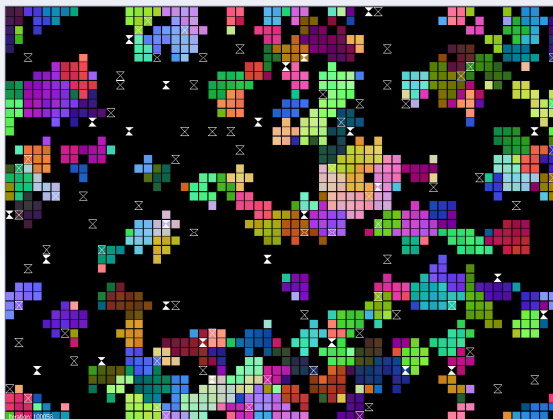


Zufällig positionierte Voxel

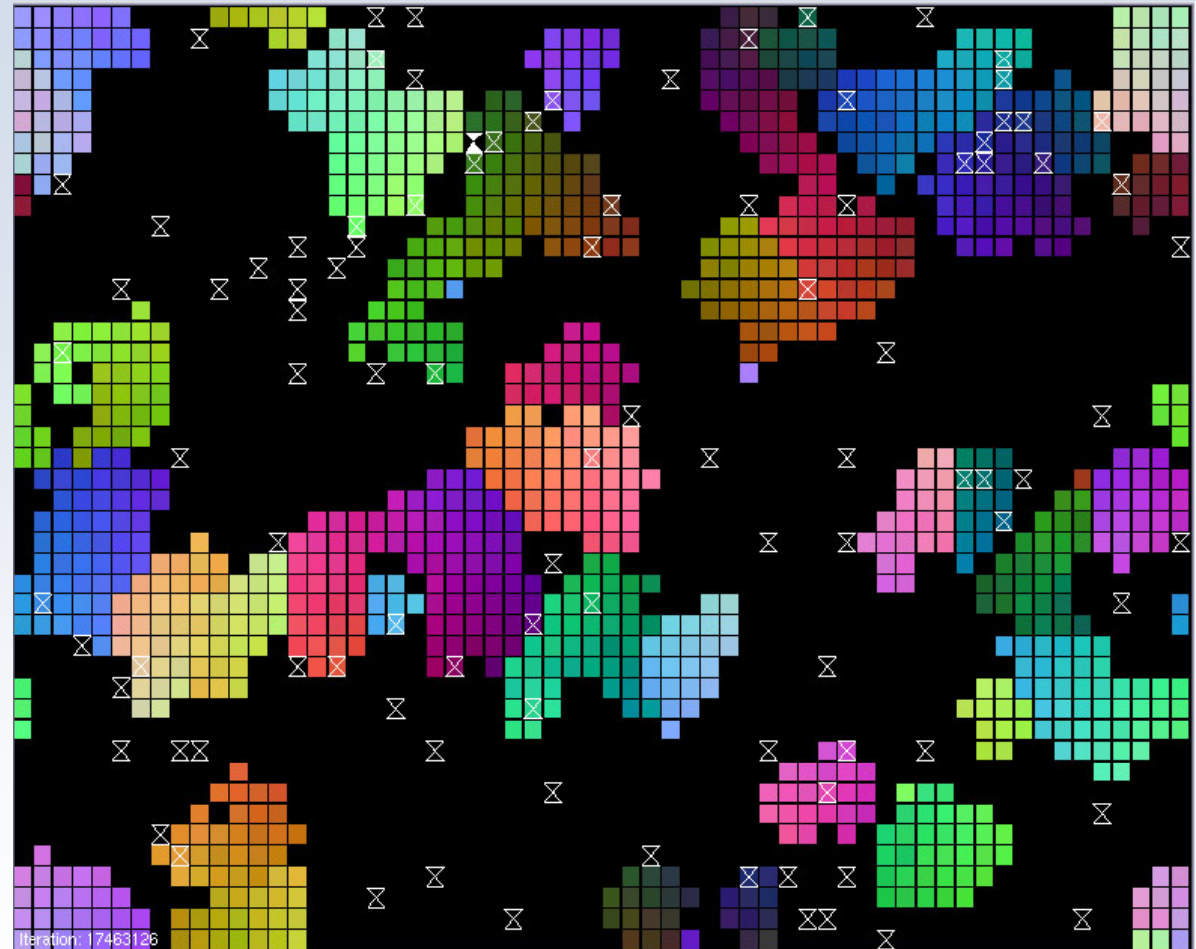
2. Prototypen



50.000 Iterationen



100.000 Iterationen



17.500.000 Iterationen

2. Prototypen

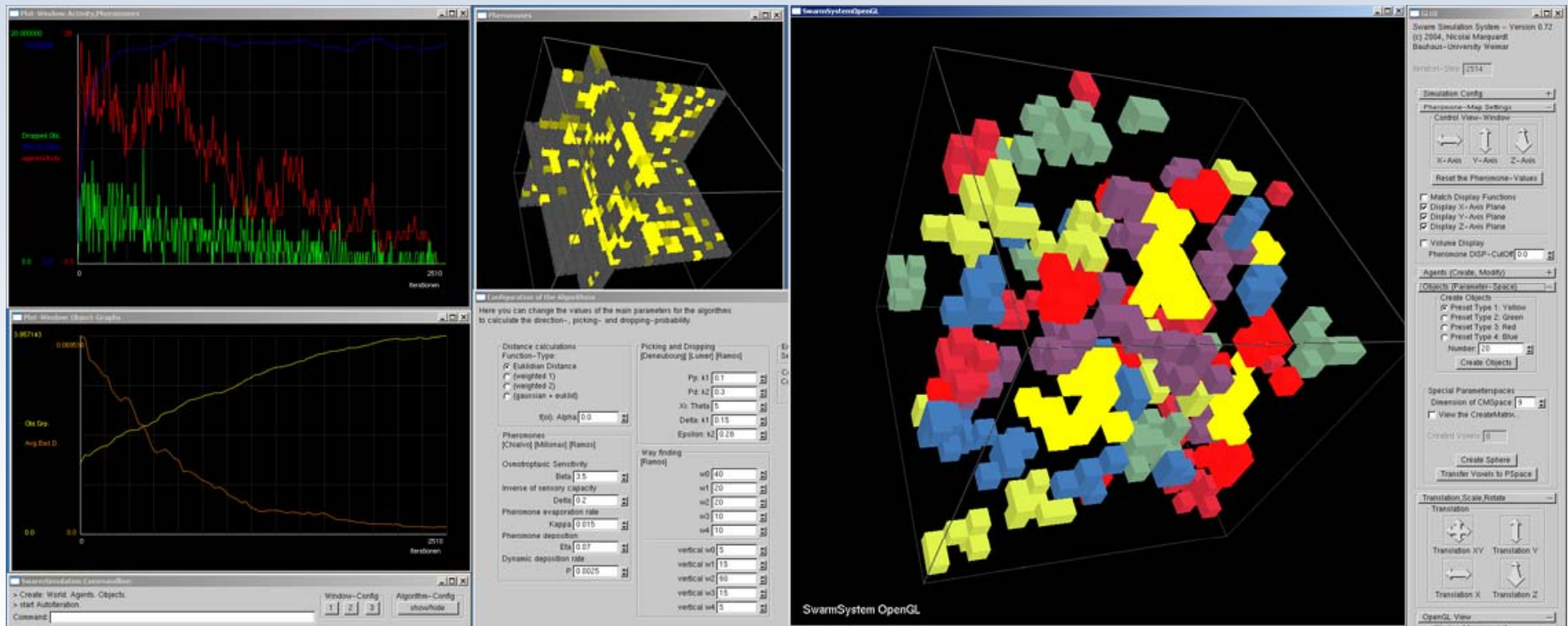
DEMO

Framework: 3D

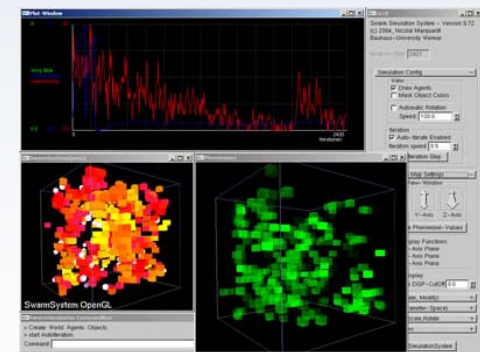
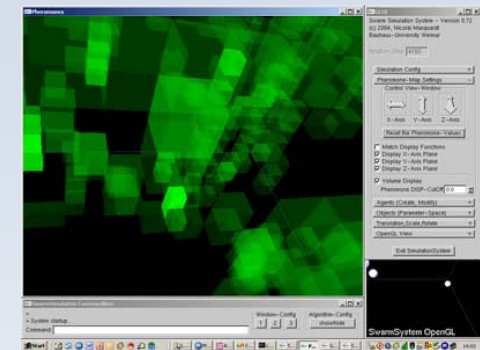
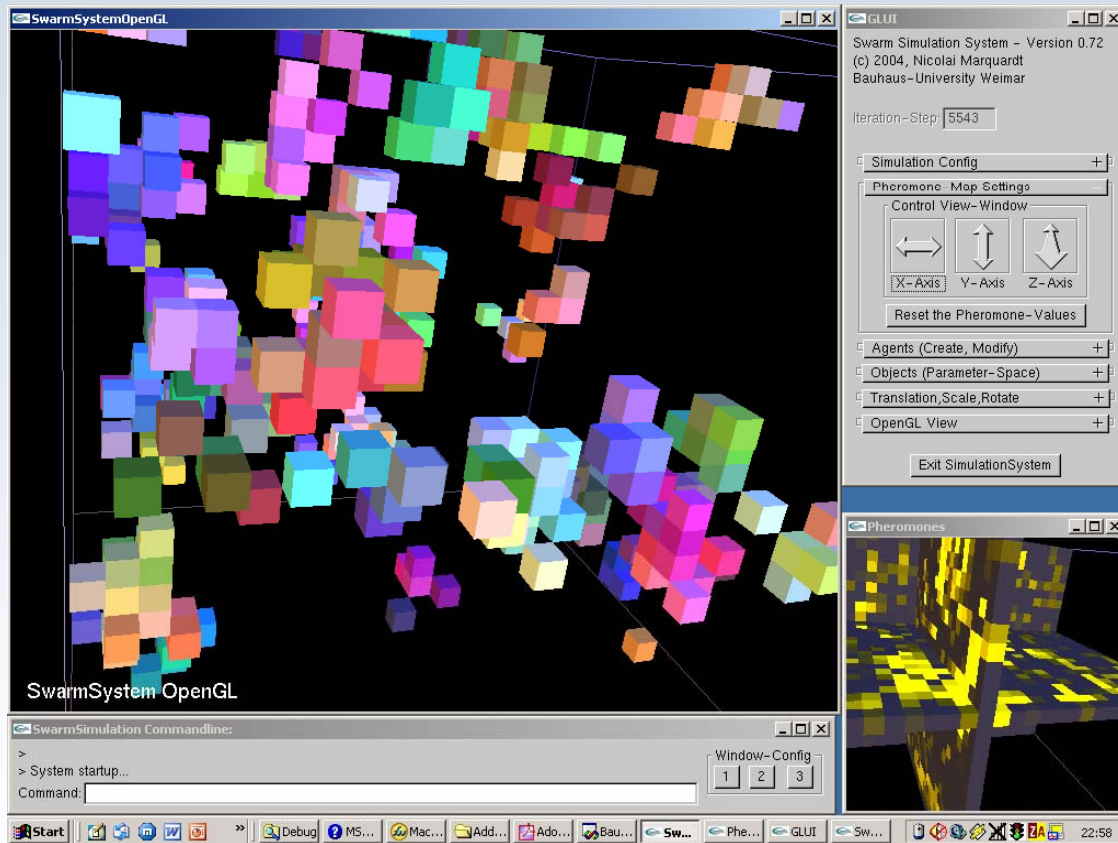
Programm starten:



2. Prototypen



2. Prototypen



2. Auswahl Aufgetretener Probleme

- Bildung von Clustern geringer Größe
- Parameter-Optimierungen schwierig
- Formeln für 3D-Richtungswahl in Kombination mit der einfließenden Pheromon-Gewichtung
- Optimierungen des Pheromonfeldes und dessen Nutzung
- Performance der Swarm-Algorithmen
- Wie können die Algorithmen parallelisiert werden?
- Entwicklung einer Entropie-Berechnung für Algorithmen-Bewertung

3. Ausblick, weitere Forschung

3. Ausblick, weitere Forschung

1. Parameter-Optimierung

- Pheromonfeld-Parameter optimieren
- Variable Einstellungen der Algorithmen optimieren (mit Hilfe der Auswertungs-Funktionen und –Graphen)
- Genetische Algorithmen

2. Agenten-Erweiterungen

- Objekte austauschen (Sortierung)
- Variierende Geschwindigkeiten
- Unterschiedliche Grenz- und Schwell-Werte
- Kurzzeitgedächtnis
- Pheromon-Arten

3. Ausblick, weitere Forschung

3. Datensätze

- Schnittstellen, Import, Konvertierungen
- Geeignete Anwendungsfälle und Daten
- Interpretation

4. Hardware-Beschleunigung

- CG-Programmierung
- Brook-Toolkit
- Algorithmen parallelisieren (abändern, optimieren)

3. Ausblick, weitere Forschung

5. Entropie

- Berechnungen für Entropie-Maße entwickeln
- Interpretation der Graph-Entwicklungen (Plot der Entropie-Berechnungen und Aktivitäten der Agenten)

6. Evaluierung

- Vergleich der Ant-Algorithmen
- Benchmarking mit anderen Clustering-Algorithmen

Vielen Dank
für Ihre Aufmerksamkeit!