

Alfred Roulier Tulpenweg 12 3176 Neuenegg	10.9.2016
Vogelschwarm	Spektrum 2012/1 C.W. Reynolds D.Pearce

Vogelschwärme sind faszinierende Beispiele von Selbstorganisation in der Natur. Ein eindrückliches Zeugnis findet sich im Internet unter <http://www.gesichtskirmes.net/watch/992-300000-stare-im-flug/>. Im *Spektrum der Wissenschaft*, Januar 2012, gibt Bryan Hayes im Artikel „Rätselhafte Koordination in Vogelschwärmen“ eine Übersicht über den Stand der Forschung und Hinweise dazu, wie man solche Phänomene mit dem Computer simulieren kann. Am Anfang solcher Analysen stand ein Programm von **Craig W. Reynolds** <http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>

Der Kern einer solchen Simulation ist einerseits die Annahme, dass sich ein bestimmter Vogel nur nach dem Verhalten seiner lokalen Nachbarschaft von etwa 7 Vögel richtet und andererseits innerhalb dieser Nachbarschaft 3 Regeln beachtet, nämlich **Separation** (Einhalten eines Minimalabstands), **Geschwindigkeitsangleichung** (Betrag und Richtung) und **Kohäsion** (den Mittelpunkt der lokalen Nachbarschaft ansteuern).

Im *Spektrum der Wissenschaft*, September 2014 wird auf eine Arbeit von **Daniel Pearce** hingewiesen, wonach eine andere Kohäsionsregel (Richtung zu maximaler Information fliegen) noch realistischere Simulationen ermöglicht.

http://www2.warwick.ac.uk/newsandevents/pressreleases/the_mystery_behind/

Wir setzen uns hier das Ziel, mit den Mitteln des TI-nspire die Pearce-Regeln in einer einfachen 2-dimensionalen Simulation umzusetzen. Das Schwarmverhalten kann nur in einer dynamischen Simulation befriedigend analysiert werden, weshalb wir sie in LUA programmieren.

Ausgangslage

Es werden nv Vögel zufällig kollisionsfrei (Abstände $>$ Grenzwert $dmin$) mit der Dichte ρ in einen Kreis mit Radius $rstart = dmin * \sqrt{\frac{nv}{\rho}}$ um den Koordinatennullpunkt gesetzt. Die Koordinatenlisten seien xk und yk . Die Vögel erhalten alle dieselbe Geschwindigkeit v und eine zufällige Flugrichtung φ

Flugrichtung

Die Winkel φ werden von der positiven x-Achse aus abgemessen und geben die Richtung von einem Startpunkt $\{x_s, y_s\}$ zu einem Zielpunkt $\{x_z, y_z\}$ an. Dabei muss man beachten, in welchem Quadranten die Punkte liegen. Da diese Berechnung an mehreren Stellen im LUA-Programm vorkommt, schreiben wir eine Funktion:

```

function phirad(x,y)
  local phi=math.atan(y/x)
  if x<0 then
    phi=phi+math.pi
  elseif y<0 then
    phi=phi+2*math.pi
  end
  return phi
end

```

Die Umrechnung erfolgt dann durch den Aufruf $phirad(xz-xs,yz-ys)$

Die Simulation läuft in Zeitschritten der Dauer dt ab. In jedem Zeitschritt fliegt jeder Vogel j die Strecke ds , und seine Flugrichtung φ_{neu} wird berechnet. Die Koordinaten nach dt sind dann

$$xk(j)_{neu} = xk(j)_{alt} + ds \cdot \cos(\varphi_{neu}); yk(j)_{neu} = yk(j)_{alt} + ds \cdot \sin(\varphi_{neu})$$

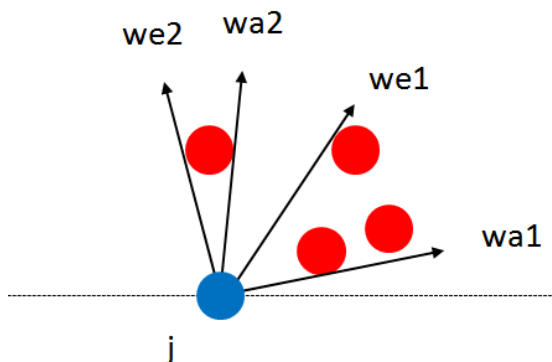
Die Flugrichtung φ_{neu} ist die gewichtete Summe von 4 Komponenten **phib**, **phik**, **phin** und **phir** mit den Gewichten **gb**, **gk** und **gn**, wobei $gb+gk+gn = 1$. Wichtig: es werden nicht Winkel addiert, gemittelt und gewichtet, sondern Vektoren.

$$\begin{aligned}
 hx &= gb \cdot \cos(phib) + (1 - gb) \cdot (gk \cdot \cos(phik) + gn \cdot \cos(phin)) + (1 - gk - gn) \cdot \cos(phir) \\
 hy &= gb \cdot \sin(phib) + (1 - gb) \cdot (gk \cdot \sin(phik) + gn \cdot \sin(phin)) + (1 - gk - gn) \cdot \sin(phir)
 \end{aligned}$$

$$\varphi_{neu} = phirad(hx, hy)$$

phib ist die Richtung der **Kollisionsbremse**, welche vom nächsten Nachbarn auf den Vogel j zeigt. Wenn der Abstand kleiner als die Dimension b eines Vogels wird, ist $gb = 1$ und φ_{neu} wird zwingend **phib**. Ist der Abstand hingegen grösser als $k \cdot b$, k ist vorzugeben, wirkt keine Kollisionsbremse, $gb = 0$. Dazwischen besteht ein linearer Übergang. Die Kollisionsbremse ist im Modell Pearce zwar erwähnt, aber nicht eingebaut. Ohne sie entstehen aber unrealistische Klumpenbilder. Zudem sind sie ein Kernelement realer Schwärme, und weil der Aufwand klein ist, haben wir sie so eingebaut.

phik ist die **Kohäsionskomponente**. Ein Vogel j versucht so zu fliegen, dass er am meisten mitfliegende Vögel sieht.



In der Figur wird gezeigt, wie die roten Vögel das Gesichtsfeld des blauen Vogels j sektorweise abdecken. Die Sektoren sind durch die Vektoren **wa** und **we** begrenzt. Die gesuchte Kohäsionsrichtung ist die Richtung des Mittels dieser Vektoren.

Zur Berechnung werden zuerst die Listen der Begrenzungsvektoren der roten Vögel, als Kreise mit Durchmesser b angenommen, aus der Sicht des blauen Vogels j bestimmt. Diese Listen werden danach reduziert, indem alle Überlappungen eliminiert werden. Pearce empfiehlt das Gewicht $gk = 0.1$

phin ist die mittlere Flugrichtung der **nn** nächsten Nachbarn. Für jeden Vogel **j** wird die Liste **dlist** der Abstände zu seinen **nn-1** Nachbarn berechnet und zusammen mit der Liste **ilist** der Indizes der Vögel aufsteigend geordnet. Die ersten **nn** Elemente dieser Listen bilden dann die lokale Nachbarschaft.

Pearce verwendet in seinen Simulationen $nn = 4$ und ein Gewicht $gn = 0.7$. Auch hier werden **nn** Vektoren und nicht Winkel gemittelt.

Die Simulation von C. Reynolds berücksichtigt, dass die Vögel nicht rundum sehen, d.h. im Schwanzbereich einen toten Blickwinkel haben. Dieser Umstand wäre eigentlich bei der Berechnung von **phin** und **phik** einzubauen. Pearce hat festgestellt, dass damit kein wesentlich anderes Schwarmverhalten resultiert und hat daher wie wir darauf verzichtet.

phir ist eine zufällige Rauschkomponente im Bereich $0 < phir < 2 \cdot \pi$ mit kleinem Gewicht $1 - gk - gn$.

Escapemode

Nach einer Einschwingzeit kann man mit einem Button einen Raubvogel (rot) starten mit Ziel Schwarmmitte und gegenüber den Schwarmvögeln doppelter Geschwindigkeit. Wenn er das Zentrum erreicht hat, fliegt er geradeaus weiter. Wenn der Abstand des Vogels **j** zum Raubvogel die Fluchtdistanz unterschreitet, flieht **j** mit 50% höherer Geschwindigkeit senkrecht weg vom Kurs des Raubvogels. Dieses Szenario kann man nach einiger Zeit (z.Z. 130 Zeitschritte) wiederholen.

Simulationsparameter

Schwarmparameter

dt :=0.2 • 0.2	Dauer Zeitschritt (s)
nv :=100 • 100	Anzahl Vögel
rho :=0.1 • 0.1	Dichte (Anzahl Vögel pro m ²)
b :=0.3 • 0.3	Dimension eines Vogels (m)
ds :=2•b • 0.6	Strecke pro Zeitschritt (m)
fd :=12•b • 3,6	Fluchtdistanz (m)
kb :=3•b • 0.9	Reichweite Kollisionsbremse (m)
nn :=4 • 4	Anzahl Vögel in lokaler Nachbarschaft
gk :=0.1 • 0.1 gn :=0.75 • 0.75	Gewichtung Projektion/Nachbarschaft ($gk+gn < 1$) (empfohlen: $gk=0.1, gn=0.75$)
mg :=1 • 1	Koordinatenliniengitter ein/aus (mit: $mg=1$, ohne : $mg=0$)
abstgl :=5 • 5	Abstand benachbarter Gitterlinien in Meter
startwinkel := -1 • -1	Ausrichtung jedes Vogels in der Startkonfiguration, $0 \leq \text{startwinkel} \leq 2\pi$ startwinkel<0: Zufällige Ausrichtung der Vögel beim Start
startkon() • Fertig	

Die Simulationsparameter werden in einer Notes-Seite eingegeben. Damit sie im LUA Programm wirksam werden, muss das LUA Skript neu festgelegt werden.

Anzahl Vögel nv, Zeitschritt dt : Die Verhaltensregeln wirken umso besser, je grösser **nv** ist. Aber der Rechenaufwand pro Zeitschritt wächst proportional zu nv^2 , weil die Wechselwirkung eines Vogels mit allen anderen zu prüfen ist. Bei zu grosser Vogelzahl vermag der Rechner nicht mehr, innerhalb von **dt** alle Programmschritte umzusetzen. $nv = 100$ und $dt = .2$ s sind für unseren Rechner (HP Compaq 6730b, 2.5 GHz) ein guter Kompromiss. Mit der Stoppuhr wurde überprüft, dass die Simulationsdauer gleich der Anzahl Zeitschritte mal **dt** ist.

Vogelgrösse b : $b = .3$ entspricht etwa der Flügelspannweite eines Stars.

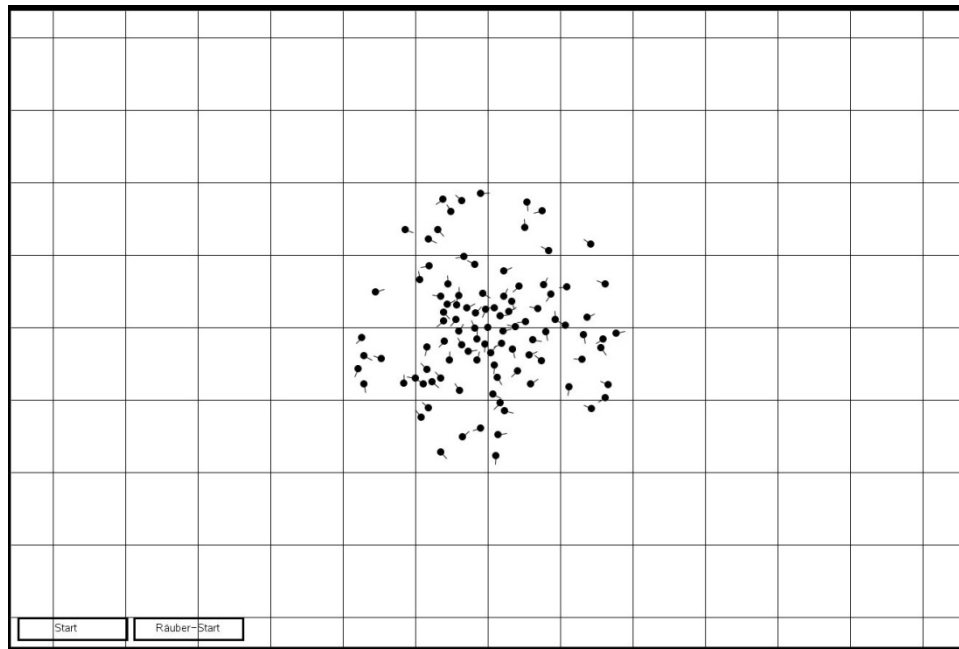
Flugstrecke ds pro Zeitschritt dt : $ds = 2b$ ist eigentlich zu kurz; denn Stare fliegen schneller als 10 km/h. Mit diesem Wert wird somit ein Zeitlupeneffekt erzielt, was das Geschehen auf dem Bildschirm merklich erleichtert.

Reichweite der **Kollisionsbremse kb** : Ohne Kollisionsbremse dominiert die Kohäsion, und es entstehen Zusammenballungen. Ein zu weiter Bereich zieht den Schwarm quer zur Flugrichtung in die Breite. Der Bereich von $kb = b$ (zwingend volle Bremse) bis $kb = 3 \cdot b$ (keine Wechselwirkung) ergibt ein plausibles Schwarmbild.

Lokale Nachbarschaft nn : Es ist kein grosser Unterschied zwischen $nn = 7$ (Reynolds) und $nn = 4$ (Pearce) zu sehen. Somit wählen wir 4 und sparen etwas Rechenzeit.

Gewichtung gk und gn : Pearce empfiehlt das Wertepaar $\{gk = 0.1, gn = 0.75\}$. Schon nach wenigen Zeitschritten bildet sich eine globale Flugrichtung je nach Ausgangssituation aus.

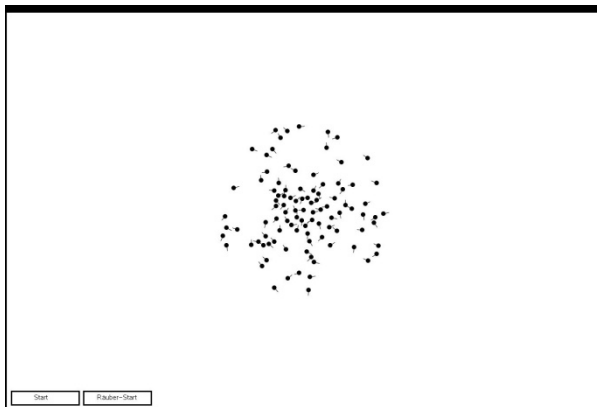
Bildschirm



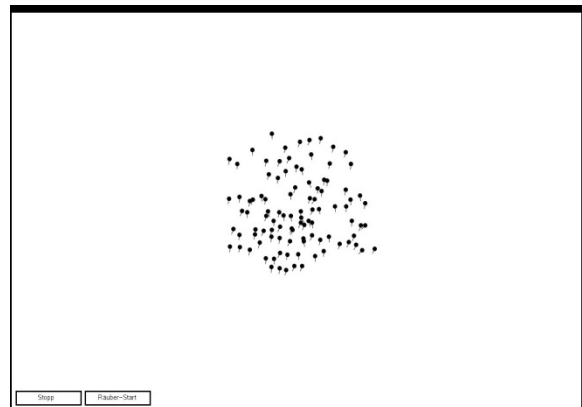
Die Vögel sind stilisiert mit Kopf(Kreis) und Schwanz (Strich) gezeichnet. Das Schwarmzentrum ist immer in der Bildmitte, und das Koordinatennetz mit Maschenweite 5 Meter bewegt sich im Hintergrund.

Simulationsbeispiel

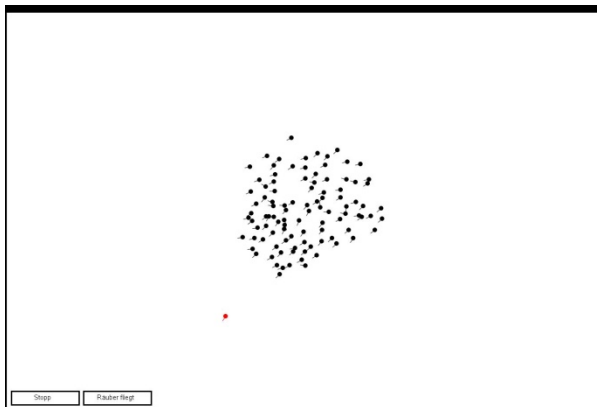
TI-nspire Screenshots einer Simulation mit den obigen Parametern :



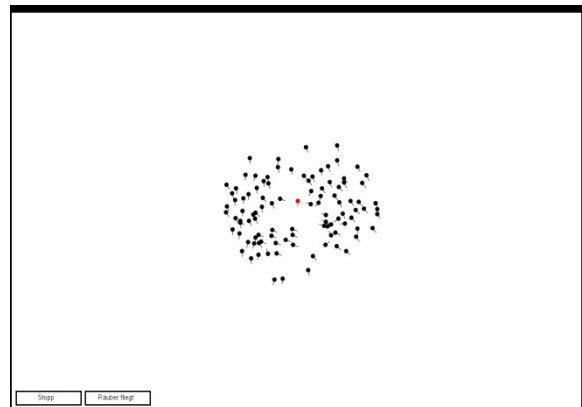
Ausgangslage mit Minimalabstand 2-b



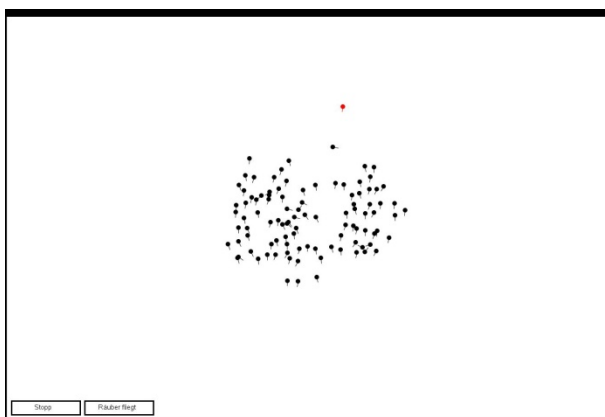
Nach ca 100 Zeitschritten. Der Schwarm fliegt geschlossen nach oben



Der Raubvogel erscheint unten links



Der Raubvogel hat die Schwarmmitte erreicht



Nach dem Durchgang des Raubvogels schliesst sich der Schwarm wieder

Nach Start bildet sich rasch eine allgemeine Flugrichtung aus. Der Schwarm bleibt kompakt und es spalten sich keine Untergruppen ab. In Zeitabständen von 50 – 100 Zeitschritten kann sich spontan die Flugrichtung ändern, genau so, wie man es auf Realmovies sehen kann. Nach einem Räuberangriff formiert sich der Schwarm relativ rasch wieder neu.