

# 洪水解析モデルとマルチエージェントモデルのカップリング

木村 一郎  
Ichiro KIMURA

北海道大学大学院 工学研究院 准教授

## 1. はじめに

### (1) 本章の背景

平成23年3月11日に未曾有の地震と津波が東北地方を襲った。従来では防潮堤や防波堤などのハードでそのような津波や洪水を極力軽減するとなされていたが、物理的に防御できない津波が存在することを教えられた。大自然災害を完全に封鎖することができるという思想ではなく、災害時の被害を最小限化する「減災」という考え方が取り上げられ、これまでのように水際での構造物に頼る防御から、「逃げる」ことを基本とする防災教育の徹底やハザードマップの整備など、ソフト面の対策を重視する方針へと転換された。

一方、一時は減った洪水による被害者数が、近年再び増加しつつある。その背景として高齢化が挙げられる。すなわち、一人では避難できない高齢者が増えていることが、その被害の増大の一因として挙げられる。高齢者の適切な避難誘導のあり方が問われている。

現在、避難の検証にはマルチエージェントシステムが用いられているものが多い。マルチエージェントシステムは交通分野、計画分野で用いられているが、津波や、洪水などの解析と合わせた避難のあり方について検証されているものというものは残念ながら少ない。

### (2) 研究の目的

氾濫や津波遡上などの流れ挙動に関する研究や解析例は膨大で、大規模水害における浸水被害の予測に関するノウハウはある程度確立されている。一方、避難行動についての研究は比較的新しい。そこで、マルチエージェントシステムを用いて避難モデルを構築し、洪水などの水災害における避難のあり方について検証できるツールの開発をめざす。これにより避難行動解析を行い、現状の避難のあり方の可視化だけでなく、どのような情報が避難に有効か、どのようなルートが避難に適しているか、ひとりで避難できないひとをどのようにサポートしていくかということを検証することをめざす。

## 2. 洪水流モデルの概要

### (1) 流れの計算の基礎式

氾濫計算はiRICを用いて行う。iRICの流れの支配方程式はマニュアルを参照されたい。

### (2) 計算格子

後述のRun1において用いる計算格子は50×50のグリッドの直交等間隔格子である。一方、後述のRun2~4においては図-1に示すGrid2を用いる。

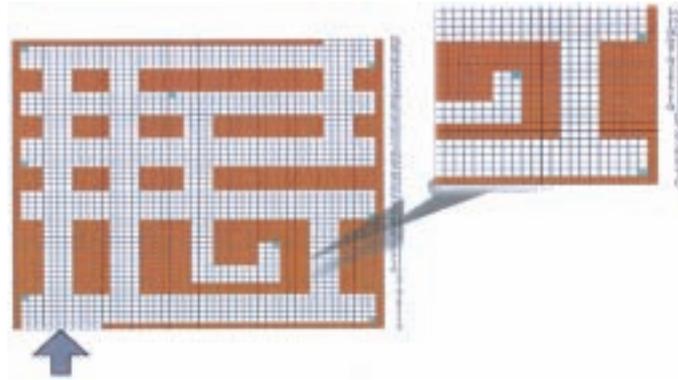


図-1 Grid2

### (3) 洪水モデルとマルチエージェントモデルのカップリング

iRICの計算結果をマルチエージェントシステム(MAS)に受け渡すことでカップリングさせる(図-2). 受け渡されるパラメーターは河床, 水位, 流速, 時間, 空間のX,Y, セル番号などである.

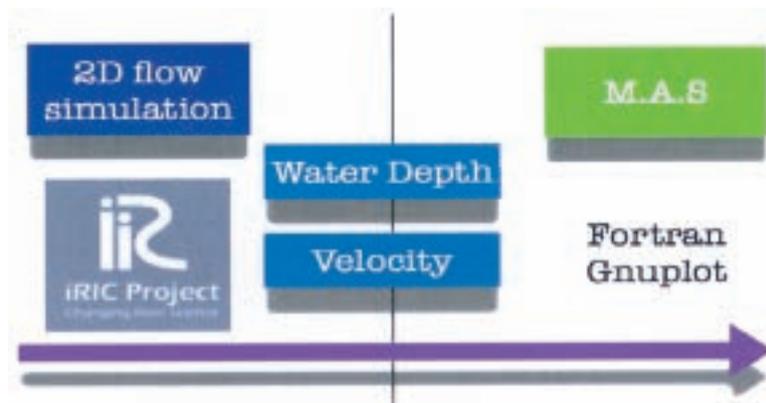


図-2 カップリングについての模式図

## 3. マルチエージェントシステムの概要

### (1) マルチエージェントモデル (MAS) の概要

マルチエージェントシステム (Multi-Agent System, MAS) とは複数のAgent(人)から構成されるシステムで, 個々のエージェントやモノリシックなシステムでは困難な課題をシステム全体として達成する. それぞれ異なった判定アルゴリズムなどの特徴 (キャラクタースティック) を持ったエージェントモデルを用い (よってマルチエージェントと呼ばれる), 一定数以上のエージェントを多数設定することで人工社会を構成し, それぞれ特徴の異なったAgentの相互作用をシミュレーションするシステムである. マルチエージェントモデルにはM.A.S.というシミュレーションソフトが存在するが, データの読み込みの量が決まっていること, データの書き出しの融通性の欠如, M.A.S.に関するマニュアルや書籍が少ないなど, 必ずしも使い勝手がよくない. そこで, Fortranを用いて一からモデルを構築することとした. Fortranにおいては構造体を用いることで各々のAgentはX座標, Y座標, 速さ, 角度, 目指している避難所の番号, 避難しているか否かなどの情報を一括して扱い, プログラムを簡略化した. なお, 可視化にはgnuplotを用いた.

### (2) 4つの行動モデル

エージェントの行動について, 今回は次の4つの異なるモデル化を行い, 結果を比較した.

- Run1: 水を感じたら, 避難所へ逃げる

- Run2: 障害物の定義, 視野の定義 +Run1の条件
  - Run3: 視野の再定義 +Run2の条件
  - Run4: 人を見たらついていく +Run3の条件
- 以下, 各モデルの概要について述べる.

#### 4. 結果の考察

##### (1) Run1によるモデル化

- ① Agentの生成: Agentの最大人数を定義すれば, 領域内にランダムにAgentが配置される
- ② 避難前の行動: Agentの向かう方向は乱数を発生させ, ランダムに向かうように定義されている.
- ③ 水の感知: Agentの座標上の水深が0[m]より大きいと水を感じたこととし, 避難を開始させた. 可視化においては水を見た人は赤色, 水を見ていない人は青色としている(図-3).

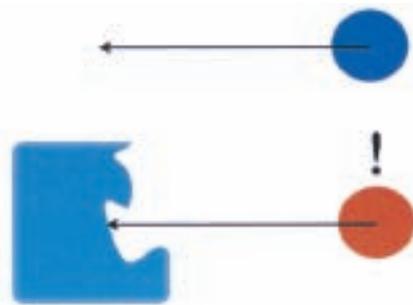


図-3 水の感知

- ④ 最も近い避難所への避難: 視野に入った避難所との角度を求め, その方向へ避難させる(図-4).

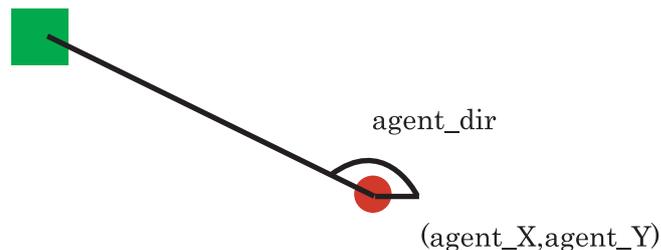


図-4 避難所とAgentの角度の決定

複数避難所があった場合は, 各々の避難所との角度と距離 (RL) を求め, 一番距離が短い避難所へと向かう (図-5).

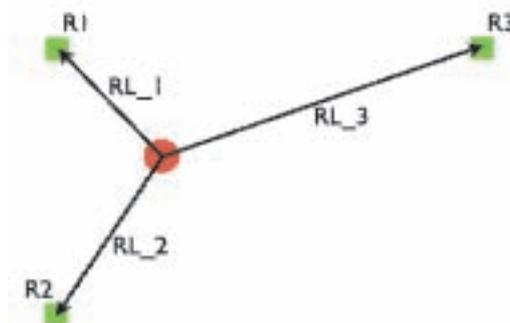
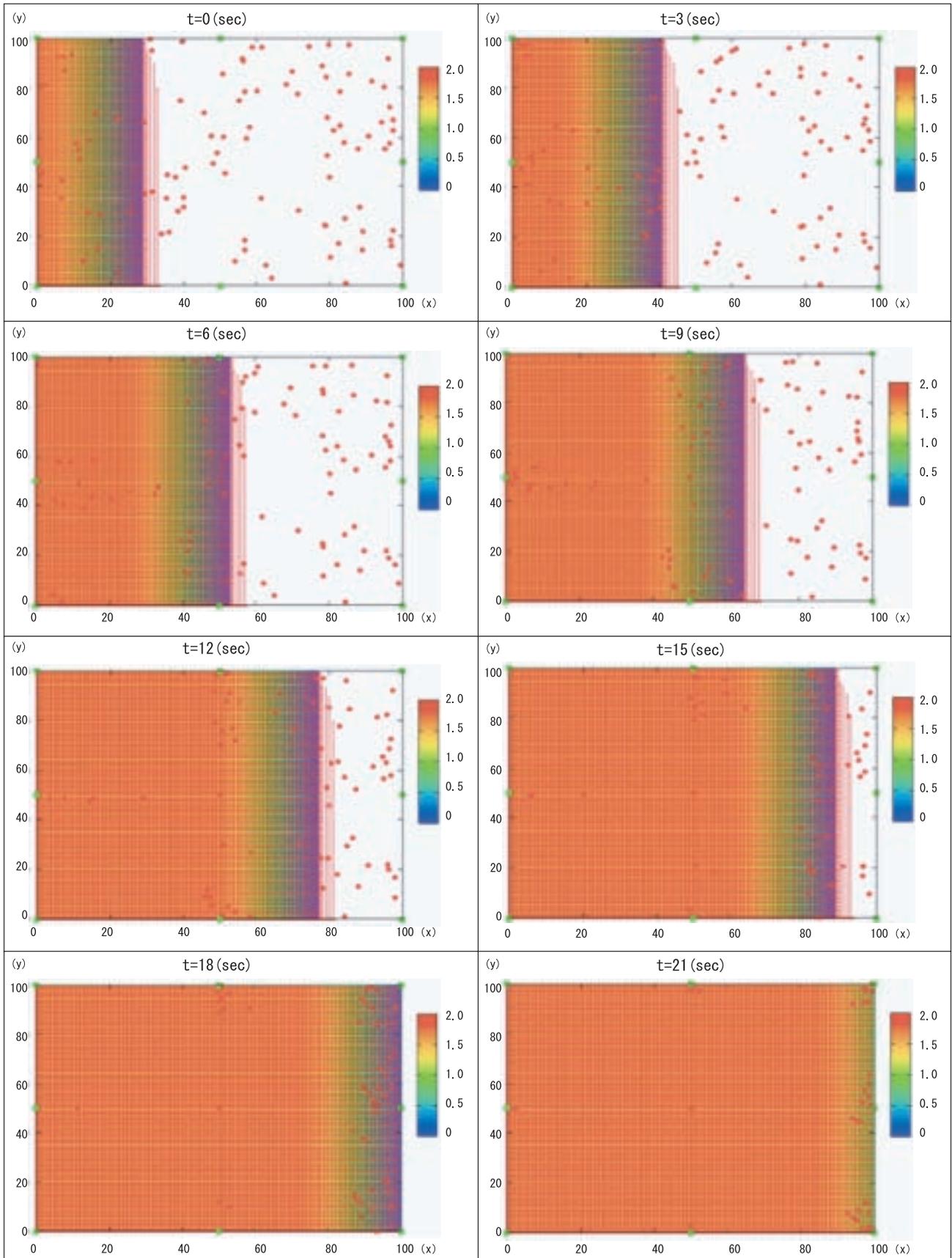


図-5 複数の避難所とAgentの角度と避難所との距離の決定

## (2) Run1による結果とその考察

図-6にRun1による計算結果を示す。これより、Agentが水に触れたら一番近い避難所へ向かって逃げる事が確認でき、適切なモデル化が行われていると判断した。



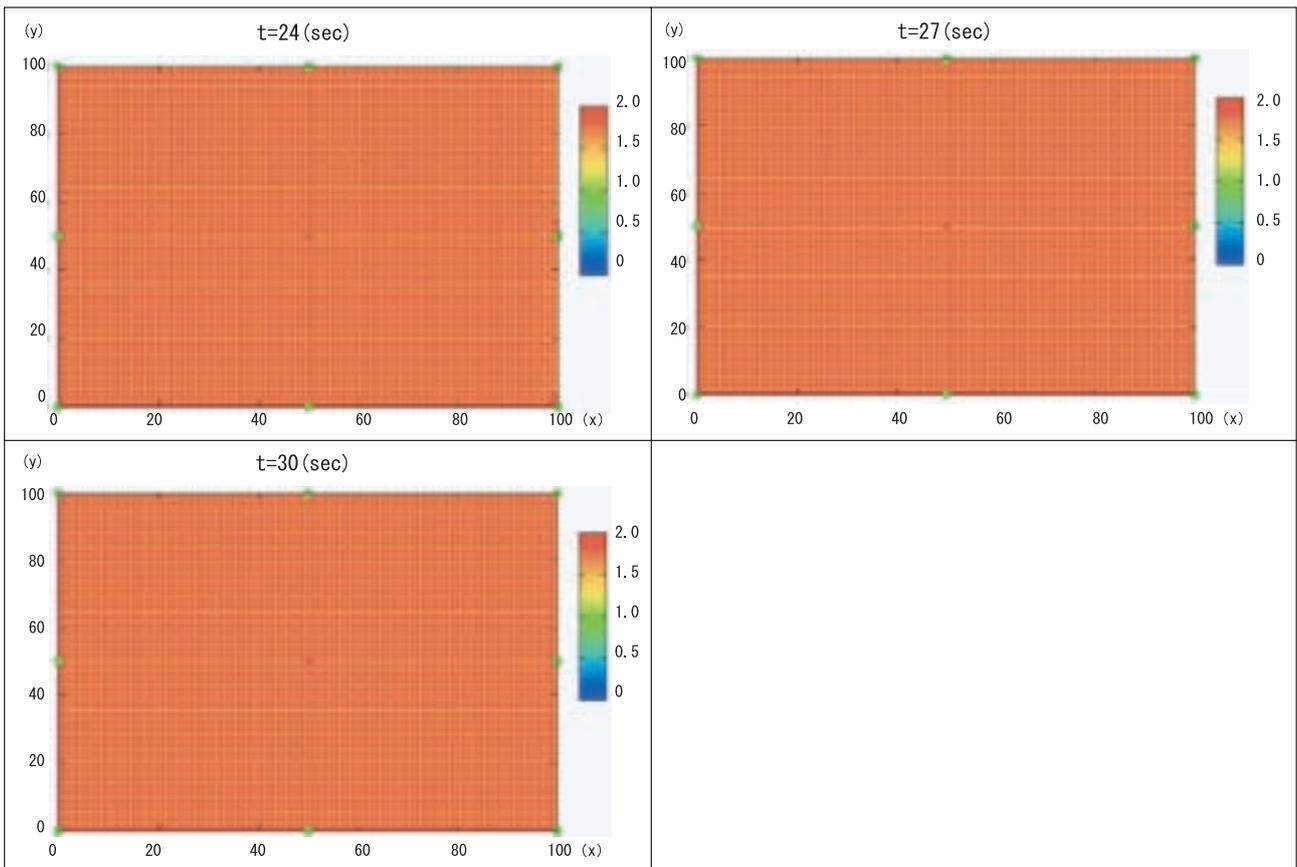


図-6 Run1計算結果

### (3) Run2のモデル化の特徴

#### a) 障害物の回避

Run1では障害物は設定していない。Run2においては障害物を設定し、Agentもその障害物を回避するようにした。もし、障害物を通過した場合は障害物を通過しないように補正する。最初に障害物を通過した面と障害物を通過した際の軌跡の交点を求め、その点からAgentの幅分をずらした場所にAgentを補正するようにした(図-7)。それによって障害物を通過できないようにした。

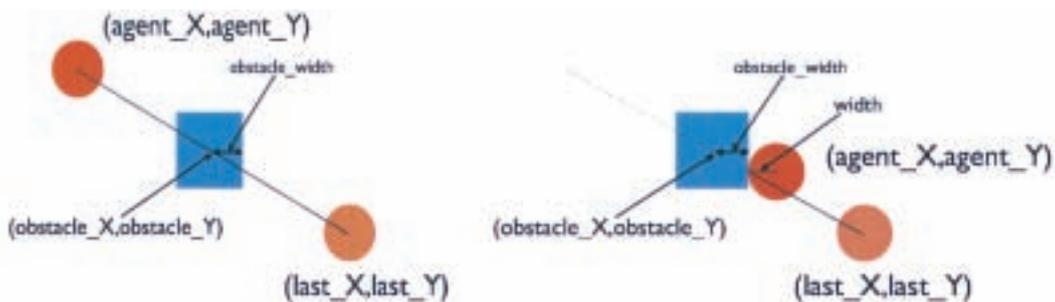


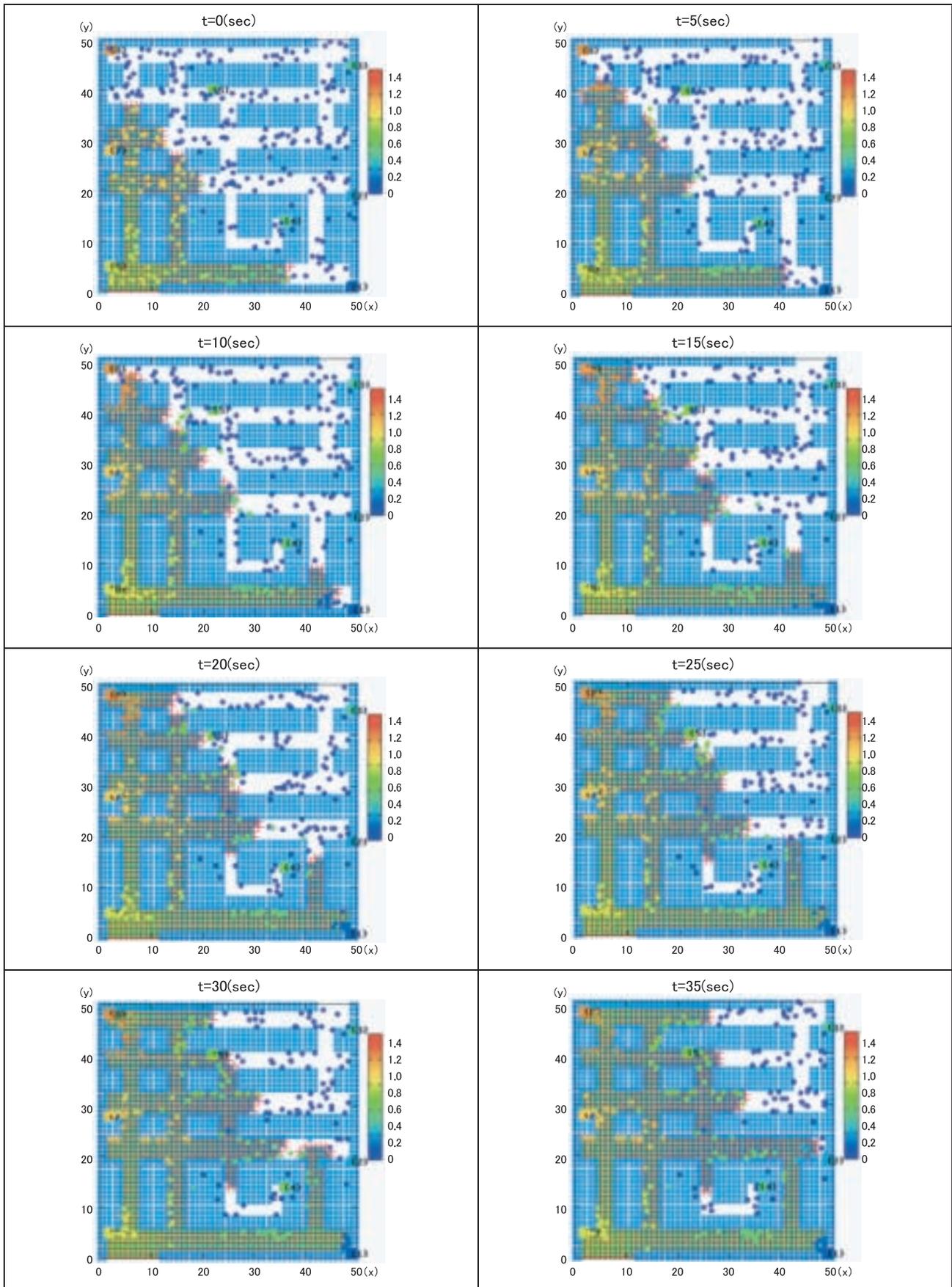
図-7 障害物セルの回避

#### b) 視野の定義

Run1では水に触れると避難を開始するとしていたが、Run2では視野を定義し、視野に水が入ると避難するようにした。Run2においては視野の長さを決め、円状にものが見えるようにした。

#### (4) Run2による計算結果と考察

図-8にRun2による避難シミュレーション結果を示す.



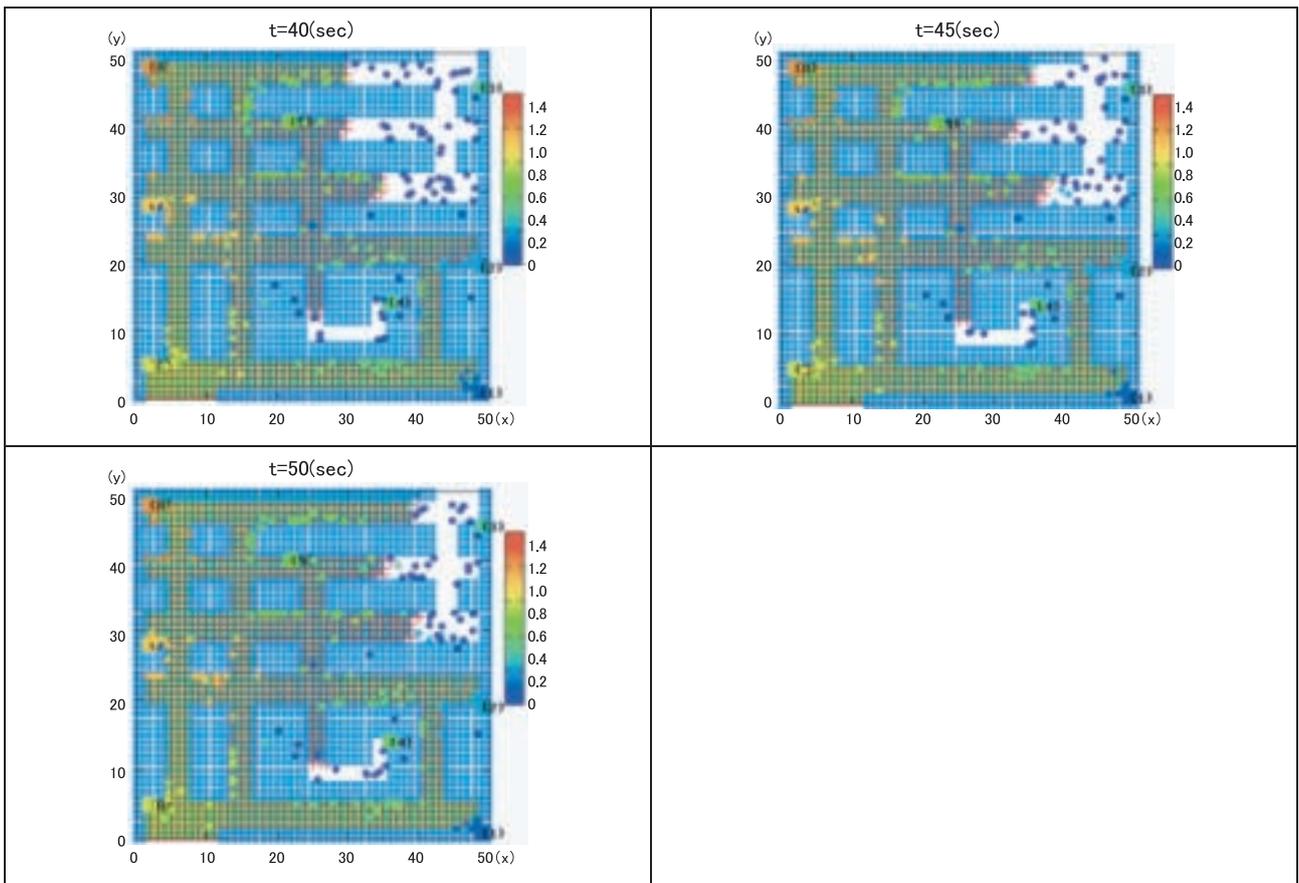


図-8 Run2計算結果

当初は視野長を決めて円として定義していたが、障害物を乗り越えて物が見えることとなってしまった。そのために障害物があるにも関わらず、避難所とagentの最短距離を取って避難しようとするために、agentが壁に張り付いてしまう結果となった(図-9)。

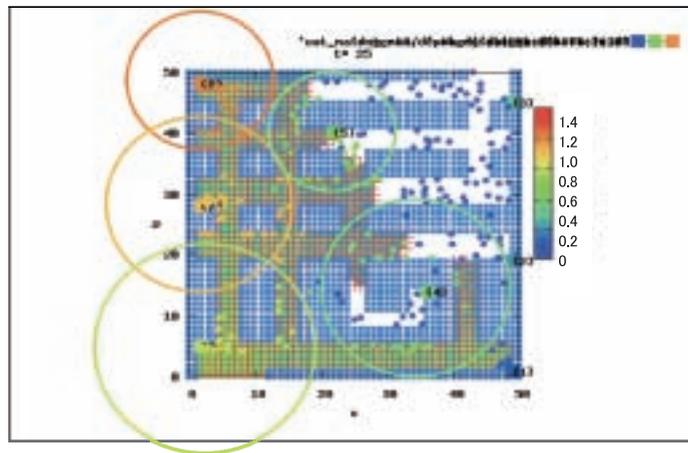


図-9 Run2計算結果ピックアップ

Agentの視野を可視化したのが、図-10である。障害物を貫通してものが見えていることが分かる。この点については、次のRun3で補正を試みる。

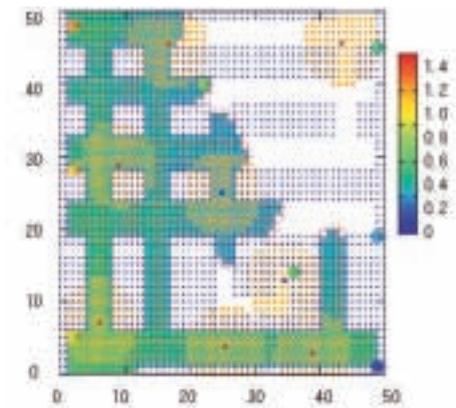


図-10 Run2におけるAgentの視野

### (5) Run3のモデル化の概要

#### a) 視野の定義

障害物を乗り越えてものが見えないようにするために、 $360^\circ$ 見渡して、障害物とぶつかった点を計算し、それによって見える範囲を定義した。

① はじめは視野距離に応じた範囲が見えている (図-11)。

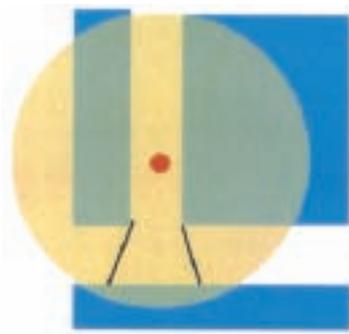


図-11 視野の範囲

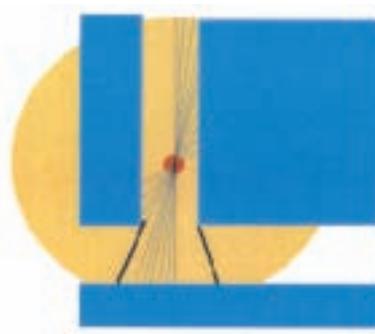


図-12 物体による視野のさえぎり

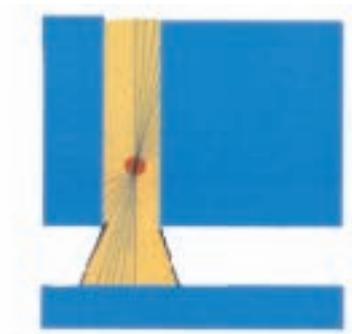


図-13 実際の視野範囲

②  $0^\circ \sim 360^\circ$ まで $1^\circ$ ずつ、障害物までの距離を測り、それぞれの角度から見える一番手前の障害物を探しAgentの距離とその距離を求め、その距離内に物体が見えている範囲とする (図-12)。

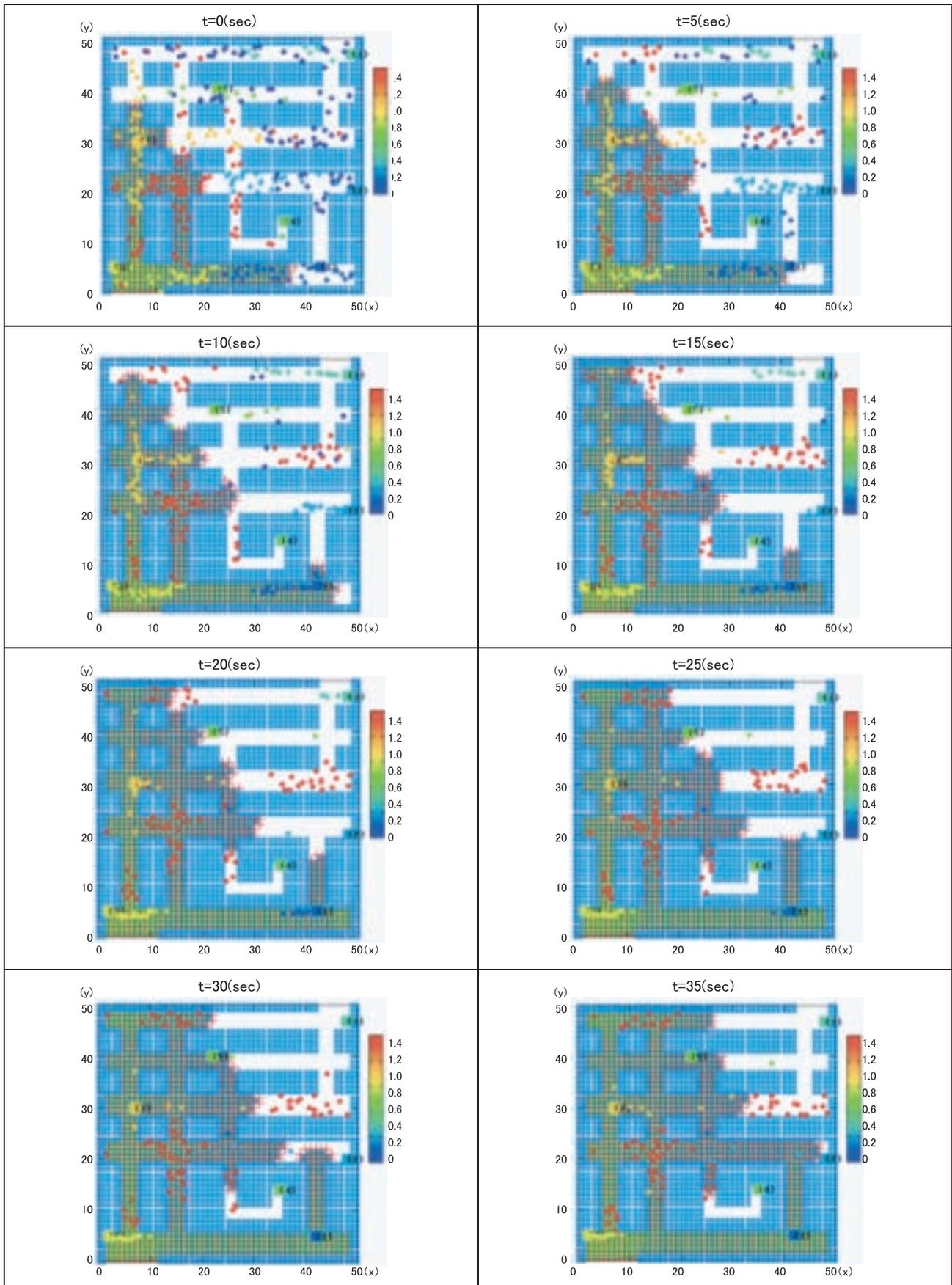
③ このような定義により、図-13のような視野が定義される。

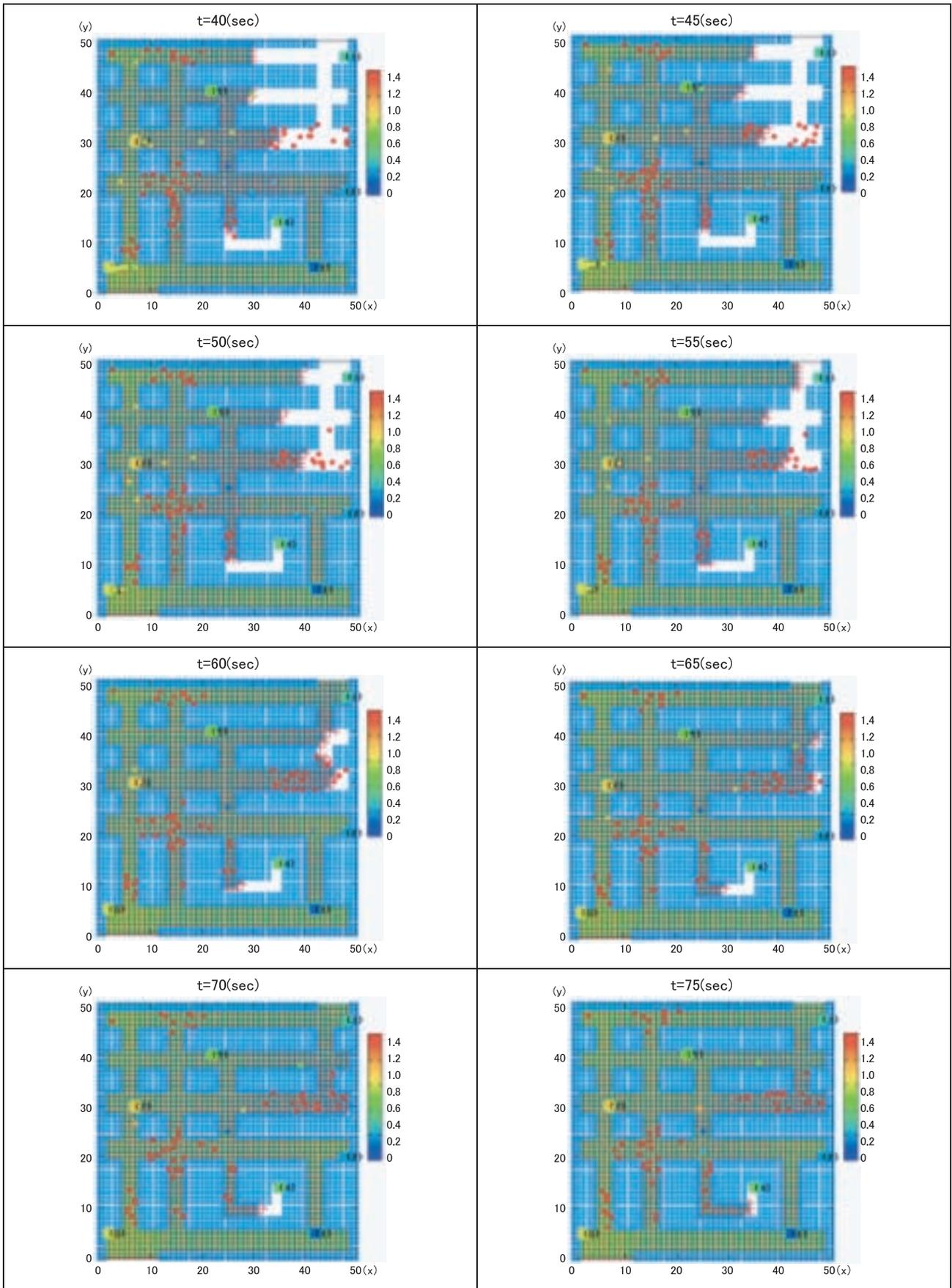
#### b) 初期のAgent配置における障害物との重なり防止

障害物を定義したため、Agentを最初に空間内に配列する際に、障害物とAgentが重なってしまうということを補正するために、障害物の中にAgentが配列される場合は、再度障害物と重ならない場所に配列し直すということを行なっている。

### (6) Run3計算結果

Run3による計算結果を次に示す.





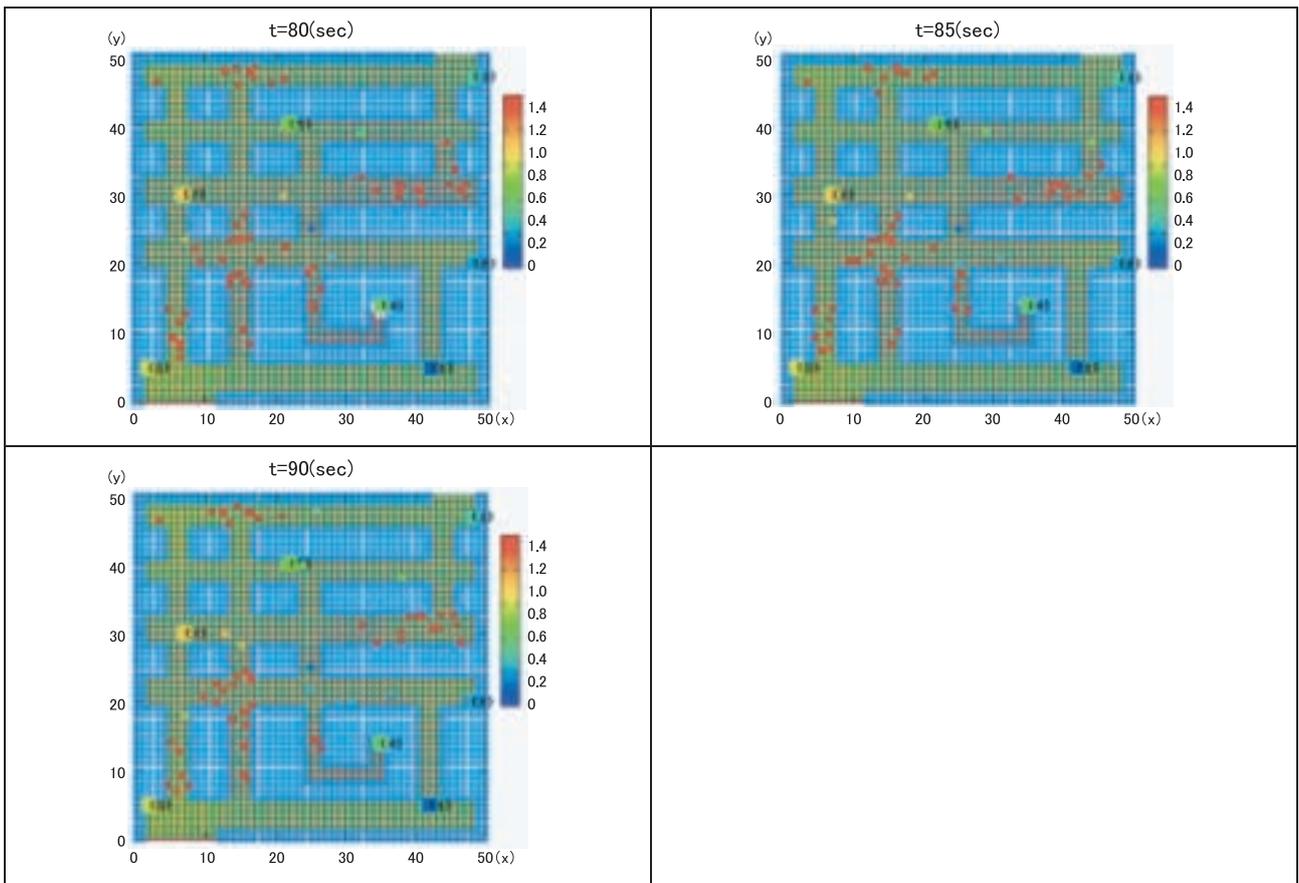


図-14 Run3の計算結果

### (7) Run3 結果と考察

視野が定義されたことにより, Agentは障害物によって避難所が見えない場合は避難所に向かうことができなくなった. よって, 壁にAgentが張り付く問題は解決された. しかしながら, 避難所が見えていない人は避難所が分からないので避難できないままとなった. 赤で示されたAgentは避難所が視野の中に入っていないので, 避難できていないのが分かる (図-15).

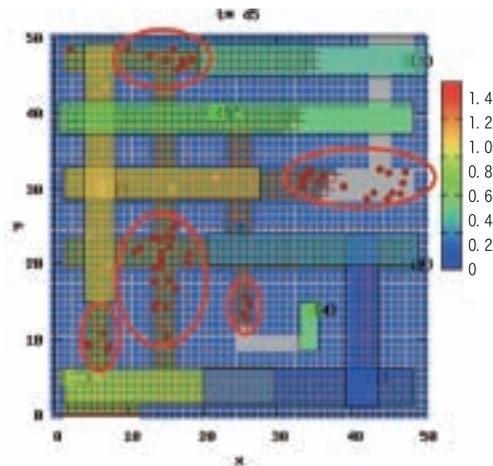


図-15 Run3計算結果における避難所がAgentの視野に入らない場所

## (8) Run4のモデル化概要

### a) 人を見ての避難

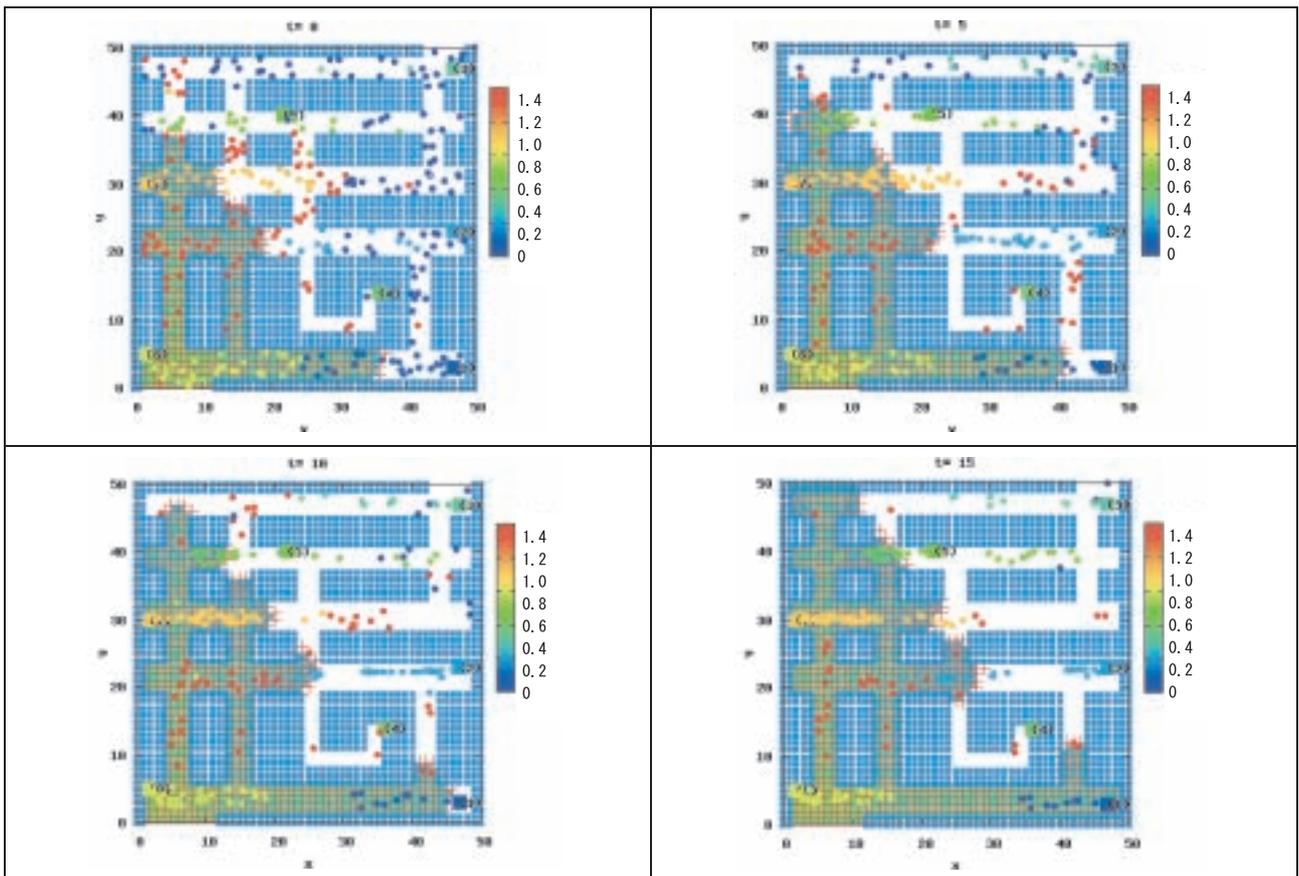
今回は避難所を見て避難している人を見たら、その人の方向へ向かうということを行った。避難所が分からない人はランダムで移動方向を決めるので、避難所が見えるまでランダムで動き続ける。ゆえに避難できない人が出てくる。避難所が分からず、それを探している場合、自分は避難所が分からないが、すでに避難所が分かっているAgentについていくことで、避難を促した。具体的には避難所番号を持っていないAgentが視野の中に入れば、そのAgentとの方向を計算し、そのAgentの方向へ向かうようにする(図-16)。避難所番号を持っているAgentが複数いれば、その中で一番近いAgentに向かって動くようになっている。

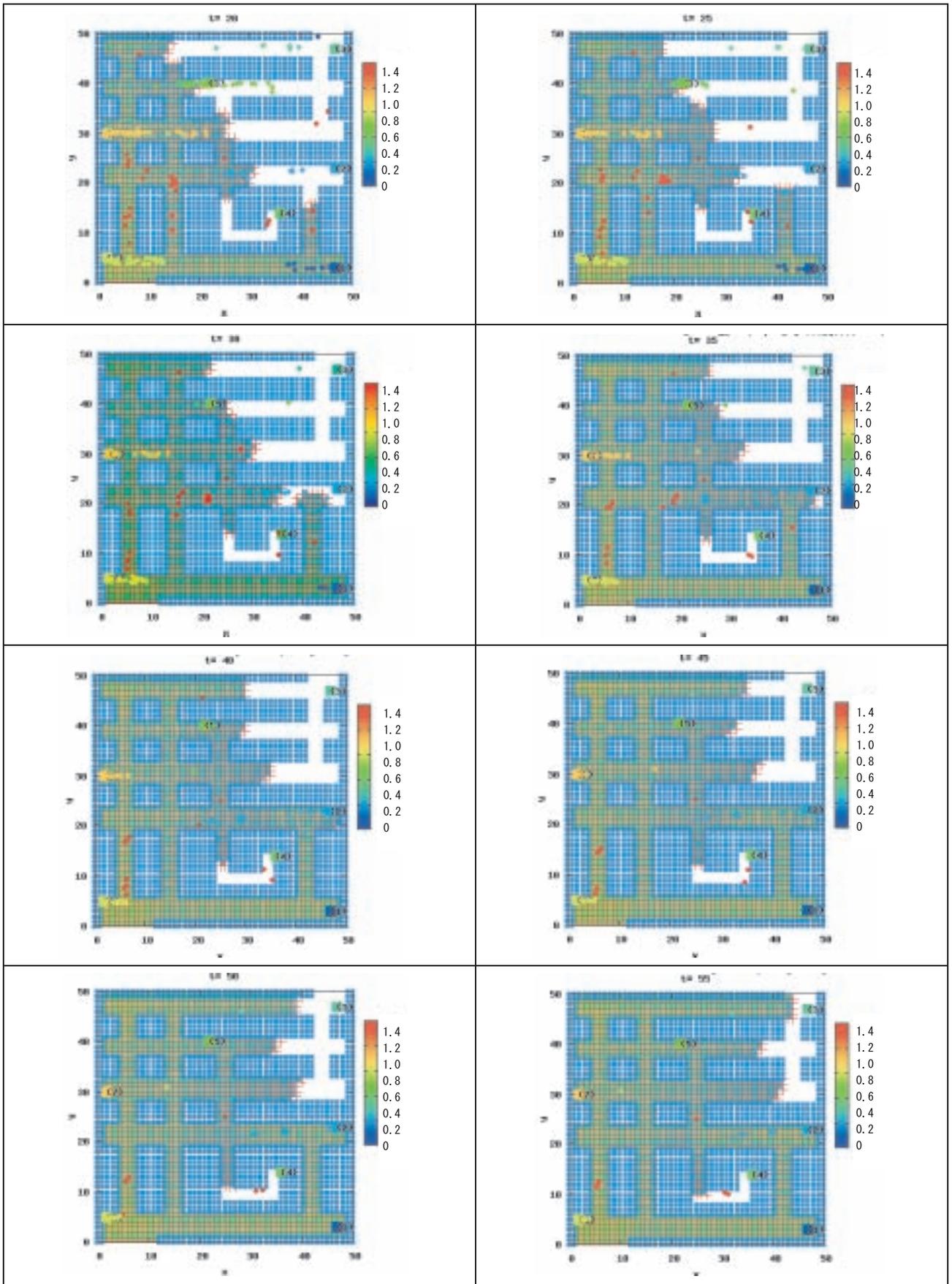


図-16 Run4のモデル化 (人への追従)

## (9) Run4計算結果

Run4による計算結果を図-17に示す。





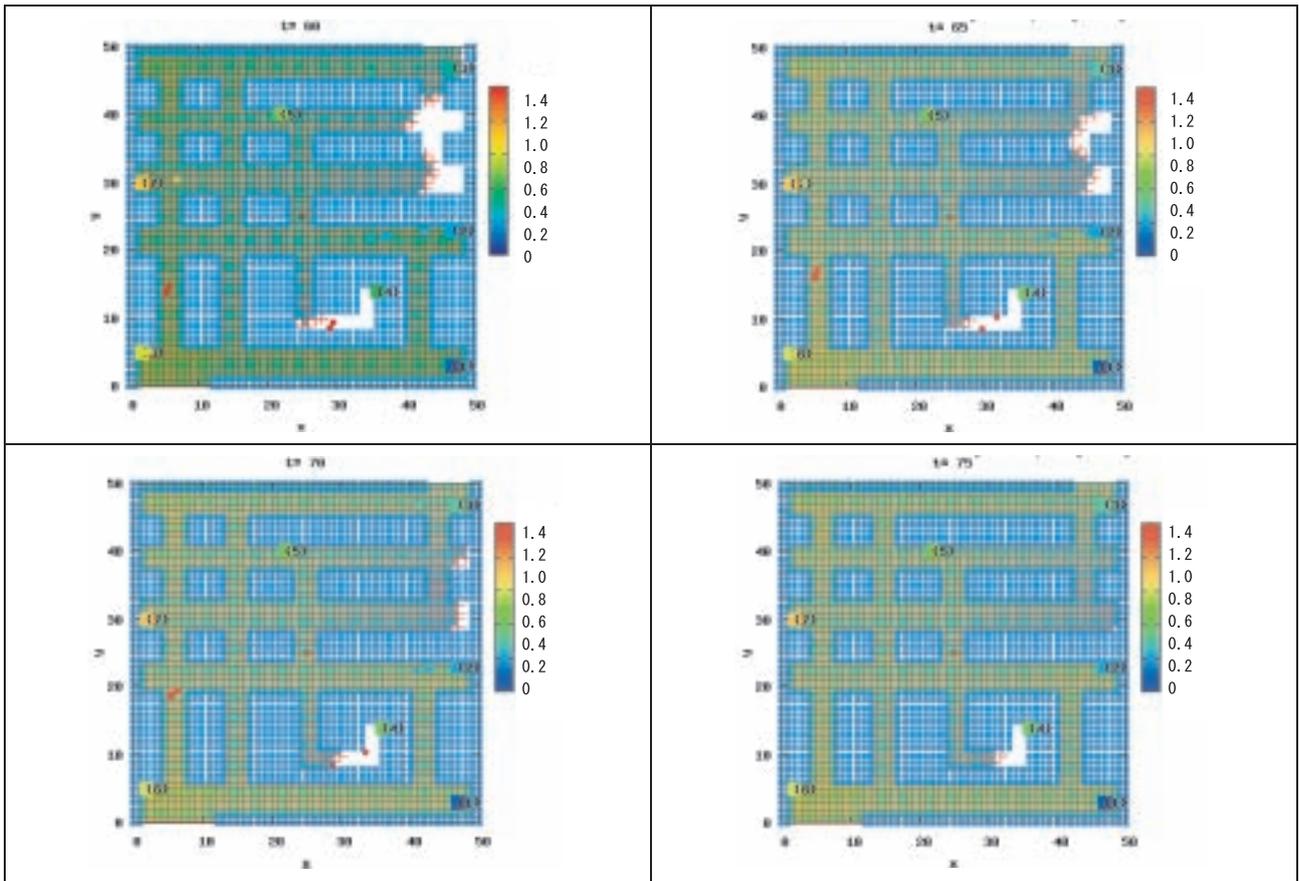


図-17 Run3計算結果

#### (10) Run4の結果の考察

人についていくという要素を加えたところ、Run3では視野に避難所が入っていない人々は逃げ遅れていたが、避難所が分かっている人についていったことによって、逃げ遅れた人、にげ場が分からないひとの数が大幅に減った（図-17）。

### 5. 結論と今後の課題

本研究ではマルチエージェントシステムを用いて避難モデルを構築し、洪水や津波などの災害における避難のあり方について検証できるツールの開発の構築をめざした。これにより避難の解析を行い、現状の避難のあり方の可視化だけでなく、どのような情報が避難に有効でどのようなルートが避難に適しているか、一人で避難できない人をどのようにサポートしていくかということを検証した。

Run1では障害物がなく、水を検知したら、避難所へ向けて避難するという簡単なモデルを考えたが、Run2,3より、障害物を避けながら、避難させることで、周辺の環境の情報を取得する視野の定義が避難シミュレーションでは重要であることがわかった。Run4では避難において、避難所の場所が分からない人は避難できないということが分かった。しかし、避難所の場所が分かっている人へついていくことで、避難所が視野に入り、避難できる人数が大幅に増える結果となった。

本モデルは構築の初期段階であり、さまざまな課題や改良が必要である。課題を次にまとめる。

- Agentの衝突防止

Agentの衝突や重なりは考慮されていない。よって、出入り口付近で発生する押し合いなどは再現できていない。また、Agent同士も重なるために、衝突しても重なってしまう。

- ・ ノードとリンクモデルによる最短距離の算出

モード&リンクモデルという線と点で道や交差点を定義することで, 交通ネットワークを表現できる.  
そのネットワーク上にある任意の2点間から最短距離を導くことができる.

- ・ iRICとMASのパラメーターの一致

iRICの時間ステップと空間の大きさと本モデルの時間ステップと空間の大きさが一致していないので, iRICのものを読取り, それを本モデルに反映させる必要がある.