災害ロボット開発における動的シミュレーション環境の一実現方法

四倉 茂† 岡谷 賢‡ 高橋友一†,‡

名城大学大学院 理工学研究科† 名城大学理工学部 情報工学専攻‡

1. 緒言

災害救助活動を人間の代わりに行うレスキューロボットの開発では，実際の災害現場で実験を行うことは難しい．そのような開発支援のために，三次元モデルで表現されたシミュレーション環境が利用されている．

- Client/Server モデルにより，ロボットの制御を行う．制御するインターフェースとして様々なコントローラを利用することができる．[1] また，シミュレーションロボットと実ロボットを同じプラットフォームで制御することができる．[2]
- Microsoft はビジュアル開発言語 (VPL) を用いてロボット工学プログラムを簡単に作成できる開発環境を提供している．[3]

これらの開発環境は，ロボットの物理モデルだけでなく，与えられる静的環境下でセンサモデルを用いてロボットのシミュレーションを可能にする．現実の災害現場の状況は時々刻々と変化するものであり，センサモデルを用いる場合においてその変化に対応できるロボットの開発が望ましい．

そこで，本稿では，建物の倒壊や火災などといった時間と共に変化する情報を他のシミュレーションから与えることにより，ロボット開発のための動的に変化するシミュレーション環境の実現方法について提案する．ロボットの開発シミュレーションは USARSim (Unified System for Automation and Robot Simulation) を用いて，環境の変化情報を RCRS (RoboCup Rescue Simulation) から与える．[1][4]

2. USARSim

USARSim とは，Unreal Tournament 2004 (UT2004) のゲームエンジン (Unreal Engine 2) を使用することにより，優れた視覚表現と物理モデルを用いてロボット工学特有のプラットフォームや制御システム，センサ，インターフェース，およびモデル化された環境でロボット開発を行うことができる． また，開発者らにより，マップに Static Mesh (UT2004 で扱うオブジェクトの種類) の追加／削除／移動することが可能である．本稿では，この仕組みを利用して USARSim の環境を動的変化させる方法について示す．

A method to create Dynamic Environments for Rescue Robots from Disaster Simulations

† Meijo University, Graduate School of Science & Technology

‡ Meijo University, Department of Information Engineering

3. シミュレーションシステム構成

3.1 システム構成

USARSim は対象となる建物情報を含むマップを管理する Unreal Server と，Robot や人の動きを表現する Unreal Client から成る Client/Server システムである．一方，RCRS は kernel を中心として火災や倒壊などの複数のサブシミュレーションから成る分散シミュレーションシステムである． (Figure.1) USARSim 内でロボットを制御する USARSim Client と USARSim 内の環境を管理する World Controller (WC) を含む USASRim Controller (UC) が本稿で新たに作成した RCRS のサブシミュレータである．

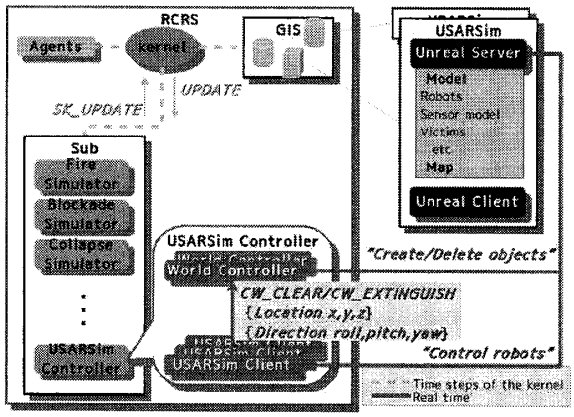

Figure 1: シミュレーションシステムの統合方式 (左: RCRS, 右: USARSim). USARSim Controller 内の World Controller が USARSim 内の世界を変化させる．

3.2 システム間プロトコル

RCRS 内の各サブシミュレータ間が扱う情報の通信システムを利用した USARSim の動的シミュレーション環境の手順について以下に示す．手順 1,2 については RCRS の標準プロトコルであり，手順 3,4 が本稿で追加したプロトコルである．

1. kernel は，各サブシミュレータから倒壊や火災レベルなどといった建物情報を含む差分データを受け取る． (SK_UAPDATE)
2. kernel は受けとった差分データをまとめて全てのサブシミュレータへ送る． (UPDATE)
3. kernel から差分データを受け取った UC 内の WC は，Table.1 に示す建物のプロパティ値をもとに USARSim のマップ内に瓦礫や炎となるオブジェク

トを追加／削除する．

4. kernel は USARSim との時間幅の比だけ停止する*．この間，建物 (USARSim) 内で行動するロボットが瓦礫の除去 (CW_CLEAR) や消火 (CW_EXTINGUISH) を行うと，UC が建物 (RCRS) のプロパティ値とエージェントがどの程度瓦礫に埋まっているかを表す値を変更する．

上記の手順を RCRS の 1 ステップで行う．

Table 1: RCRS が扱う建物情報の一部

状況	プロパティ	値	状態	色
倒壊	brokenness	0	無傷	
		25	部分的に倒壊	灰色
		50	半壊	
		100	全壊	
火災	fieryness	0		
		1		黄色
		2		オレンジ
		3		赤
		5	消火済	水色
		8	全焼	黒

4. 試作システムの評価

試作システムとして，RCRS の建物情報をもとに瓦礫や炎となるオブジェクトの追加することによって動的に変化するシミュレーション環境を実現した．(Figure.2)

4.1 倒壊による環境の変化の例

RCRS の建物の倒壊レベル (brokenness) に応じて，USARSim のマップ内に瓦礫となるブロックを配置することで倒壊の様子を表現した．評価には，USARSim のコントローラの一つである MOAST (Mobility Open Architecture Simulation and Tools) を用いて作成したセンサマップを比較した．[6]

実装結果 (Figure.2) より，作成したセンナマップが RCRS のステップの更新時に変化するのが確認できた．

4.2 火災による環境の変化の例

RCRS の建物の火災レベル (fieryness) に応じて，USARSim のマップ内に炎を配置することで火災を表現した．

5. むすび

本稿では，災害シミュレーション RCRS から建物の変化情報を与えることで，レスキューロボット開発のためのシミュレーション USARSim の動的環境の実現方法について提案した．その結果，動的に USARSim の環境を変化させることで，作成したマップ情報が災害状況によって変化してしまうシステムを構築することができた．このシステムにより，現実の災害時と同じように時々刻々と災害状況が変化する環境下でレスキューロボットの開発が可能になったと考える．

今後は，瓦礫の除去や消火作業により，USARSim 内で変化した情報を RCRS に反映させるシステムを構築する．さらに，炎を感知できるセンサをロボットに用意することで，消火活動を行うロボットに火災感知センサを持たせる予定である．

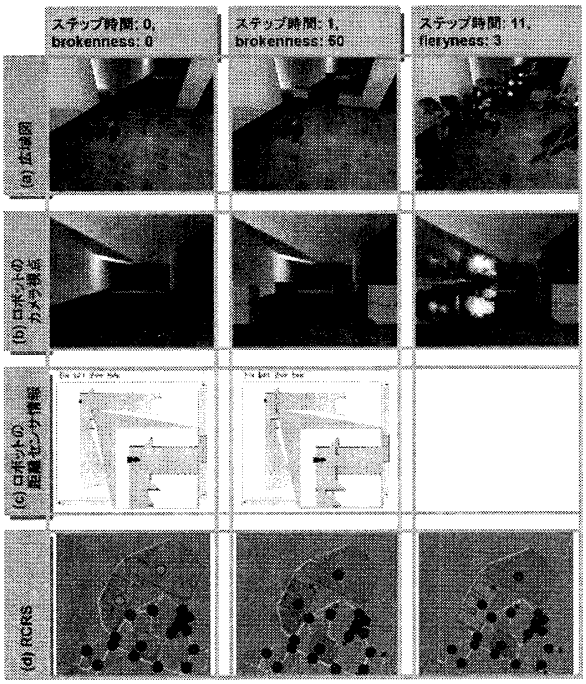

Figure 2: 試作システムの実行例 (横軸は RCRS のステップ時間の変化によるシミュレーション環境の変化を，縦軸は(a)ロボットの上から(b)ロボットのカメラから見た USARSim 内の状況，(c)ロボットのセンサマップ，(d) RCRS のシミュレーション状況，を示す．)．災害が起こった建物 (RCRS) の色が，初期状態→灰色→オレンジ (明るい灰色) という順に変化する．

参考文献

[1] S. Balakirsky, C. Scrapper, D. Carpin, M. Lewis; USARSim: A RoboCup Virtual Urban Search and Rescue Competition, SPIE, Vol. 6561 (2007)

[2] Player Project, http://playerstage.sourceforge.net/

[3] Microsoft Robotics Studio, http://www.microsoft.com/japan/robotics/prodinfo.mspx

[4] 田所 諭, 北野宏明; ロボカップレスキュー大規模災害救助への挑戦, 共立出版 (2000)

[5] S. Yotsukura, K. Sato, T. Tomoichi; A Framework of Simulation System for Rescue Control/Training Center, SICE Annual Conference 2008, 1A18-2

[6] C. Scrapper, S. Balakirsky, E. Messina; MOAST and USARSim – A Combined Framework for the Development and Testing of Autonomous Systems, SPIE, Vol. 6230 (2006)

* USARSim, RCRS のシミュレーション時間の同期が必要．[5]