

R-V 空間相互間で運動状態を伝達する 複合現実遷移モジュールの開発

石田 隼也*¹ 若林 優*¹ 萩尾 恭兵*¹
木村 朝子*¹ 柴田 史久*¹ 田村 秀行*²

Abstract --- As an advanced use of mixed reality technology, we are aiming at systematizing the technology, called “mixed reality transition”, in which the motion state of real objects is transmitted to the virtual world or that of virtual objects are transmitted to the real world. In this paper, we categorize this kind of motion state into patterns and report the design and implementation of “RV-TransitionKit Ver. 1.0” which includes software modules and hardware units for realizing typical RV transitions.

Keywords: mixed reality, RV transition, real space, virtual space

1 はじめに

Virtual Reality (VR)なる言葉が生まれたのは1989年であるが、1990年代前半には現実世界を電子的に強化する Augmented Reality (AR)が登場し、日本の「複合現実感研究プロジェクト」[1]が契機となって Mixed Reality (MR)研究も本格化した。2016年頃から新たな VR ブームが再来し、その余波で AR/MR の応用も活発化している。この実用化が進行しているのは、既にいくつかの教科書や技術解説書[2]が生まれるだけの基幹技術が存在し、ツール類も整備されてきたためと考えられる。

我々は、こうした技術基盤が既に存在していることを前提に、AR/MR技術の実用化のレベルを高め、より魅力的な MR 情報提示を実現し、より幅広い応用分野を開拓することを目指している。本論文で提案するのは、これまで余り試みられて来なかった現実仮想両空間にまたがる物体の動的遷移(これを「複合現実遷移」と呼ぶことにする)を円滑に表現する方法論の体系化である。

これまでの MR システムで、仮想空間側で CG オブジェクトが複雑な動きをする現象を描画している事例は多数あるが、その大半は静的な現実世界を背景に重畳描画しているものが大半である。現実・仮想両空間を同期させ、実物体と CG オブジェクトとの相互作用を実現した事例 [3][4] は存在するが、両空間の接合点で、動物体のシームレスな遷移を実現している訳ではない。

筆者らは、既に「複合現実遷移」実現例として、

「ドミノ倒し」を題材とした MR アトラクション「DOMINO Toppling」を開発した[5]。様々な工夫を凝らしたことでシームレス度は高く、公開した技術デモ展示でも高い評価を得たが、設計・実装は「ドミノ倒し」に特化したものであった。また、研究開発の動機自体が、アート&エンターテインメント分野における MR 表現力の向上を目指したものであった。本研究は、この成功事例のエッセンスを一般化し、産業用、極限作業用、教育用の様々な用途で利用できる技術体系化を指向することにした。

本論文の以下では、RV 両空間で伝達されるべき運動形態の類型化を試み、遷移を実現するソフトウェアモジュールやハードウェアユニットの設計方針を述べ、実装した RV-TransitionKit Ver.1.0 の仕様と利用事例を紹介する。

2 複合現実遷移の意義・対象

2.1 視認する対象空間と遷移方向の表記

現実物体から仮想物体への運動の伝達や遷移を $R \rightarrow V$ のように書くが、体験者が視認しているのは常に現実世界と仮想世界が融合した MR 空間であり、遷移後に視界が全て覆われたフルバーチャルの VR 空間を眺める訳ではない。以下で、R 空間、R 領域のように呼ぶ場合は MR 空間中の実存する物体を視認する部分空間のことを指すものとする。V 空間、V 領域に関しても同様の部分空間である。

R 空間から V 空間への運動状態を伝達する遷移を「 $R \rightarrow V$ 」「 $R2V$ 」「 $RtoV$ 」のように表記し、場合によって使い分ける。「 $V \rightarrow R$ 」「 $V2R$ 」「 $VtoR$ 」はその逆現象であり、「 $R-V$ 」「 RV 」は両方を含むか、特に方向性がなく、両空間を総称する場合に用いる。

*1 立命館大学 情報理工学部

*2 立命館大学 総合科学技術研究機構

特に断らない限り、本研究の「R-V 遷移」では「視覚的 MR」のみを考える。我々は「視覚的 MR」と「聴覚的 MR」の併用技術を開発済みであり [6]、「聴覚的 MR」の導入はさほど困難ではないからである。

2.2 対象とする R-V 遷移現象とその意義

本研究では、以下の R-V 遷移を想定している。

◆Type I (運動伝達の基本形) : R-V 接合部の前後で動物体の運動状態が正しく伝達され、R 物体と V 物体が区別できない状態が理想形である。

◆Type II (非日常的物理条件下への拡張) : V 空間を現実には実現しがたい非日常的物理条件 (極限作業現場や大規模災害被災地等) で模擬実験する。

◆Type III (物理法則に反する超常現象の導入) : V 空間を、地球上で起こり得ない超常現象や空想上の現象を起こすように拡張する。

AR/MR では体験者の R 空間の行為や動作が V 空間に反映される利用法が一般的であるが、本研究では、R 空間での行為がトリガーとなって V 物体に変化が起る場合や、ON/OFF のように記号化された情報伝達は扱わない。あくまで RV 両空間の接合点での物体間での運動状態の伝達を対象として考える。

3 複合現実遷移の機能設計

3.1 RV 間で伝達する情報

我々が目指す R-V 遷移の技術体系化とは、個々の特定の MR 空間に特有の現象に対して専用システムを開発することではなく、類似した R-V 遷移の中から共通項を探し出して類型化することであり、代表的な運動状態の伝達を達し得る処理モジュール群を開発することである。R-V 間の遷移部の設計においては、利用者の使い勝手と調達しやすい部品の利用を最優先し、できるだけ単純化することを目指す。

R 空間から V 空間へ伝達する情報としては、運動状態を直線運動と回転運動に分けた上で、表 1 に示

表 1 RtoV 遷移の項目

Table 1 Categorization of transitions from real to virtual

直線運動	速度
	衝突の強さ
	移動量
回転運動	回転角度
	回転数

表 2 VtoR 遷移の項目

Table 2 Categorization of transitions from virtual to real

直線運動	押圧
	牽引
	随伴運動
回転運動	回転 (角度)
	回転 (数)

す項目を考える。なお、溝やレール上を曲線運動する物体も大いにあり得る形態であるが、そうした自由度 1 の運動は、本質的に直線運動と同じであるので、ここでは直線運動に代表させた。

一方、V 空間から R 空間への力の伝達については、R 空間から V 空間への力の伝達と同様の運動状態に対して、表 2 に示す項目を伝達することとする。

なお、力の伝達においては、力の方向 (ベクトル) についても重要な検討要素となるが、R-V 間の遷移部では、実際に機械的な仕組みを製作する際に方向を考慮することになるため、本研究では、当面、遷移の項目からは除外することにした。

3.2 R-V 間遷移部の実現方法

前節で述べた RV 間の力の伝達方法について、入手が比較的容易な既存のセンサおよびアクチュエータを考慮した上で、その実現方法を検討した。

まずは、RtoV 遷移部について、表 1 に挙げた項目ごとに検討した結果について述べる。R 空間から V 空間に運動の状態を伝えるには、R 空間において、対象の物体をセンシングし、得られた情報を適切に V 空間に伝達する必要がある。

物体の直線運動の速度をセンシングする場合、容易な方法としては以下の 2 つが考えられる。

- ・距離が既知の区間における通過時間を計測
- ・一定の経過時間中に移動距離を計測

このような計測には、表 3 の「速度」の項目に挙げたセンサが利用でき、これらは直線運動における移動量の計測にも使える。

物体の直線運動における衝突の強さをセンシングするには、衝突する地点に衝突を検知可能なセンサを設置する方法がある。これには、表 3 の「衝突の強さ」の項目に挙げたセンサが利用できる。

一方、回転運動については、回転角度、回転数と

表 3 RtoV 遷移のセンサの例
 Table 3 Sensors for real to virtual transition

計測項目	センサ
速度	超音波センサ
	赤外線距離センサ
	フォトリフレクタ
衝突の強さ	圧力センサ
	圧電素子
移動量	超音波センサ
	赤外線距離センサ
	フォトリフレクタ
回転角度	ロータリーエンコーダ
	ホールセンサ
	フォトリフレクタ
回転数	ロータリーエンコーダ
	ホールセンサ
	フォトリフレクタ

もに対象がどの程度回転したかを計測すればよい。これには以下のような方法が考えられる。

- ・直接計測するロータリーエンコーダの利用
- ・回転体に取り付けた磁石をホールセンサで検出
- ・回転体表面の色変化をフォトフレクタで検出

RtoV 遷移部では、対象物体の状況に応じてこれらの中から適切なものを選択する。

次に、VtoR 遷移部について、表 2 に挙げた項目ごとに検討した結果について述べる。V 空間から R 空間へ運動の状態を伝える場合、V 空間側では仮想物体の動きはすべて計算機上で演算された結果となる。したがって、VtoR 遷移の要諦は、V 空間での動きを R 空間へ如何にして繋げるかにある。これには、VtoR 遷移で伝達したい動きに見合ったアクチュエータを利用するしかなく、基本的な選択肢は表 4 に挙げるようなものとなる。

3.3 実現方式を考慮した具体例の選択

本研究では、表 3 および表 4 に掲げたすべてを実現するのではなく、表 5 にあげた項目と対応する各種センサ／アクチュエータを用いて具体的な RV-TransitionKit を実現した。選択基準は、伝達された値による変化が目に見えてわかるかどうか、センサ・アクチュエータによる動作の制約が厳しくないかどうか、という点である。

4 RV-TransitionKit のソフトウェア実装とハードウェア実装

4.1 RV-TransitionKit の設計方針

RV-TransitionKit を実現する上で、要となるのは遷移部を実現する表 3 および表 4 に掲載したようなセンサ／アクチュエータとそれを制御するためのソフトウェアである。ここでは、個々の制御ソフトウェアを「ソフトウェアモジュール」、各センサ／アクチュエータとそれを制御する小型計算機の組み合わせ

表 4 VtoR 遷移のアクチュエータの例
 Table 4 Actuators for virtual to real transition

付与項目	アクチュエータ
押圧	ソレノイド
牽引	ソレノイド
随伴移動	リニアアクチュエータ
回転 (角度)	サーボモータ
回転 (数)	モータ

表 5 RV-TransitionKit の実装対象
 Table 5 Implementation targets of RV-TransitionKit

	項目	センサ・アクチュエータ
RtoV 遷移	衝突の強さ	圧電素子
	回転角度	ロータリーエンコーダ
VtoR 遷移	随伴移動	リニアアクチュエータ

せを「ハードウェアユニット」と呼ぶ。

RV-TransitionKit を様々な状況で利用できる汎用的なものとするためには、ソフトウェアモジュールをハードウェアに依存しない形で抽象化して設計した上で、個々のハードウェアの差異を吸収する仕組みを実現する必要がある。ソフトウェアモジュールには、プログラムを作成する上で必要な汎用的な API (Application Programming Interface) を定義し、個々のハードウェアの差異は、ハードウェアごとに準備するコンフィグレーションファイルから必要な情報を読み込むことで吸収する設計とした。個々のハードウェアユニットに特化したソフトウェアについてもコンフィグレーションファイルの中で指定するような形式をとる。

4.2 ソフトウェアモジュールの設計と実装

RtoV 遷移に関するソフトウェアモジュールとしては、表 3 に示した 5 つの計測項目ごとに別のモジュールとして設計した。それぞれの計測項目と対応するモジュールの名称を表 6 に示す。

物体の運動状態を計測することを考えた場合、プログラマが必要に応じて任意のタイミングで API を呼び出してセンシングの結果を取得するフロー駆動型のスタイルと、何らかの状態変化 (イベント) が発生した際にセンシングの結果が得られるようなイベント駆動型のスタイルが考えられる。各モジュールには、両方のスタイルに対応できるような API を準備することとした。R2V_SenseVelocity モジュールの主な API を表 7 に示す。

VtoR 遷移についても、RtoV 遷移と同様に表 4 に示した動作項目ごとに別のモジュールとして設計した。それぞれの動作項目と対応するモジュールの名称を表 8 に示す。

VtoR 遷移の場合、V 空間での仮想物体の動きは

表 6 RtoV 遷移のソフトウェアモジュール
 Table 6 Software modules for real to virtual transition

計測項目	モジュール名
速度	R2V_SenseVelocity
衝突の強さ	R2V_SenseCollision
移動量	R2V_SenseTranslation
回転角度	R2V_SenseRotationAngle
回転数	R2V_SenseRotationFrequency

表 7 R2V_SenseVelocity モジュールの主な API
 Table 7 APIs of R2V_SenseVelocity module

関数名	処理内容
getVelocity	物体の速度を取得
registerListener	イベントリスナを登録
unregisterListener	イベントリスナを登録解除

計算機上で演算された結果であり、これを R 空間の実物体の動きへと繋げるようなプログラムを作成することになる。したがって、RtoV 遷移とは異なり、必要な API はフロー駆動型のスタイルのみになる。ここでは一例として、V2R_ActuateLinearMove モジュールの主な API を表 9 に示す。VtoR 遷移の場合、アクチュエータの現在の状態を取得するような API と、動きを細かく制御するための API などを準備することになる。

ソフトウェアモジュールには、上で述べた API に加えて、次節で述べるハードウェアユニットと連携するためのメッセージ通信部を実装する。予め、センサやアクチュエータの種類に応じて送受信するメッセージのプロトコルを決めておき、このメッセージを使ってハードウェアユニットをソフトウェアモジュール側から制御する。

ハードウェアユニットの違いについては、それぞれのハードウェアごとに準備したコンフィグレーションファイルから必要な情報を読み込むことで、その差異を吸収する。例えば、VtoR 遷移において、リニアアクチュエータを使って随伴移動動作を伝達する場合、移動量をアクチュエータごとに規定された値で直接的に与えるのではなく、メートル系単位で与えた上で、コンフィグレーションファイルの情報にしたがって値を変換し、アクチュエータを動作させる。一例として、図 1 にリニアアクチュエータのコンフィグレーションファイルの一部を示す。

RV-TransitionKit を利用してリニアアクチュエータを制御するプログラムの構造を図 2 に示す。図中の濃青色領域が RV-TransitionKit に該当する。キットの使用者は、自身で作成するプログラム

表 8 VtoR 遷移のソフトウェアモジュール
Table 8 Software modules for virtual to real transition

付与項目	モジュール名
押圧	V2R_PushRealObject
牽引	V2R_PullRealObject
随伴移動	V2R_ActuateLinearMove
回転 (角度)	V2R_ActuateRotateAngle
回転 (数)	V2R_ActuateRotateFrequency

表 9 V2R_ActuateLinearMove の主な API
Table 9 APIs of V2R_ActuateLinearMove module

関数名	処理内容
getModelName	使用するアクチュエータの名称
getMaxSpeed	最大の移動速度
getPosition	現在の位置
moveAbsolute	絶対位置を指定して移動
moveRelative	相対位置を指定して移動

Develop.cs において、キットが提供する API を利用してリニアアクチュエータを制御する。プログラムを作成する時点で、どのようなハードウェアユニットを利用するかは確定しているため、そのユニットにあわせてコンフィグレーションファイルが読み込まれ、リニアアクチュエータの制御に必要なメッセージが SerialHandler.cs を介して、ハードウェアユニット側のプログラム (Linear.ino) へと送られる。ハードウェア側のプログラムは、メッセージの内容に従って、リニアアクチュエータを駆動させる。

4.3 RtoV ハードウェア実装

RV-TransitionKit におけるハードウェアユニットは、センサ/アクチュエータとそれを制御するための小型計算機から成る。前節で説明したソフトウェアモジュールとの間でメッセージを送受信するために、小型計算機には IEEE 802.11a/b/g/n や Bluetooth, USB, Ethernet など、何らかの通信規格が搭載されていることを前提とする。

小型計算機の候補としては、Raspberry Pi や Arduino などを想定しているが、これらに限定しているわけではない。対象となるセンサやアクチュエータをプログラムによって制御可能で、ソフトウェアモジュールとの間でメッセージを送受信可能であれば、どのような計算機でも利用可能である。

ハードウェアユニットでは、画像処理などの高度な処理を行う必要がない。そのため、今回の実装では、小型計算機として動作が軽くリアルタイム性が確保される Arduino Uno を採用した。

ハードウェアユニットは、個々のセンサやアクチ

```

RSA0N11M9A0K // 型番
100.0 // 最大移動量
0.3 // 最小作動力
1.3 // 最大作動力
5.0 // 最低速度
500.0 // 最大速度
80 // 最低速度設定値
255 // 最大速度設定値
    
```

図 1 コンフィグレーションファイルの例

Fig. 1 Example of configuration file

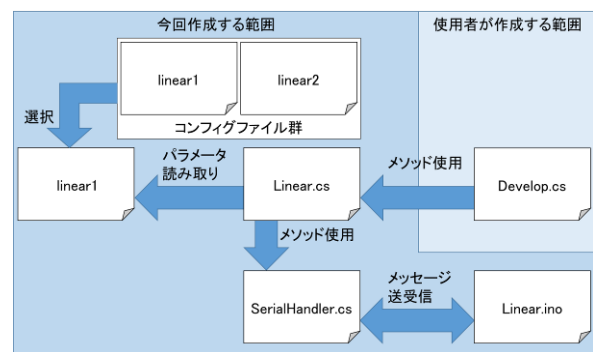


図 2 RV-TransitionKit によるプログラムの構造
Fig. 2 Structure of a program based on RV-TransitionKit

ューータと小型計算機をひとまとめにして扱うこととし、例えば、アクチュエータの種類を変更する場合も、アクチュエータとそれを制御するためのプログラムを内蔵した小型計算機を同時に交換することになる。このような運用をすることによって、ハードウェアユニットを簡便に利用できるようになり、ソフトウェアモジュールとの不整合が発生するような事態を避けることが可能となる。

個々のハードウェアユニットは、制御用の小型計算機の種別や制御対象のセンサ/アクチュエータの種類などに応じた名称が設定されるものとし、上記の設計方針に基づいて RV-KT Ver.1.0 で実現したハードウェアユニットを表 10 に示す。

5 RV-TransitionKit の利用事例

上記のような機能設計を経て実装した RV-KT Ver.1.0 を利用して実現した「複合現実遷移」を（紙幅の都合で）2 例のみを示す。尚、以下で示すのは、各ユニットで正しく動作が伝達されていることを例示するため、少し極端なコンテンツを選んでいる（論文上、静止画でしか表示できないが、運動状態を把

握しやすいビデオ [7] も参照されたい)。

(A) 衝突の強さの伝達による射出物体数の変化

- ・利用ユニット：HU-Coll-ArPie1
- ・R 空間の運動物体：傾斜した溝を伝って落下する球形の物体（ビー玉、鉄球）
- ・意図する運動伝達：球体が固定した板に衝突した際の襲撃の強さをその位置で伝達する
- ・V 空間での現象：板の内部が物理的/化学的に変化し、衝突の強さに依存して多数の小物体が射出され、飛翔する（見やすいようカラーで表示した）
- ・実現結果の解説：衝突の強さは、物体の質量と衝突時の速度に依存するので、球体を置く位置（言わば、斜面での助走距離）と球体の材質を変えて実行した。図 3 には、衝突時までの時間、圧電素子が検出した数値を示したが、その値の違いにより、V 空間で射出する小物体の数が異なっている。

(B) 随伴移動の VtoR 伝達による R 空間での変化

- ・利用ユニット：HU-Lin-ArLa1
- ・R 空間での初期状態：傾斜した通路内に多数の小さな球体が木製の扉で堰き止められている
- ・意図する運動伝達：V 空間のロボットアームが扉

表 10 各種センサ/アクチュエータとハードウェアユニットの構成例
Table 10 Initial set of hardware units including sensors and actuators

HU-Velo-ArUs2		HU-Coll-ArPie1		HU-Ang-ArRe1	
	対応モジュール		対応モジュール		対応モジュール
	R2V_Sense		R2V_Sense		R2V_Sense
	Velocity		Collision		RotationAngle
	ユニット制御器		ユニット制御器		ユニット制御器
Arduino Uno		Arduino Uno		Arduino Uno	
センサ		センサ		センサ	
超音波センサ (SainSmart. HC-SR04) 2 個		圧電素子 (Universal (Changzhou) . FGT-15T-6.0A1W40) 1 個		ロータリーエンコーダ (アルプス電気. EC12E2420801) 1 個	
動作内容		動作内容		動作内容	
上記センサを計測したい運動物体の径路の 2 点に配置し、通過時間差で速度を計算する。		上記センサを衝撃の伝わる物体に取り付けることで衝撃を計測する		上記センサの回転軸を回すことにより回転角度を計測する	
HU-Push-ArSol1		HU-Lin-ArLa1		HU-Rot-ArSm1	
	対応モジュール		対応モジュール		対応モジュール
	V2R_PushReal		V2R_Actuate		V2R_Actuate
	Object		LinearMove		RotateAngle
	ユニット制御器		ユニット制御器		ユニット制御器
Arduino Uno		Arduino Uno		Arduino Uno	
アクチュエータ		アクチュエータ		アクチュエータ	
ソレノイド (Zonhen. ZHO-420S-05A4.5) 1 個		リニアアクチュエータ (アルプス電気. RSA0N11M9A0K) 1 個		サーボモータ (Tower Pro. SG92R) 1 個	
動作内容		動作内容		動作内容	
V 空間からの信号により R 空間物体を動かす		V 空間から指定される移動量だけリニアアクチュエータを利用して R 空間物体を動かす		V 空間から指定される回転角度だけサーボモータを利用して R 空間物体を動かす	

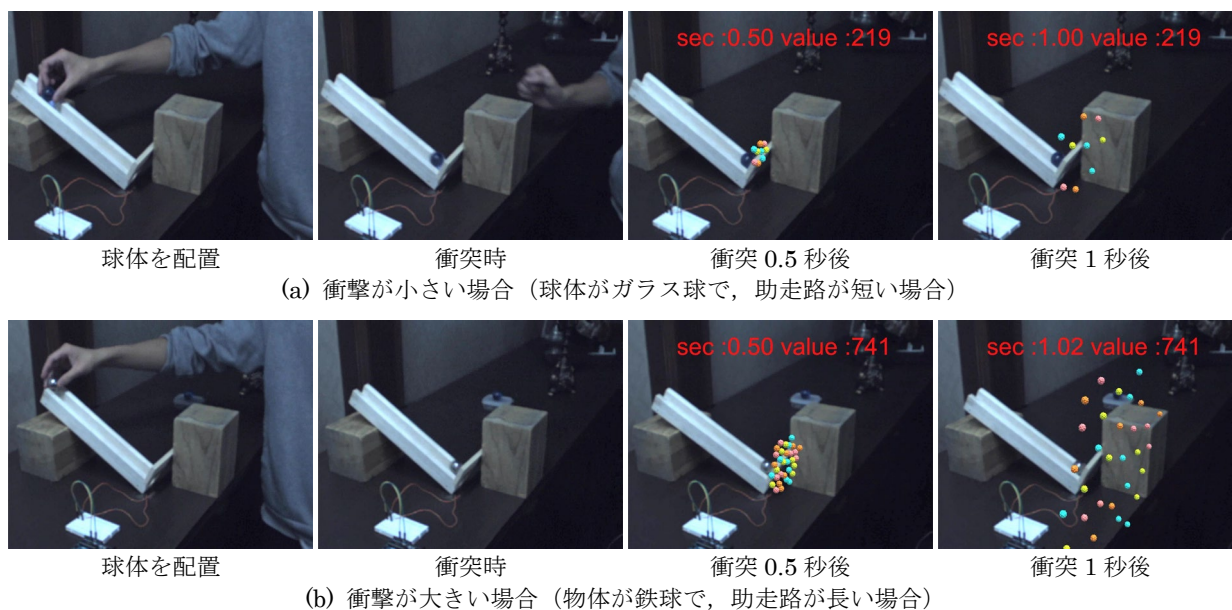


図3 衝突の強さの伝達による射出物体数の変化
Fig.3 Impact strength of real collision affects the number of discharged virtual particles

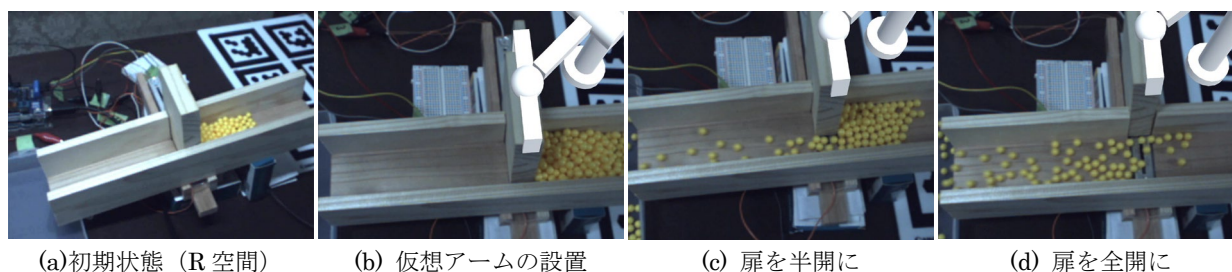


図4 随伴移動の VtoR 伝達による実物扉の開閉の影響
Fig.4 Virtual robot arm slides a real door and changes the real world scene

を掴んで移動させると、R 空間の実物扉がその量だけ移動する。

- ・V 空間での現象：伝達されるのは扉の移動量だけだが、扉（引き戸）の開閉量によって、付属的に小さな球体が転がり落ちる状態が変化する。
- ・実現結果の解説：図4に、扉の開閉結果の2つの状態示している。実際には、扉を引く速度によって、小さな球体が流れ出す状態も変わる。

6 むすび

複合現実感 (MR) 技術の実用化のレベルを高め、より魅力的な MR 型情報提示を実現する方策として、現実・仮想両空間の間での運動物体のシームレスな遷移に着目した。これを円滑に実現できる技術の体系化が、MR 技術の新しい応用分野を開拓することに繋がると考えている。

そのためには、対象毎に個別に専用 MR システムを開発するのではなく、比較的単純でデザインしやすい運動形態の伝達に置き換えて「複合現実遷移」を達成する方式が当該技術の普及に適している。その考えの下に、RV-TransitionKit と名付けたツールキットの Ver.1.0 を試作し、利用例を紹介した。

参考文献

- [1] 田村秀行:複合現実感研究プロジェクト, 計測と制御, Vol. 14, No. 9, pp. 645 - 648 (2002)
- [2] D. Schmalstieg and T. Hollerer: *Augmented Reality: Principles and Practice*, Addison-Wesley Professional (2016)
- [3] 杉本麻樹, 他: 複合現実感技術と小型ロボットを用いたゲーム環境, 情処論, Vol. 48, No. 11, pp. 3490 - 3500 (2007)
- [4] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii: inFORM Dynamic physical affordances and constraints, Proc. 26th Symp. on User Interface Software and Technology (UIST 13), pp. 417 - 426 (2013)
- [5] 平田遼太郎, 石橋朋果, チェカネイ, 森尚平, 池田聖, 柴田史久, 木村朝子, 田村秀行: DOMINO Toppling: 実物体と仮想物体のシームレスな遷移を可能にした MR アトラクション, 日本 VR 学会論文誌, Vol. 21, No. 3, pp. 463 - 472 (2016)
- [6] 比嘉恭太, 西浦敬信, 木村朝子, 柴田史久, 田村秀行: 視覚・聴覚の現実と仮想を融合する 2x2 方式複合現実感システムの実現, 同上, Vol. 13, No. 2, pp. 227 - 237 (2008)
- [7] http://www.mclab.ics.ritsumei.ac.jp/~j_ishida/video.html

© 2020 by the Virtual Reality Society of Japan (VRSJ)