

Pinball Camera

出淵亮一朗*
女子美術大学

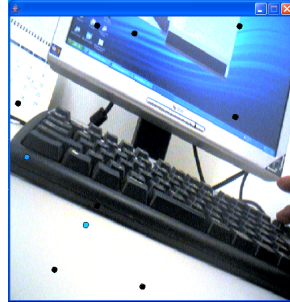


Figure 1: Pinball Camera

Abstract

システム構成として、ウェブカメラ入力画面をリアルタイムで計算される2Dコンピュータグラフィックスと合成して、PCモニター上、あるいは、プロジェクターで表示する、インタラクティブアート作品、「Pinball Camera」(図1)について紹介する。併せて、この作品のメディアアートとしての見地と、実現するためのプログラミング技術に関して簡単に記述する。

Keywords: 複合現実, 画像処理, インタラクティブアート, Processing

1 作品概要

この作品では、観客はウェブカメラを手に持ち、自由に周りの空間や人物を現場で撮影できる。映像はリアルタイムでPCモニターに表示、あるいはプロジェクターで壁面にプロジェクトされる。

このモニター世界(バーチャルワールド)内に飛び回っている複数の小さな黒い物体がある。それは、ピンボールゲームの玉、あるいは気体分子のようにリアルタイムで画面の端で跳ね返り、また、画面に映っている物や人物の黒色の部分でも跳ね返る。観客は現実世界の上に見えない物体が現実と干渉して反応している様子を、インタラクティブに観察することができる。

観客はカメラを持つ側、カメラの被写体となる側いずれにもなることができる。写される側はバーチャルワールドとの関係性を発見してインタラクションを起こしていく。写す側もカメラの移動との関係性に気づきインタラクションを起こしていく。

2 複合現実と拡張現実

1968年にユタ大学のアイバン・サザランド(Ivan Edward Sutherland)によってヘッドマウントディスプレイ(Head Mounted Display HMD 頭部搭載型ディスプレイ)が提案されたものが最初といわれるバーチャルリアリティ(Virtual Reality VR 仮想現実)は、その後、ミックスドリアリティ(Mixed Reality MR 複合現実)、オーグメンティッドリアリティ(Augmented Reality AR 拡張現実)と発展した。

初期のミックスドリアリティを使ったアート作品には、村上泰介による「コンタクト ウォーター」2000-2001 [1] 等がある。

*e-mail: debuchi07005@venus.joshibi.jp

シースルーヘッドマウントディスプレイを使って、現実の視覚にリアルタイム計算によるコンピュータグラフィックスを重ねていたMR技術は、最近ではARと呼ばれるようになったと見受けられる。

ARは、頓智・(トンチドット)によるiPhoneアプリケーション「セカイカメラ」Sekai Camera [2] のように、GPSによる位置情報から、現実世界をデバイスで写した映像にコンピュータグラフィックスを重ねて、「アノテーション」と呼ばれる情報を付加するものと、奈良先端科学技術大学院大学の加藤博一教授が開発したARToolkit (Augmented Reality Toolkit) [3]、博報堂DYメディアパートナーズと慶應義塾大学大学院メディアデザイン研究科デジタルライフプロジェクトによる「テノヒラアド」[4]のように、現実世界をデバイスで写した映像を画像処理を行い、「標識」のパターンを認識させ、その結果に対応するコンピュータグラフィックスを重ねて表示するものがあるようだ。

また、インタラクティブアート作品として、近森基と久納鏡子による「KAGE」[5]のように、現実のオブジェに仮想の影を投影してありえない視覚を作る作品がある。ミックスドリアリティのバリエーションであるといえるだろう。

3 この作品の目指すもの

現実の上に別の世界を見ていた「コンタクト ウォーター」のような作品を実現するミックスドリアリティ技術は、近年ではARに吸収された感がある。しかし現状、ARは現実のものに対する付加情報を視覚に対して与えることを技術の目的としているようだ。

また、「KAGE」のような作品では、MR技術はいわば手品のトリックとして扱われており、ミックスされる非現実の「影」は、作者の意図としては観客に現実の側にあると認知されることである。

この作品では、現実を重ねたもうひとつの世界、あるいは、人の目には見ることでできない現実のもう一つの姿に対して、現実が動的に働きかけることができ、また逆に、そのオーバーラップした世界から現実积极主动にインタラクションする技術の可能性を探ってみようとした。それは、上橋菜穂子のファンタジー小説でよくモチーフとなる、この世界に重なって存在する精霊の飛び交うあちら側の世界を同時に見るようなものである。

4 衝突判定のプログラミング技術

この作品の制作には Processing [6] を使い、Java でプログラミングを行った。重要な点は、カメラ映像をリアルタイムで解析し、現実物体にコンピューターグラフィックス物体を物理シミュレーションで跳ね返らせることであるが、計算速度が観客の印象に大きく関係するはずである。Java は特に計算処理速度が遅くなるので、衝突判定のための画像処理技術に工夫を施した。

カメラから入力された画像内の物体を認識する画像処理として、輪郭線抽出法や運動している領域の抽出法があるが、今回は処理速度に重点を置くために色で認識させる方式を取った。また、最低でも 1 秒 10 フレーム程度以上のリアルタイム計算を行うためには、すべてのピクセルでの処理は難しいので、画面をグリッドで切って、サンプリングピクセルにおけるカラー値を使用した。カラー値は何でもよいが、今回は各 RGB 値があるスレッシュホールド値よりも小さいという判定で黒領域に対して反応を行うとした。

コンピューターグラフィックスによる小円は等速度で運動するが、スクリーンの 4 辺で直線に対する反射を行う。スクリーン内の現実物体は、先に定義したグリッド点上に、ある半径の円があるとし、現実物体をこれらの範囲円で近似させ、それら範囲円に対して飛行する小円が衝突反射するとした。(図 4) 特定の色、例えば黒と判定されたグリッドにある円が衝突に対してアクティブとなる。

飛行する小円の衝突運動は次のステップで計算する。

ある時点で座標 Pold の位置にある小円は速度ベクトルを持ち、次のステップで座標 Pnew に移動するとする。

1. 範囲方形内含有テスト 範囲円を取り囲む方形領域に Pnew が侵入しているか？
2. 範囲円内含有テスト 1) を満たした場合、範囲円内に Pnew が侵入しているか？
3. 2) を満たす場合、運動する小円は範囲円に衝突して方向を変えたとする。

- (a) 半径 R の範囲円と Pold(x1,y1) と Pnew(x2,y2) を結ぶ軌線直線の交点 K(x,y) を求める。(図 2)

適当な係数 a、b を考えて、直線と円の方程式から、

$$y = ax + b \quad (1)$$

$$y1 = ax1 + b \quad (2)$$

$$y2 = ax2 + b \quad (3)$$

$$R = \sqrt{(x - x0)^2 + (y - y0)^2} \quad (4)$$

(1)(2)(3)(4) から求まる二つの交点 K1,K2 の内、Pold に近い方を交点 K として選択する。

- (b) 反射ベクトルを計算する。(図 3) 交点 K から Pold へ向かうベクトル S、円の中心から交点 K に向かう円の法線ベクトル N、交点 K から反射方向に向かうベクトル T を考える。すべて単位ベクトルとする。

適当な係数 c、d を考えて、

$$T = cS + dN \quad (5)$$

入射角 = 反射角より、

$$(T \cdot N) = (S \cdot N) \quad (6)$$

また定義より、

$$|T|^2 = 1 \quad (7)$$

(5)(6)(7) より、

$$T = -S + 2(S \cdot N)N \quad (8)$$

(8) により T が求まるので、反射後の位置は次の位置 Pnew に変更される。但し、v は適当な速度係数。

$$Pnew = K + vT \quad (9)$$

これらを元に運動小物体を定義する class を作り実装している。

5 終わりに

今回の作品は大変シンプルであり、アート作品とするにはもっと味付けすることもできるだろう。飛び回る物体の形状自体やその変化、衝突した位置の色によって、物体のステータスを変えてもよいだろう。たとえば、ハエが飛び回っていて、画面の中の黒色では反射、肌色であれば表面にくっついて止まってしまう等。サウンドもステータスによりいろいろ変化させると面白いだろう。

また架空のボールを現実世界で打ち合うピンポンやテニスのようなゲームも可能であるし、架空に飛び回る虫や鳥、あるいは魚を現実世界で追い込んで捕まえるような作品を作ることでもできそうである。

カメラを持つ側、視線としての観客側へのインタラクションとして、覗き見る現実世界をリアルタイムで変換してしまう作品を作ることもできるだろう。例えば、降りしきる雨粒が現実物体に跳ね返る世界や、妖精や花弁がふわふわと飛び交う幻想的な世界などである。

技術的には Processing を使えばウェブカメラ入力画像へのアクセスが簡単であるが、Java プログラミングは処理速度において問題がある。今後さらに発展させるためには、Windows ならば VC++ というように、使用 OS のコアなアプリケーション開発プログラミングで行うべきであろう。

謝辞

作品のヒントを与えてくれた、椿谷綾子さんに感謝します。

References

- [1] 村上 泰介 コンタクトウォーター 2000-2001 <http://plaza.bunka.go.jp/festival/2001/degital/000039/>
- [2] Sekai Camera <http://support.sekaicamera.com/ja>
- [3] HITLab ARToolkit <http://www.hitl.washington.edu/artoolkit/>
- [4] 慶応義塾大学大学院メディアデザイン研究科 稲見研究室 テノヒラアド <http://inami-lab.kmd.keio.ac.jp/webdesign/>
- [5] 近森 基 久 鏡子 1997 KAGE http://plaza.bunka.go.jp/museum/next_ages/vol4/portfolio.php
- [6] Processing <http://processing.org/>

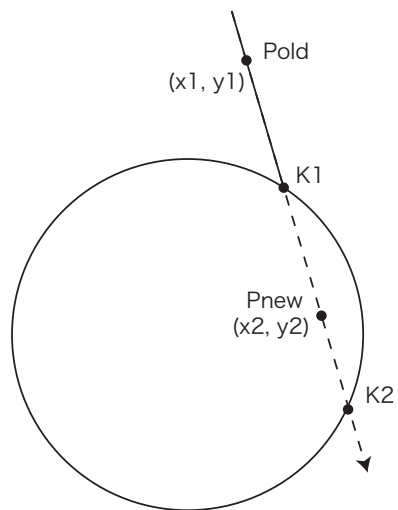


Figure 2: 領域円と軌線の交点の計算

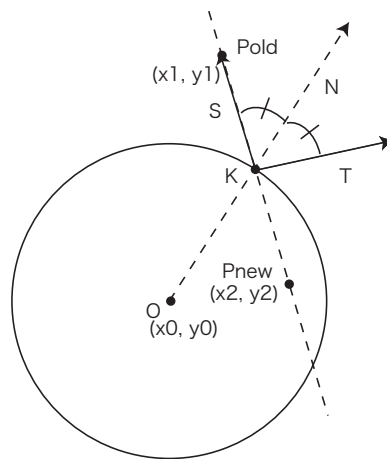


Figure 3: 反射ベクトルの計算

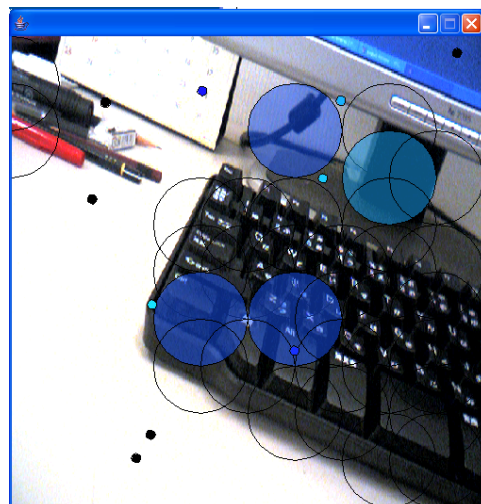


Figure 4: 領域円の表示