

図 3: オブジェクトの出し入れの流れ

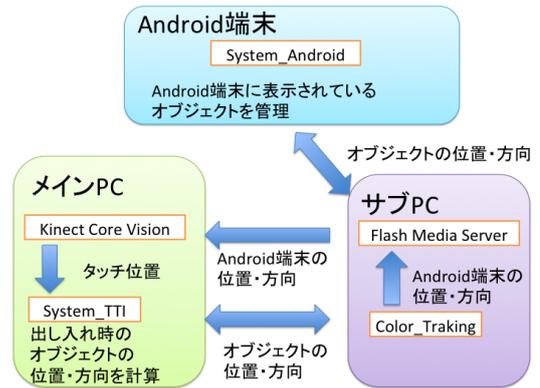


図 4: ソフトウェアの構成

### 2.3 オブジェクトの出し入れ

図 3 に TTI と Android 端末のオブジェクトの出し入れの流れを示す。出し入れ時の Android 端末の方向を  $\theta_1$ ，移動前のオブジェクトの方向を  $\theta_2$ ，移動後のオブジェクトの方向を  $\theta_3$  とする。

以下に，図 3 (A) TTI から Android 端末へオブジェクトを移動する場合の流れを示す。(1) TTI のオブジェクトを Android 端末に向けてタッチ操作でドラッグ&ドロップを行う。(2) オブジェクトが Android 端末に接触したら，Android 端末画面の接触位置にオブジェクトを表示する。そのときのオブジェクトの方向は  $\theta_3 = -\theta_1 + \theta_2$  である。(3) 先で表示されたオブジェクトをタッチすると，TTI に表示されている移動前のオブジェクトが削除される。

以下に，図 3 (B) Android 端末から TTI へオブジェクトを移動する場合の流れを示す。(1) Android 端末のオブジェクトを TTI の画面に向けてタッチ操作でドラッグ&ドロップを行う。(2) オブジェクトが Android 端末の画面端に接触したら，TTI の画面の接触位置にオブジェクトを表示する。そのときのオブジェクトの方向は  $\theta_3 = \theta_1 + \theta_2$  である。(3) 先で表示されたのオブジェクトをタッチすると Android 端末に表示されている移動前のオブジェクトが削除される。

オブジェクトの出し入れ時の Android 端末の位置と方向を考慮することにより，出し入れ時のオブジェクトの位置と方向を維持することで，空間的連続性を確保することが可能になる。そのためユーザは直感的に操作を行うことができる。

### 2.4 ソフトウェアの構成

図 4 に使用するソフトウェアの構成について示す。メイン PC で動作する「System.TTI」では「Kinect Core Vision」からタッチ位置の取得，TTI に表示されているオブジェクトの管理，出し入れ時のオブジェクトの位置・方向の計算を行う。Android 端末で動作

する「System\_Android」では Android 端末に表示されているオブジェクトの管理を行う。サブ PC で動作する「Color\_Tracking」では Web カメラの画像から Android 端末の位置・方向の検出を行う。オブジェクトと Android 端末の位置情報はサブ PC の Flash Media Server を通して「System.TTI」と「System\_Android」で共有される。

### 3 実装

TTI は 80cm (高さ) × 80cm (横) × 120cm (縦) の磨りガラステーブルと鏡 1 枚，プロジェクター 1 台，KINECT1 台，Web カメラ 1 台，マイクスタンドで構築した。KINECT と Web カメラはテーブルから高さ 95cm の位置に設置した。ソフトウェアの開発は ActionScript3.0(FlexSDK4.6)を使用した。実行環境は TTI 用 PC と FMS 用 PC の OS が Windows7 32bit, Adobe Air 3.4 である。スマートフォンは Nexus S の AndroidOS4.1.2, Adobe Air 3.1 である。通信は Adobe Flash Media Development Server 4.5 である。

### 4 おわりに

本稿で提案した手法を用いることで，TTI と携帯端末で相互にオブジェクトを交換する場合に直感的な移動が可能となる。今後は，この手法を取り入れたアプリケーションの開発を行っていきたい。

### 参考文献

[1] Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. "phonetouch: a technique for direct phone interaction on surfaces". UIST '10, pp. 13–16. ACM, 2010.

[2] 高橋良平, 橋田光代, 片寄晴弘. デジタルボードゲーム「我瓶引水」の開発. 全国大会講演論文集, Vol. 72, No. 4, mar 2010.