AR 表示のための Ubiquitous Display システム開発

RT ミドルウェアコンテスト 2008

- manual -

2008 年 11 月 30 日

立命館大学　理工学研究科

李研究室

宮下　智至

e-mail: miyashita[at]ais.ics.ritsumei.ac.jp

目次

1. はじめに

本コンポーネント(群)は，作者らの研究室で開発しているロボット”Ubiquitous Display(以降，UD)”の実験に用いている機能・システムである．UD とは回転プロジェクタを搭載した移動ロボットで，環境(部屋や廊下など)中における壁・床などの平面に対して，投影という手段で情報提供やインタラクションを目指すものである([1], 図 1)．今回のコンポーネントでは，環境中の既知の面に対して画像を投影する際，プロジェクタと面のなす角によって起きる投影画像の幾何歪みを修正する．

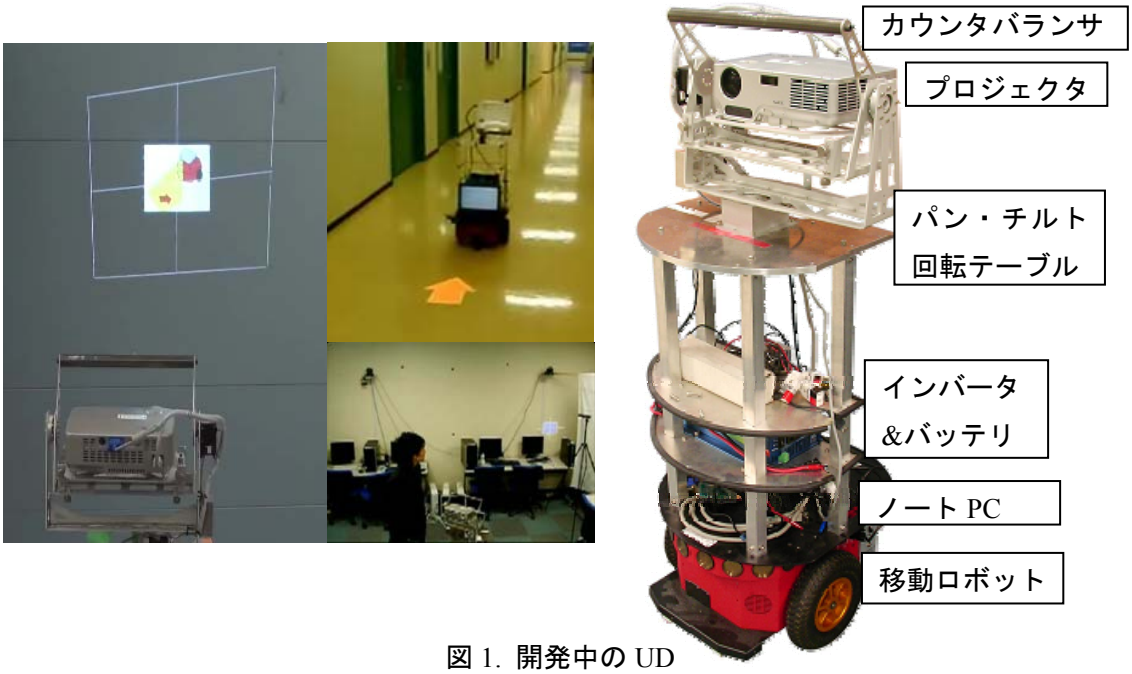

図 1. 開発中の UD

2. 開発環境

本コンポーネント(群)は表 1 のような環境で開発した．

表 1. 開発中の UD

PC	ThinkPad X61
CPU	Intel Core 2 Duo T7500 2.20GHz
OS	Windows XP Pro SP2
RAM	2.0GB
IDE	Visual C++ 2005Express Edition
Library	Boost C++ Libraries 1.34.1 OpenCV 1.0
RTM	OpenRTM-aist0.4.0
Python	python2.4.2
omniORB	omniORB4.0.7

3. ハードウェア

本コンポーネント群のためのハードウェアについて以下に示す.

- 最低限必要なもの
  - 手持ちのプロジェクタ
  - ROBOTIS 社製 Dynamixel シリーズアクチュエータ
  - ROBOTIS 社製 Dynamixel シリーズ対応 USB2Dynamixel
  - ROBOTIS 社製ロボットケーブル
  - プロジェクタと Synamixel アクチュエータで構成されるプロジェクタ回転テーブル
- あるといいもの
  - P3-DX
    - 本コンポーネントは，MOBILE ROBOTS 社製の移動ロボット”P3-DX”の機能である，オドメトリによる自己位置推定を利用している．そのため P3-DX があればより開発環境に近い挙動をするものと思われる.

本コンポーネントを用いる場合は，ハードウェアを以下のように構成する必要がある.

- プロジェクタ回転テーブルの FC-PT 化[2]
- ロボット座標系において，二輪移動ロボットの車軸中心とプロジェクタの回転中心を x-y 平面上で一致させる(図 2)

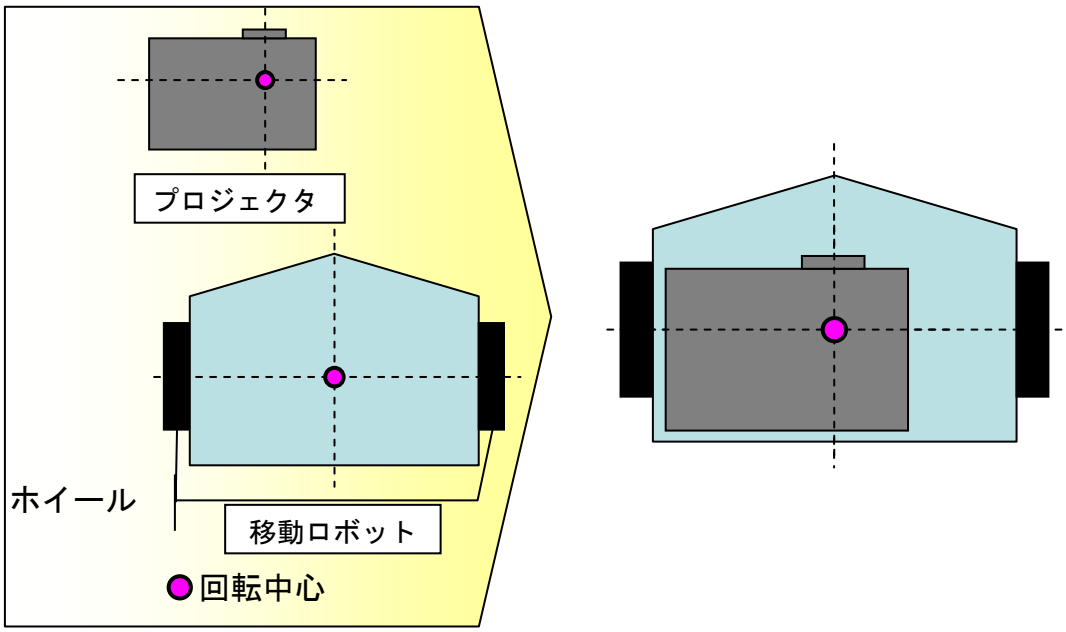

図 2. 移動ロボットとプロジェクタの位置関係

4. コンポーネント構成

本コンポーネントは以下の４点からなる(図 3).

- UdRobotManager コンポーネント

UD のモビリティである P3-DX(MOBILE ROBOTS(R))と TCP によるプロセス間通信を行い，現在位置を取得して OutPort に流す.

- UdBehaviorController コンポーネント

UD の現在位置を InPort から受け取り，プロジェクタが指定した位置に向くよう角度を算出し，OutPort に流す．同時に，その時出力したい画像の 4 端点の座標を算出し，OutPort に流す.

- UdDynamixelManager コンポーネント

プロジェクタの角度を InPort から受け取り，プロジェクタの回転に用いるアクチュエータ RX-64(ROBOTIS)の角度に変換しコントロールする.

- UdImageProseccor コンポーネント

投影する画像は事前に変形しておく. 変形後の 4 端点の座標とプロジェクタの pan 方向の角度を InPort から受けとり，OpenCV の機能で画像を透視投影変換する．変換した画像を全画面表示し，プロジェクタからの出力とする.

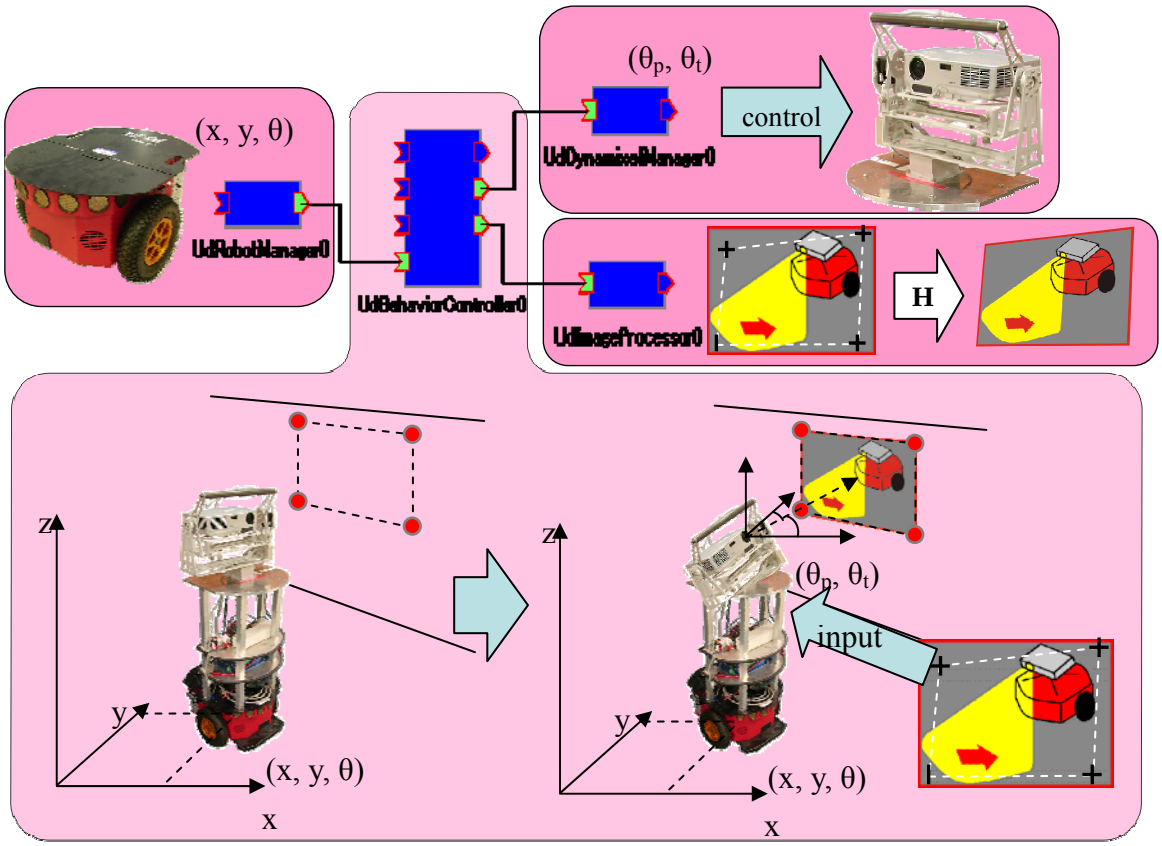

図 3. UD システムのコンポーネント

(ア) UdRobotManager コンポーネント

・ TCP にて，移動ロボット制御用のプロセスと通信を行い，ロボットの姿勢(x, y, θ)を受信する．こうすることで，移動ロボットの換装時など多様なロボットに対応できる．通信の準備のため，IP とポートを指定する(初期設定では localhost:3000)

・ 実際の通信においては，パケットの構成は以下の通りである．

[移動ロボット X 座標]	'A'	[移動ロボット Y 座標]	'B'	[移動ロボット回転角 θ]

・ 実文字列'A'が出現するまでをロボットの X 座標，文字列'B'が出現するまでをロボットの Y 座標，終端文字が出現するまでをロボットの回転角 θ とする

・ 取得したロボットの姿勢(x, y, θ)は，OutPort から出力される．

・ 本コンポーネントで使用されるポートのデータ型と構成は以下の通りである．

・データポート 1　OutPort

・ポートネーム 1　　outputRobotState

・データ型 1　　TimedFloatSeq

(イ) UdBehaviorController コンポーネント

・ (ア)の UdRobotManager コンポーネントから OutPort したロボットの姿勢を InPort し，内部で定義されている投影位置方向へのプロジェクタの角度(pan, tilt)を算出し，OutPort から出力する．

・ プロジェクタにつけられた角度によって，投影した画像は幾何学的に変形してしまうため，プロジェクタに入力する画像を事前に歪ませておくことで投影後の画像の形状を補正する．出力として，歪ませた後の画像の 4 端点の座標を出力する．

・ 以上の処理は，満上らの手法[2]を移動ロボット用に拡張した手法を利用している．

・ 本コンポーネントでは，計算処理ライブラリ"Boost C++"を利用しているので各自インストールのこと．

・ プロジェクタの角度(pan, tilt)と，プロジェクタに入力する画像の 4 端点の座標(x1, y1, x2, y2, x3, y3, x4, y4)がそれぞれに対応する OutPort から出力される．

・ 本コンポーネントで使用されるポートのデータ型と構成は以下の通りである．

・データポート 1　InPort

・ポートネーム 1　　inputRobotState

・データ型 1　　TimedFloatSeq

・データポート 2　OutPort

・ポートネーム 2　　outputDynamixelAngle

・データ型 2　　TimedFloatSeq

・データポート 2　OutPort

・ポートネーム 2　　outputImageCoordinate

・データ型 2　　TimedFloatSeq

(ウ) UdDynamixelManager コンポーネント

・ (イ)の UdBehaviorController コンポーネントから OutPort したプロジェクタの角度(pan, tilt)を InPort し，プロジェクタが搭載されている回転テーブルのアクチュエータへ命令を渡す.

・ 本コンポーネントは ROBOTIS 社の Dynamixel シリーズのみに絞って制御しているため，UdRobotManager のように TCP 通信の設定をする必要は無い．ただしシリアルケーブルを通してアクチュエータに命令を送信するため, ポートの指定のみ行う. コンポーネント起動時にポートの入力を求められるので，環境に合わせたポート名を入力すること.

・ 本コンポーネントで使用されるポートのデータ型と構成は以下の通りである.

・データポート 1　InPort

・ポートネーム 1　　inputDynamixelAngle

・データ型 1　　TimedFloatSeq

(エ) UdImageProseccor コンポーネント

・ (イ)の UdBehaviorController コンポーネントから OutPort した，歪ませた後の画像の 4 端点の座標(x1, y1, x2, y2, x3, y3, x4, y4)を InPort し続ける

・ 本コンポーネントは無償で利用できる画像処理ライブラリ”OpenCV”を利用しているので各自インストールのこと.

・ 投影したい源画像の 4 端点と，それに対応する入力されてきた 4 つの座標から透視投影変換行列を作成し，全画面表示し続ける．全画面表示のために，OpenCVの関数であるcvNamedWindow()の中でCreateWindowのスタイルを改変する.

・ 本コンポーネントで使用されるポートのデータ型と構成は以下の通りである.

・データポート 1　InPort

・ポートネーム 1　　inputImageCoordinate

・データ型 1　　TimedFloatSeq

(オ) 補足

・ 各コンポーネントには，UD 全体でのフィードバック処理を想定して余分にポートが作成されている．しかし今回提供するコンポーネントでは，それらのポートは使用しない.

5. 特徴

- UD というロボットの特徴から，将来的にエンターテインメントやサイネージビジネスに利用でき，そういった市場に RT を広げていく礎となるのではないか.
- コンポーネント群全体としては非常に限られた状況下での機能となるが，以下の 2 コンポーネントは汎用的に利用できると考えられる
  - UdImageProcessor コンポーネント
  - UdDynamixelmanager コンポーネント

6. 使用方法

各コンポーネントを起動し，RTCLink 上で図 3 の様に接続し，Activate する．UD と投影面の初期位置関係は図 4 の通りである.

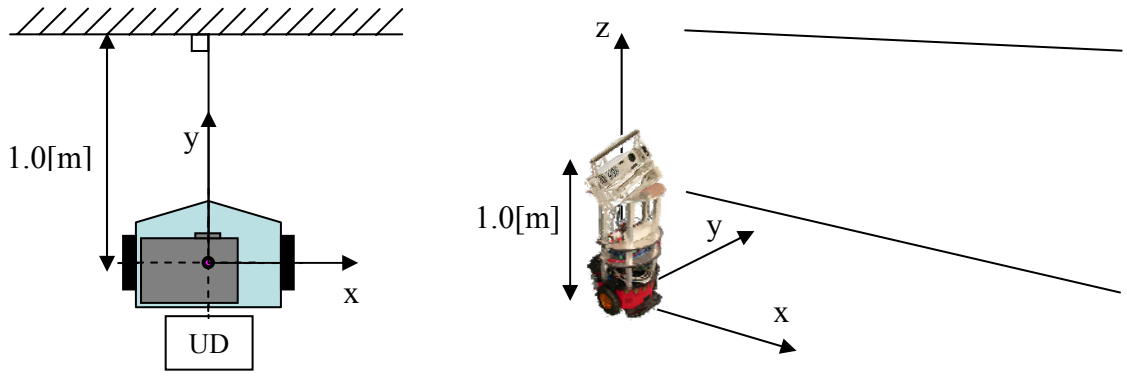

図 4. UD と投影面の位置関係

また，実行時に以下の点は特に注意が必要である.

- UdRobotManager コンポーネント

起動時に TCP 通信でサーバへの接続を行うため，事前に移動ロボット制御用のプロセスをサーバとして起動しておく．サーバが起動されていない場合，あるいは一定時間が経過すると connect error と表示されるので再度起動する必要がある.

7. 終わりに

現時点では(RT ミドルウェア・UD 共に)実験の段階であり，ソースコードの完成度は見るに耐えない．しかしながら，5.で挙げた特徴や，将来的に RT 化されたスペースなどとの連携のためには非常に重要な取り組みで，今後は実質的に稼動させるためシステムを洗練していく.

また，UdBehaviorController コンポーネントには，清水らの提供するプログラム[3]を用いている．ここで感謝の意を述べる.

8. 参考文献

[1] Satoshi Miyashita, Joo-Ho Lee, “UBIQUITOUS DISPLAY FOR HUMAN CENTERED INTERFACE -Fixed Shape Projection and Parameter Optimization-,” Proceedings of the 17th World Congress IFAC Seoul, Korea, July 6-11, 2008. pp.2436-2441

[2] 満上育久，浮田宗伯，木戸出正継，”投影中心固定型パンチルトプロジェクタを用いた複数面投影”，画像の認識・理解シンポジウム，pp.1590-1597，2005.

[3] 清水慶行，太田直哉，金谷健一，“信頼性評価を備えた最適な射影変換の計算プログラム”，情報処理学会研究報告 98-CVIM-111-5 (1998-05), pp.33-40, 1998.5.27