

Android OS とカメラを用いた対象物追跡における処理の高速化 Effective calculation of processing for target tracking using Android OS

高木 佑介[†] 齋藤 翼[†] 金丸 隆志[†]
Yusuke Takagi Tsubasa Saito Takashi Kanamaru

1. 研究背景

オープンソースである Android は、その汎用性の高さから様々な機器に利用可能である。現在では主にスマートフォンの OS として注目を集めているが、我々はこれを組み込みデバイス用の OS として利用する。これにより、スマートフォンの OS としてだけではなく、Android の新たな利用の場が開拓できるのではないかと考えている。

2. 目的

本テーマでは、Android OS の利用の一例を提案する。組み込みデバイスのターゲットとして Texas Instruments 製の BeagleBoard を用いる。モータで上下左右に稼働可能な台座にカメラを取り付け、Android 上で画像処理をし、対象追跡を行う。その際、画像処理の計算手段を改善し、処理速度の向上を図る。将来的な展望として、車載カメラや、防犯カメラへの技術的な転用が可能ではないかと考えている。図 1 に構想図を示す。

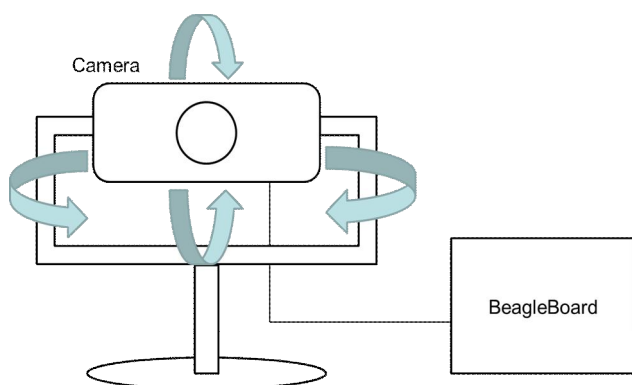


図 1 カメラに用いる台座の構想図

3. 研究内容

3.1 開発環境・機材について

- (1) Android Version … Android 2.2 (Froyo)
- (2) Java 統合開発環境 … eclipse 3.6
- (3) 組み込みデバイス … BeagleBoard Rev.C4
- (4) サーボモータ … 双葉電子工業製 RS304MD-FF (2 個)
- (5) シリアル USB 変換機 … 双葉電子工業製 RSC-U485
- (6) カメラ … Logicool Webcam C210
- (7) 7 インチタッチパネルモニター … Hanwha 製 HM-TL7T
- (8) SD カード … Transcend (8GB)
- (9) LAN アダプター … corega CG-FEUSBTXCW
- (10) USB wifi デバイス … Planex 製 GW-US54Mini2
- (11) USB Bluetooth … Buffalo 製 BSHSBD02

3.2 Android の設定

Android ソースをダウンロードし、本テーマのターゲットである BeagleBoard 用に設定した。その際 sola 氏の android-development-environment という HP を参考とした [1]。その環境をベースに、wifi, USB カメラ, Bluetooth などのデバイスを利用可能するため kernel や Android ソースを変更した。おおまかな変更手順を以下に記す。また、この作業は Linux の端末上で行う。Linux は ubuntu10.04 を用いた。

- (1) kernel 変更…wifi, USB カメラ, Bluetooth の対応
- (2) Android ソース変更 … kernel の変更への対応
- (3) kernel と Android のビルド

3.3 アプリ作成

3.3.1 Windows 用モータ制御アプリの試作

本テーマで利用するモータを制御するアプリケーションの試作を行った。これは Android のアプリケーションではなく、Windows 上で動かすアプリケーションである。モータ制御の基礎を学び、プログラムに対する知識とスキルを高めるために取り組んだ。作成したアプリケーションをもとにしながら、Android でモータを制御するアプリケーションの作成に移った。

3.3.2 Android 用モータ制御アプリ

Cygwin, Android NDK を用いて、JNI を利用し、2 つのサーボモータをそれぞれ 2 つのシークバーで操作できるアプリケーションを作成した。JNI の利用方法等については複数の HP を参考にした。詳細については参考文献 [2][3][4] 参照。このアプリを利用して、モータでカメラの台座を上下左右に自由に動かす。図 2 にアプリのスクリーンショットを示す。シークバーを左右に動かすことで、モータが対応した値分だけ動く。

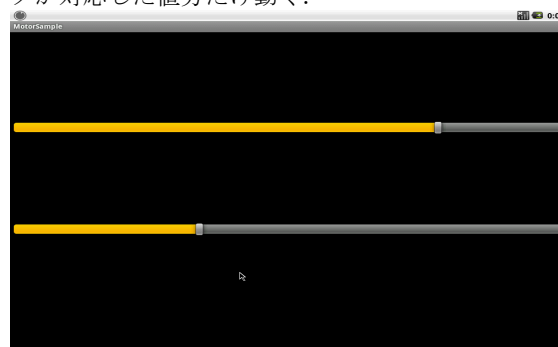


図 2 モータアプリのサンプル画像

3.3.3 カメラアプリ

モータアプリと同様に、カメラアプリを作成した。カメラアプリとモータアプリをそれぞれ独立した 2 つのアプリケーションにするのではなく、1 つのアプリケーション

ンでモータの制御とカメラの画像を処理できるようにした。図 3 にカメラアプリのサンプルを示す。タッチパネルに触れるたびに、通常画像と境界検出が切り替わる。

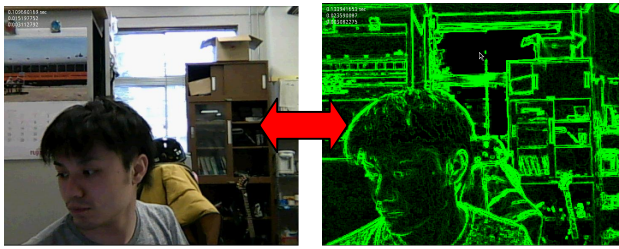


図 3 カメラアプリのサンプル画像

以下ではこのカメラアプリの動作の流れを図 4 に従って解説する。USB カメラの映像は Android ソースの `frameworks/base/camera/libcameraservice/V4L2Camera.cpp` ファイルの関数 `GrabPreviewFrame` にてユーザーに利用可能になっている。この関数は画像のバッファを RGB16 形式で返すよう設定している。ユーザアプリではこの形式を画像処理アプリで利用しやすいよう ARGB32 形式または輝度 Y のみに変換する JNI モジュールを作成し、利用している。そのピクセル値を Bitmap オブジェクトに変換し、最終的にディスプレイに表示している。上記の処理は全て Java のみでも行えるが、JIT コンパイラを搭載した Android 2.2 であっても Java での画像処理は JNI による C モジュールの倍近くの時間がかかった。そのため、画像処理部は全て JNI を用いて C 言語で記述している。なお、画素値を Bitmap オブジェクトに変換する処理を JNI により行うためには、Android 2.2 以上の OS である必要がある。

3.4 台座の製作

モータを 2 つ使った台座を作成した。上下の稼働と、左右の稼働をそれぞれ 1 つずつのモータが担当し、上下左右に動かせるようにした。また、カメラとモータを BeagleBoard に接続し、作成したアプリケーションでカメラの起動とモータの制御を行う仕組みである。

3.5 対象追跡

作成したカメラアプリをもとに対象追跡の機能を付加させる。対象を認識するようにし、認識した対象から、座標を取得する。取得した値をシークバーに返し、モータを動かす。こうすることで、カメラが対象を追跡できるようにする。

3.6 処理の高速化

画像処理の計算手段を改善し、Android 上での画像処理速度の向上を図る。BeagleBoard は OMAP3 プラットフォームを採用しており、NEON と呼ばれる SIMD 拡張を利用できる。その利用についても検討する。

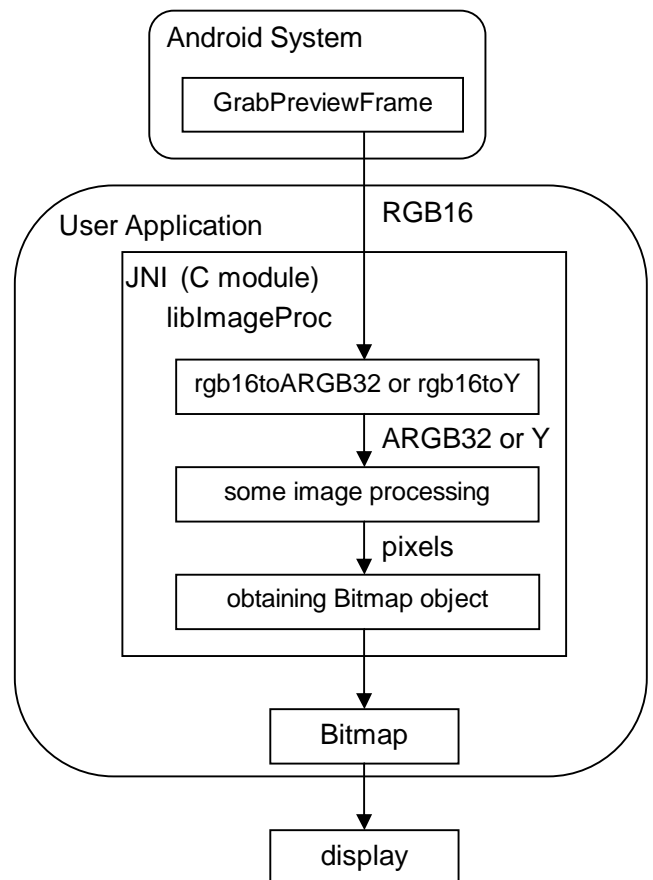


図 4 本システムにおける画像処理の流れ

4. 成果

Android OS を用いた対象追跡のカメラの作成に成功した。また、画像処理速度の向上も見られた。

参考文献

- [1] %android-development-environment+
<http://code.google.com/p/android-development-environment/>
- [2] %アンドロイド開発環境の構築 (その 5) NDK のインストールと設定+
http://www.usefullcode.net/2010/12/android_sdk_inst05.html
- [3] %Android JNI プロジェクトをゼロから作る+
http://www.usefullcode.net/2010/12/android_jni.html
- [4] %Android NDK のサンプルプロジェクトをビルド/実行する+
http://www.usefullcode.net/2010/12/android_ndk_hello_jni.html