

# スマートフォンならではの グラフ表示

鈴木 晃 (神戸大学)

2010年3月23日

数学ソフトウェアとフリードキュメント10 @ 慶應大学日吉キャンパス



スマートフォンって？

# モバイルデバイス

- 1980年代までは電子手帳、ポケコン
- 1990年代からPDAと呼ばれるジャンルの製品が登場
- Newton, HP200LX, WindowsCE, PSION, Palmなど
- 2002年 Linux Zaurus登場

# スマートフォン

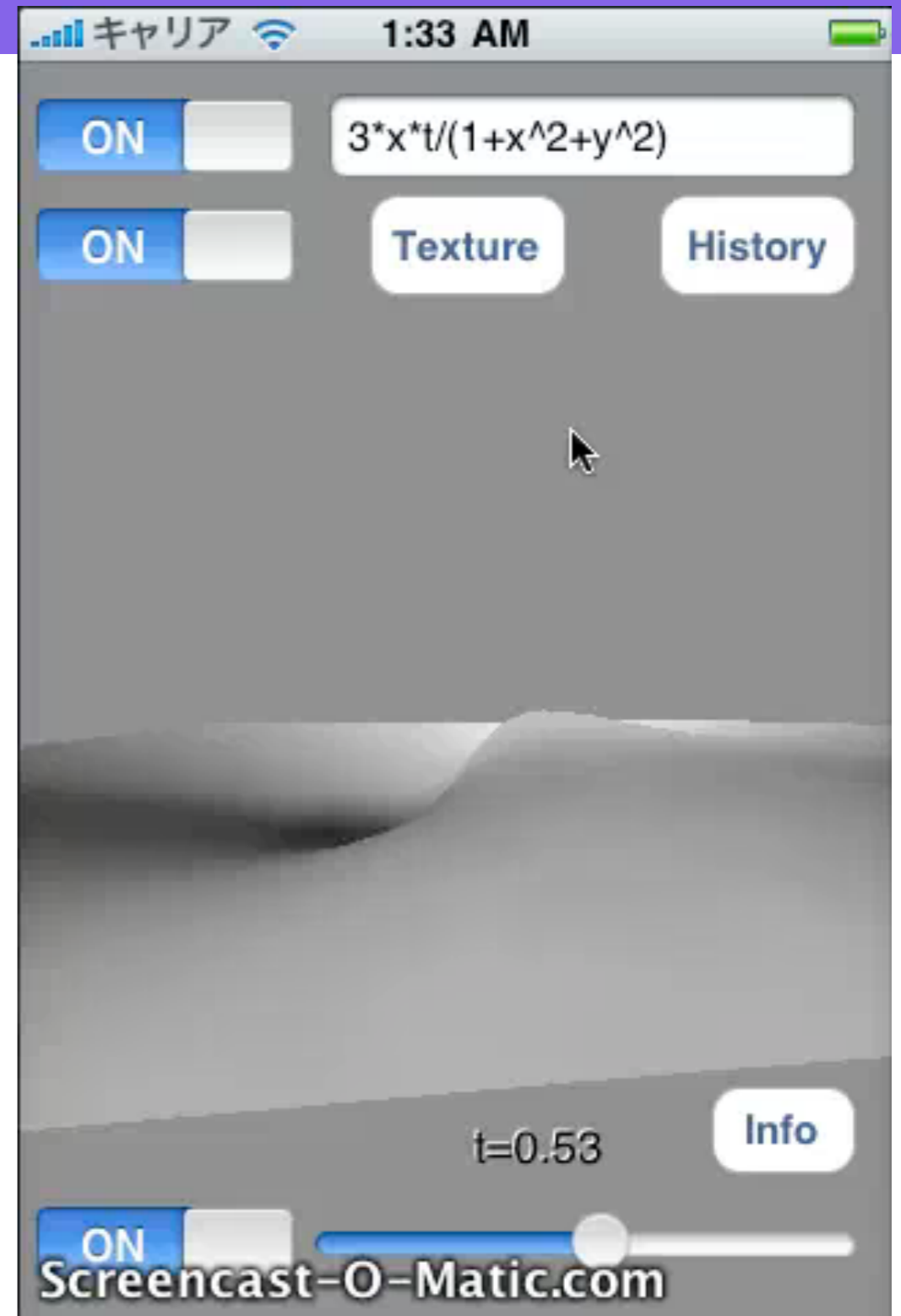
- 通信機能内蔵の「スマートフォン」が登場
  - Windows Mobile (Microsoft)
  - Symbian S60 (Nokia)
  - BlackBerry (RIM)
  - iPhone OS X (Apple)
  - Android (OHA / Google)

# iPhoneとAndroid

- iPhone以前は通信量が控え目
  - 例えばBlackBerryは極力パケットを使わずに高機能を実現する仕組みを備える
- iPhone以降はパケットを湯水のように使う
  - クラウドサービスを積極的に利用
  - Androidも基本的に同じ方向

# どんなものを作った？

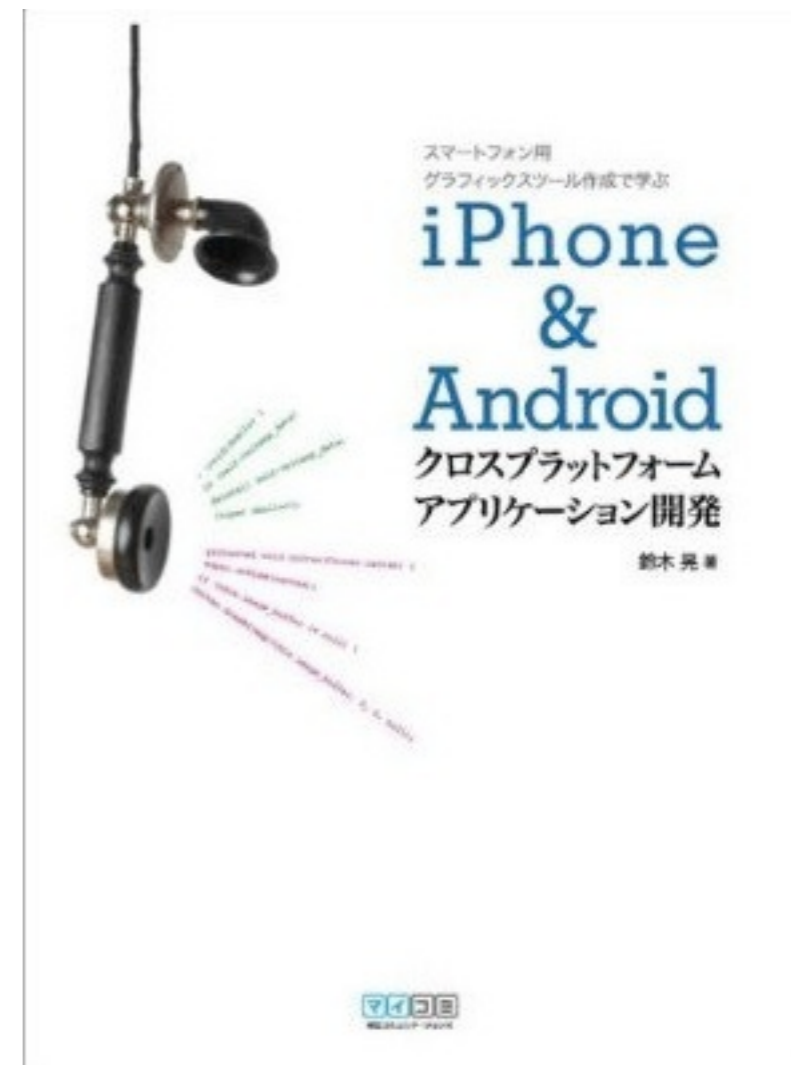
- $z=f(x,y,t)$ を入力するとグラフを表示する
- 「3D Graph Surface View」



# 対象プラットフォーム

- iPhone及びiPod touch

最近こんな本を出したこともあって、Androidも対象とするつもりでしたが、間に合いませんでした…



# iPhoneプラットフォーム



# iPhoneプラットフォーム

- iPhone 3G/iPhone 3GS
  - 電話機能, WCDMA, GPS, カメラ
- iPod touch 第1世代～第3世代
  - GPSやカメラもないが、それ以外はiPhoneと同じ

# iPhoneプラットフォーム

- 115.5mm×62.1mm×12.3mm / 115~133g
- 480×320 3.5inch LCD
- ARM11 (VFPにより8個float同時計算)
- 412MHz / 533MHz / 600MHz
- メモリ 128MB / 256MB
- ストレージ 4GB~64GB

# iPhoneプラットフォーム

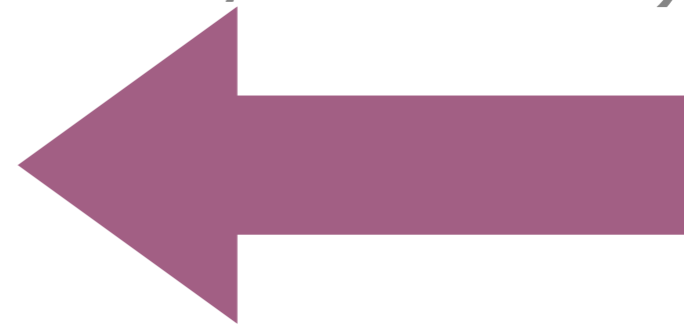
- 無線LAN (802.11b/g)
- 加速度センサー
- マルチタッチ
- Open GL ES 1.1 / 2.0
- Objective-C 2.0 (但しGCは除く)
- OSを最新版にすれば古い機種でも最新機能を

# iPhone OS

- Appleは「OS X」と呼称。BSDベース
  - マルチタスク、マルチユーザ
- Core Service (セキュリティ, SQLite, XML)
- Media (CoreGraphics, OpenGL)
- Cocoa Touch (iPhone OS独自の機能)
  - 各種ウィジェットやマルチタッチなど

# AppStore

- iPhoneやiPod touchのアプリケーションを一般に配布できる唯一の方法
- 学内、社内に配布、もしくはβテスト用の配布手段は別に用意されている
- Appleによる審査が必要。(ポルノ/暴力禁止)
- プログラミング言語の禁止
- 有料登録(99USD/年)が必要
- 課金はAppleが代行



# 開發環境

# 開発環境

- iPhone SDKが無料配布されている
  - ここではjailbreaking環境での開発には触れません
- 2008年6月公開。
- 開発にはMac OS X 10.5 (Leopard)以降を搭載したintel Macが必要
- IDE, iPhoneシミュレーターなどを含む
- ドキュメントは豊富
- Objective-C

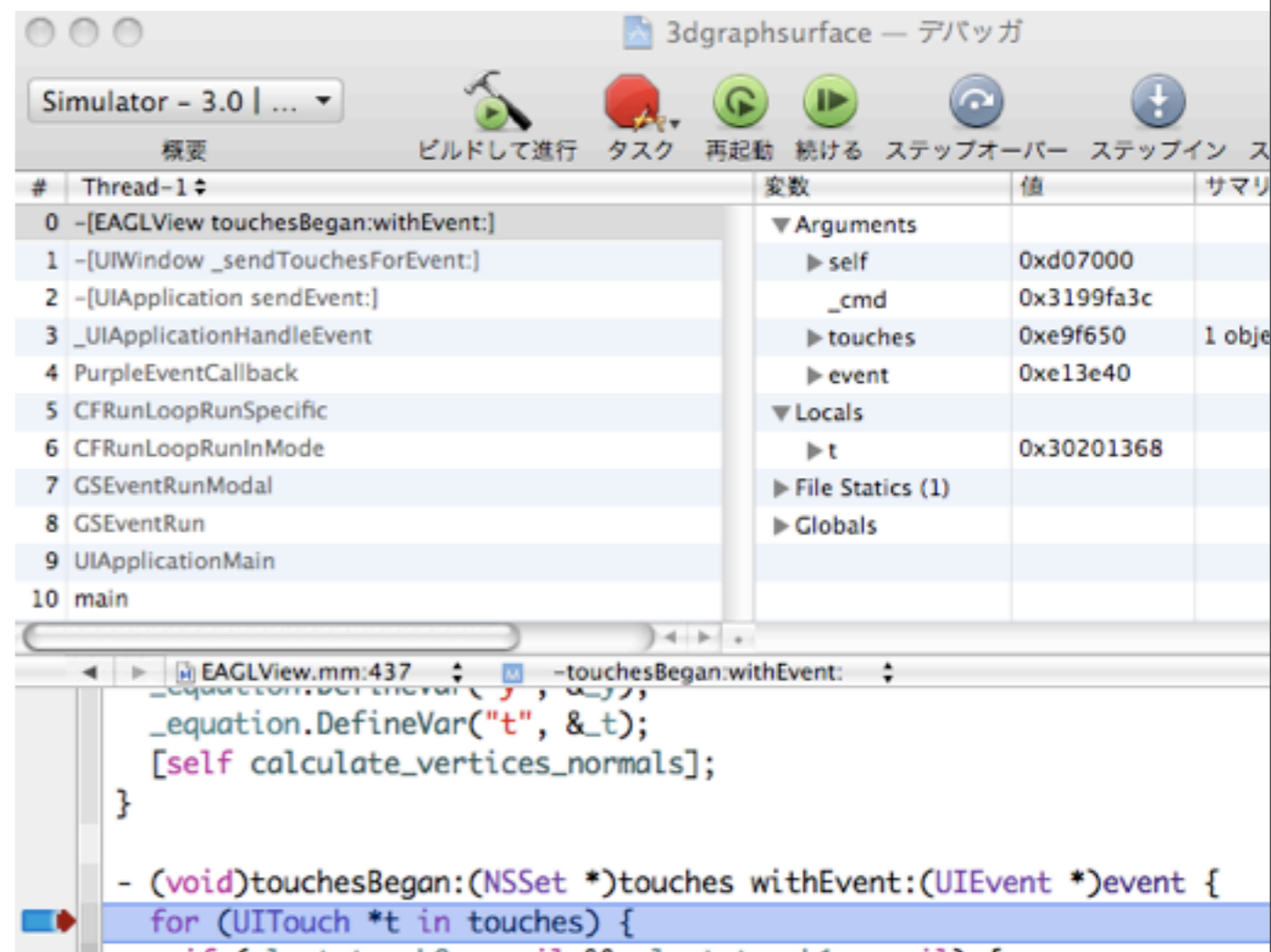
# Objective-C++

- Objective-C 2.0 (GCなし)
- retain, release, autorelease
- 実はC++を使えます。STLも使えます。
  - AndroidのNDKではC++のサポートは限定的 (例外処理/STLPort)
- エンジンにはC++、API呼び出しだけ Objective-Cという使い方も大丈夫



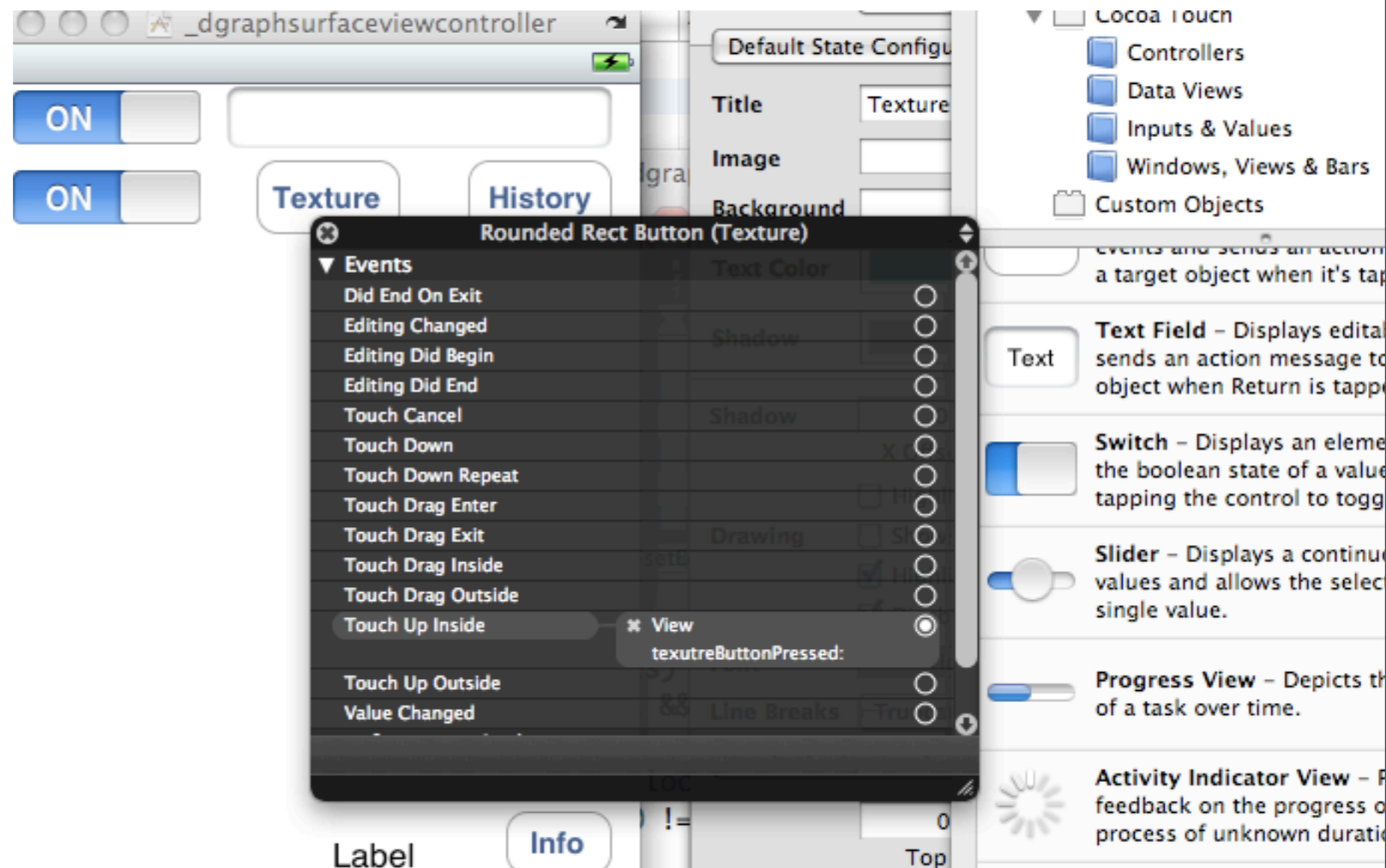
# Xcode

- Objective-C++のコードを書き、ビルドし、デバッガやドキュメントとの接続
- GCCによるアプリケーションのビルド
- GDBのGUIラッパー



# Interface Builder

- 画面の設計を行うツール
- Xcodeとの連携が便利



# その他のツール

- iPhone Simulator
  - エミュレータではないので、実機と動作が異なる場合も。(x86のコードで動く)
  - 加速度センサー未対応
  - マルチタッチは指二本まで
- Instrument
  - リアルタイムにメモリリークの状況の監視などが可能

# 3D Graph Surface Viewで 使われている技術

# OpenGL ES

- OpenGLの機能縮小版
- 3DCGとして極めて標準的な設計
- 視野の設定、視点の設定、 $4 \times 4$ 行列による物体の移動・変形、頂点座標の指定、ポリゴンの指定、法線ベクトルの指定、光源、マテリアルの指定など

```
glEnableClientState(GL_VERTEX_ARRAY);
glEnableClientState(GL_NORMAL_ARRAY);
// glEnableClientState(GL_INDEX_ARRAY);
glVertexPointer(3, GL_FLOAT, 0, _surface_vertices);
glNormalPointer(GL_FLOAT, 0, _surface_normals);
```

- インターフェイスはC言語
  - 頂点座標の指定にfloat[]を使うなど

# マルチタッチ

- 二本以上の指を同時に認識する機能
- iPhone/iPod touchでは5本までは認識
- touchesBegan, touchesMoved, touchesEnded, touchesCancelled
- 各メソッドにはUITouch\*のNSSetが渡され、どの指がどう移動したかの追跡は、ポインタとしての比較で可能

# 加速度センサー

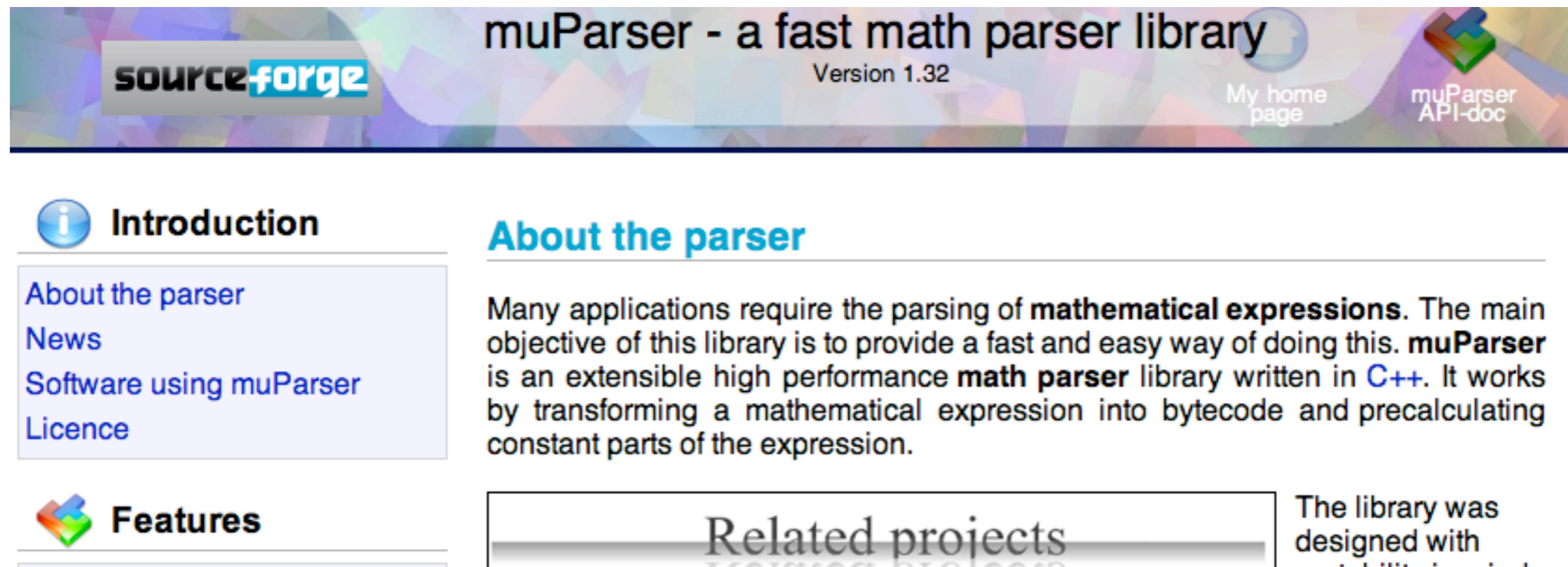
- 主に重力加速度を検知することで、iPhoneがどちらに向いているかを判断
- 衝撃などの検知も当然可能
- 下のコードでdelegate内で、XYZ各軸に対してどれだけの加速度がかかっているかの検知可能

```
//Configure and start accelerometer  
[[UIAccelerometer sharedAccelerometer] setUpdateInterval:(1.0 /  
    kAccelerometerFrequency)];  
[[UIAccelerometer sharedAccelerometer] setDelegate:self];
```

# 数式の解釈

- 以前のバージョンでは自前で実装していた
- 新バージョンではmuParserを利用  
<http://muparser.sourceforge.net/>

Copyright (c) 2010 Ingo Berg Copyright (c) 2010 Ingo Berg



The screenshot shows the top portion of the muParser website. At the top left is the SourceForge logo. The main title is "muParser - a fast math parser library" with "Version 1.32" below it. To the right are links for "My home page" and "muParser API-doc". Below the header is a navigation menu with "Introduction" selected. Under "Introduction" are links for "About the parser", "News", "Software using muParser", and "Licence". To the right of the menu is the "About the parser" section, which contains a paragraph of text describing the library's purpose and features. Below the menu is a "Features" section, and below the "About the parser" section is a "Related projects" section.

sourceforge

muParser - a fast math parser library  
Version 1.32

My home page  
muParser API-doc

**i** Introduction

About the parser  
News  
Software using muParser  
Licence

**About the parser**

Many applications require the parsing of **mathematical expressions**. The main objective of this library is to provide a fast and easy way of doing this. **muParser** is an extensible high performance **math parser** library written in **C++**. It works by transforming a mathematical expression into bytecode and precalculating constant parts of the expression.

**Features**

**Related projects**

The library was designed with portability in mind.



# muParser使うの簡単

- C++言語で書かれている  
(CとC#のラッパーも  
用意されている)

```
- (void)setEquation:(NSString*)eqn {  
  _equation.SetExpr([eqn UTF8String]);  
  _equation.DefineVar("x", &_x);  
  _equation.DefineVar("y", &_y);  
  _equation.DefineVar("t", &_t);  
  [self calculate_vertices_normals];  
}
```

文字列を  
std::stringで設定

数式解釈時に"x"へ  
\_xの値を代入

# muParser便利

- ビルトイン関数:

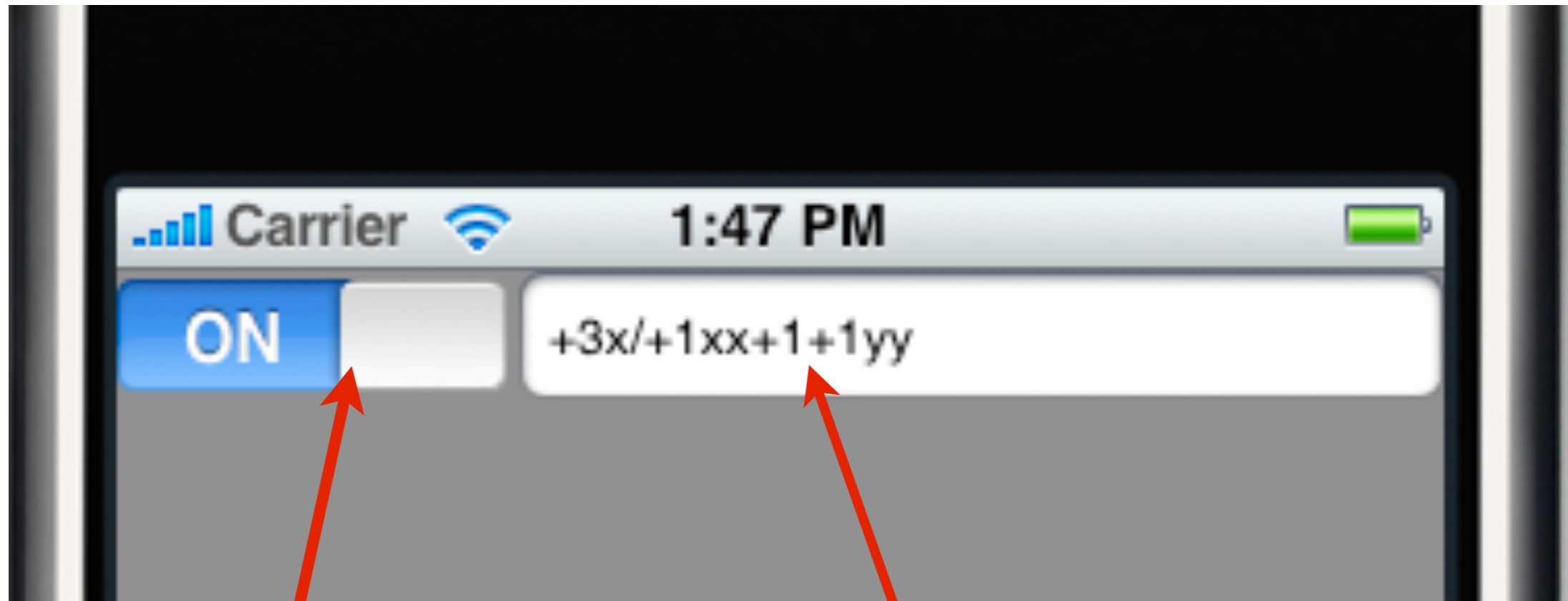
sin, cos, tan, asin, acos, atan, sinh, cosh,  
tanh, asinh, acosh, atanh, log2, log10,  
log, exp, sqrt, sign, rint, abs, if, min, max  
sum, avg

- ビルトイン二項演算子:

=, and, or, xor, <=, >=, !=, ==, >, <,  
+, -, \*, /, ^

# 3D Graph Surface Viewの 簡単な説明

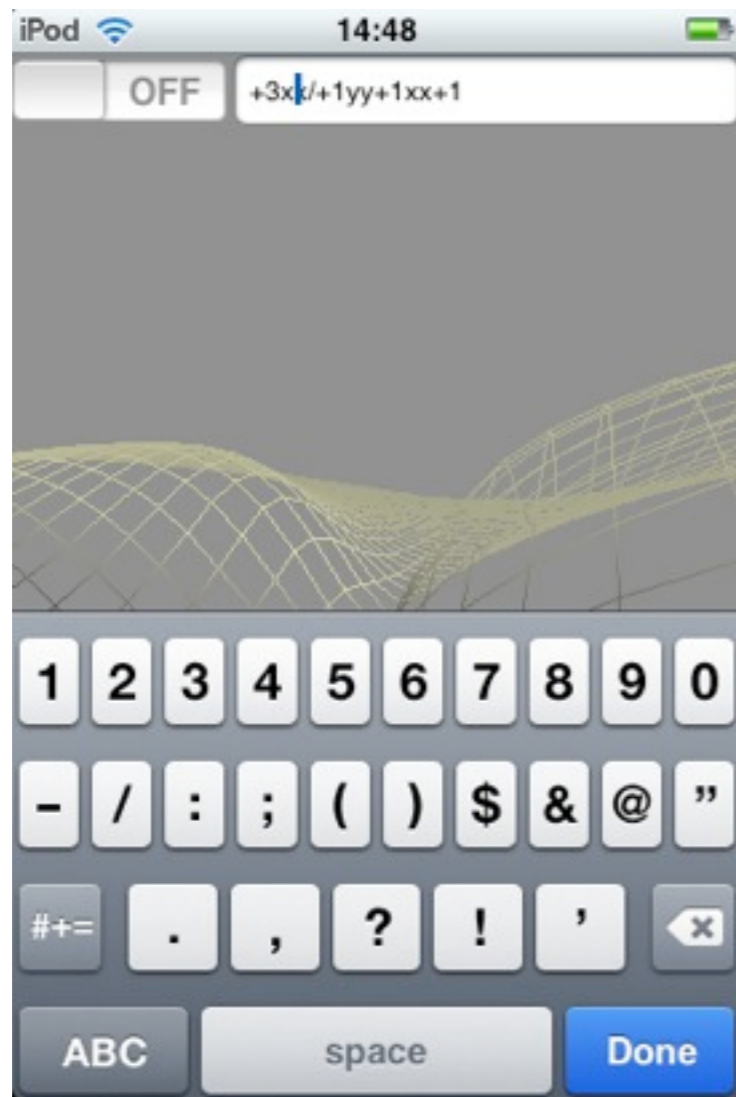
# 基本的な使い方



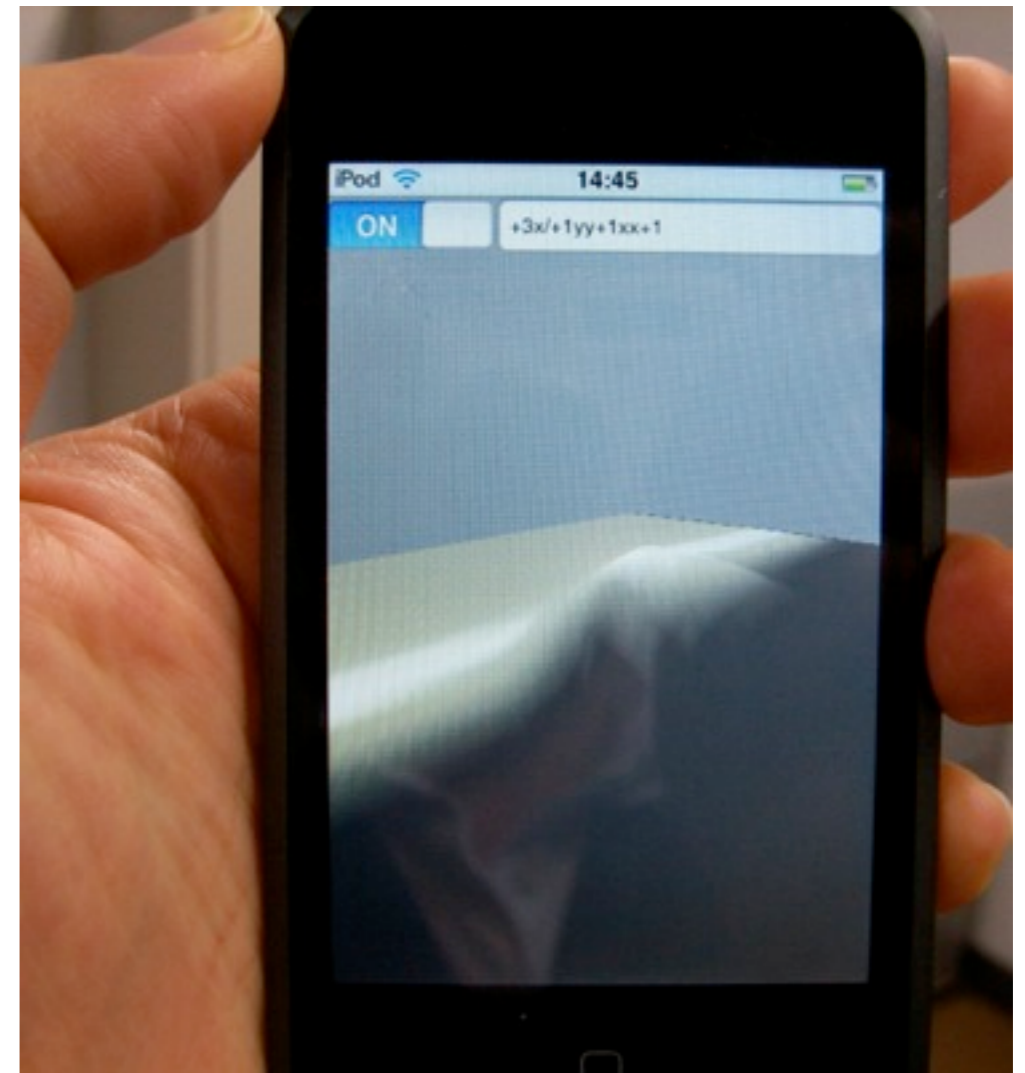
ここに表面を描くか  
どうかを指定

ここに(多項式)/(多項式)を入力

# 3D Graph Surface View



数式編集集中/  
ワイヤーフレーム表示

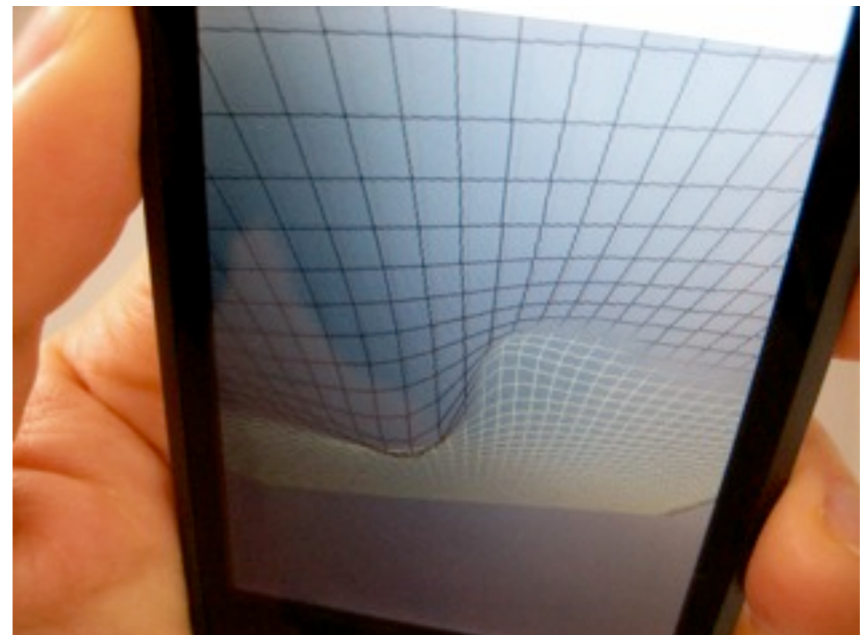
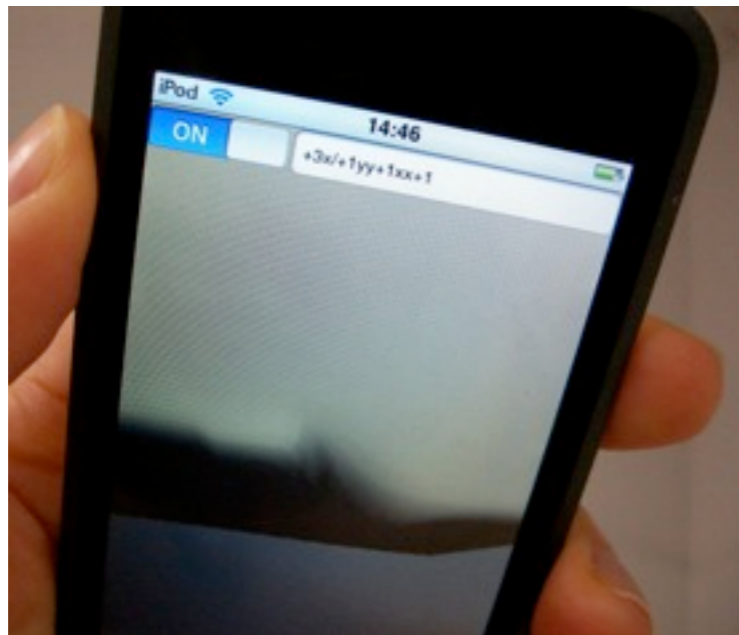


実機(iPod touch)  
での画面

# 3D Graph Surface View



加速度センサーの利用により  
上から見下ろしたり  
下から覗いたり  
直感的に可能



# 3D Graph Surface View

- デモ

さいごに



# やるべき/やりたいこと

- Androidでの実装
  - 無視できない
    - 国内でもSBM, DoCoMo, auが端末発表を予定
- 軸や目盛の表示
- $f(x, y, z, t)=0$  の形のグラフの描画
  - 複数のグラフの交線のハイライトも