



Дополненная реальность через веб-камеру

21.09.2010

Михаил Кокорев (shade@byte-force.com). BYTE-force

Дополненная реальность: что это?

- Дополненная реальность (англ. augmented reality, AR), — термин, относящийся ко всем проектам, направленным на дополнение реальности любыми виртуальными элементами.
- Другое определение: Дополненная реальность - добавление к поступающим из реального мира ощущениям мнимых объектов, обычно вспомогательно - информативного свойства.
- Не путать с virtual reality: Их коренное различие в том, что дополненная вносит отдельные искусственные элементы в восприятие реального мира, а виртуальная конструирует новый искусственный мир.

Зачем это может быть нужно?

- Автомобили – ИЛС
- Авиация – нашлемное целеуказание
- Спортивные трансляции – отрисовка траекторий

Ближе к нашей теме

- Портативные устройства: волшебная линза
- Web-camera – волшебное зеркало

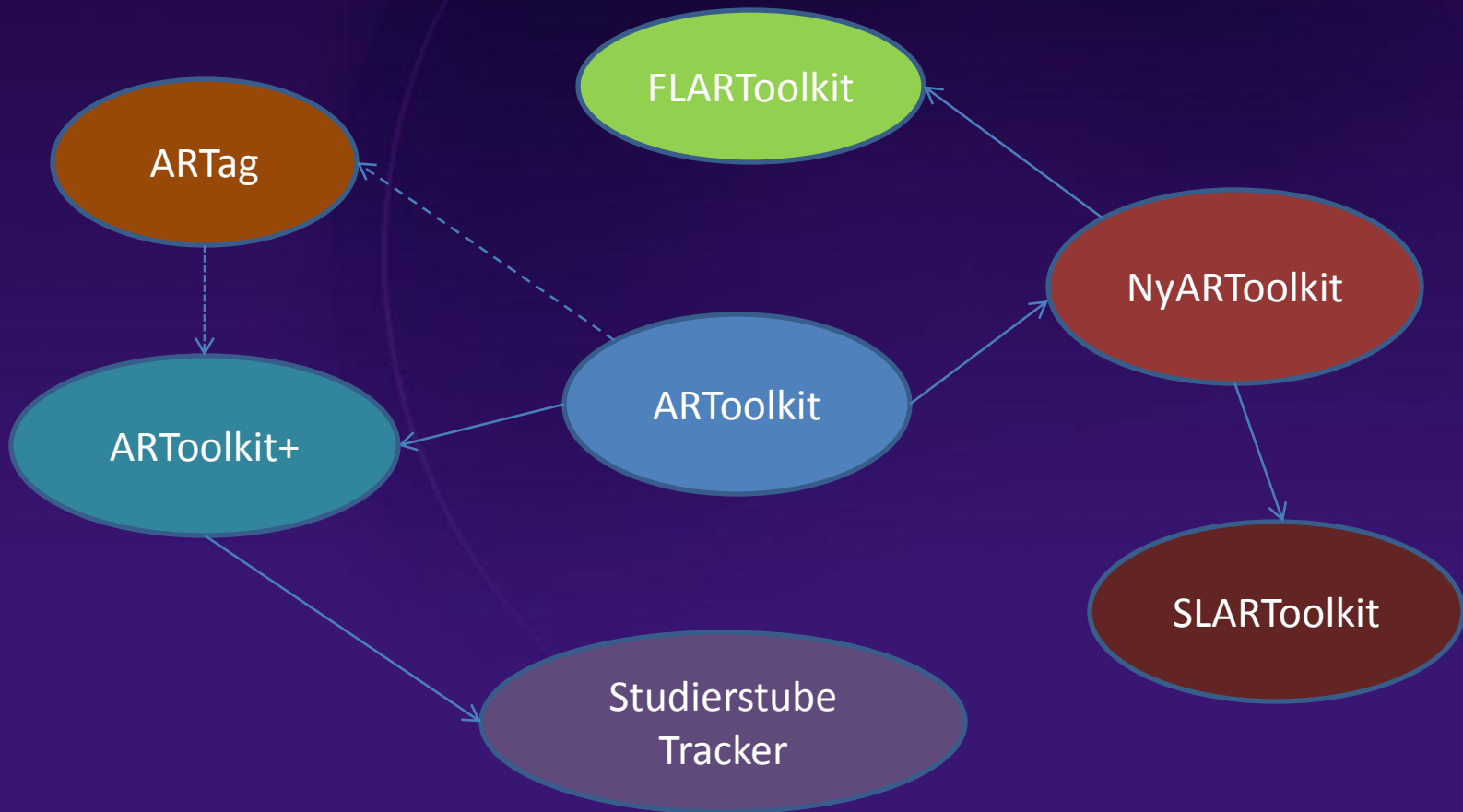
Что будем распознавать?

- Черно-белые маркеры
- Отрезки и кривые
- Ключевые точки
- Лица

Шаги реализации

- Захват изображения
- Анализ изображения
 - Обнаружение наличия искомых объектов
 - Их локализация
- Отрисовка виртуальных объектов

Черно-белые маркеры



ARToolkit

<http://www.hitl.washington.edu/artoolkit/>

- Разработана Dr. Hirokazu Kato
- Написана на С.
- При слежении используются черные квадраты.
- Можно использовать любые квадратные шаблоны.
- Простой код калибровки камеры.
- Достаточно быстрая для приложений реального времени.
- SGI IRIX, Linux, MacOS и Windows OS версии.
- Поставляется с исходными текстами.

NyARToolkit

<http://nyatla.jp/nyartoolkit/wiki/index.php?FrontPage.en>

- Реализация ARToolkit для java, c#, c++ и android
- С исходными текстами. Возможности аналогичные.
- Код довольно своеобразный.

FLARToolkit

<http://www.libspark.org/wiki/sagoosha/FLARToolKit/en>

- Порт java версии NyARToolkit на flash
- Доминирует в сети, т.к. поддержка flash'ем камеры делает это возможным.
- Лицензия: GPL
- Пример:
http://ge.ecomagination.com/smartgrid/#/augmented_reality

SLARToolkit

<http://slartoolkit.codeplex.com/>

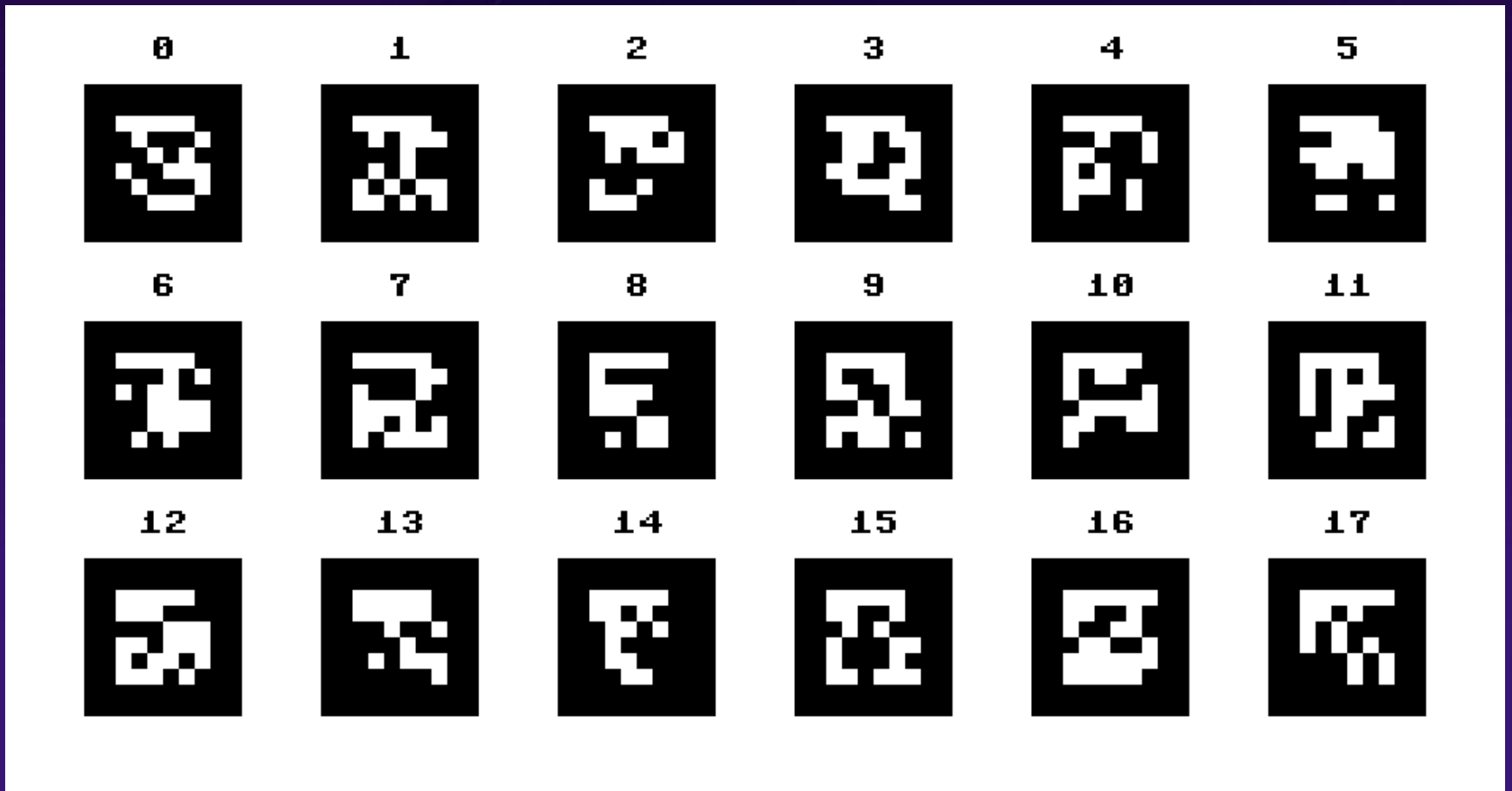
- Реализация под Silverlight4
- Частично базируется на NyARToolkit
- Поддержка Silverlight объекта CaptureSource
- Гибкость через детектор на базе WriteableBitmap
- Распознавание нескольких маркеров
- Простые черные маркеры
- Пользовательские маркеры
- Работает в реальном времени
- Легка в использовании
- Документирована
- Базируется на устоявшихся алгоритмах и технологиях.
- Использует библиотеку Matrix3DEx
- Крайне неудобно отлаживать ☹

ARTag

<http://www.artag.net>

- Сделан по мотивам ARToolkit
- Отличия от ARToolkit
 - Содержит набор из 2002 predetermined маркеров (ID-маркеры), поэтому файлы шаблонов больше не нужны.
 - За счет применения другого алгоритма уменьшено число ложных срабатываний
 - Маркеры обнаруживаются и при неравномерном освещении
 - Маркеры обнаруживаются при частичном перекрытии бордюра.

ARTag

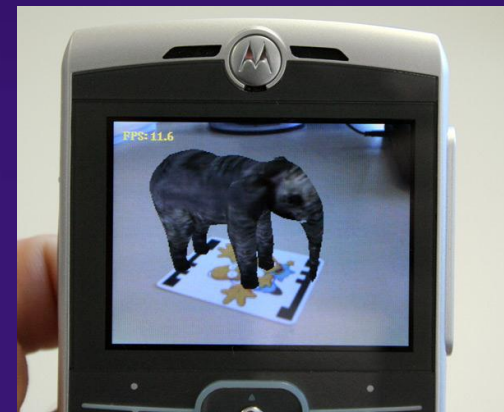
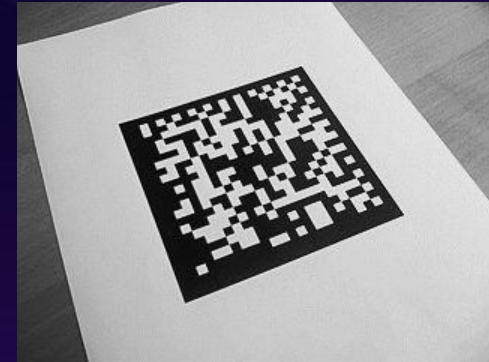


ARToolkitPlus

- Создан на базе ARToolkit, но под влиянием ARTag
- Написан на C++ Class-based API с использованием шаблонов
- До 4096 predetermined markers
- Новые форматы изображений камеры (RGB565, Gray)
- Переменная ширина границы маркеров.
- Множество ускорений для устройств класса low-end (арифметика с фиксированной точкой и т.п.)
- Более не обновляется, заменен Studierstube Tracker

Studierstube Tracker

- http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php
- API основанный на классах
- Несколько типов маркеров
 - Шаблонные (В стиле ARToolkit)
 - ID-маркеры (ARTag)
 - DataMatrix маркеры
 - Рамочные маркеры
 - Разорванные маркеры
- Высочайшая производительность даже на low-level устройствах.
- Низкое потребление памяти.
- Маленькие размеры кода
- До 4096 id-маркеров
- До 4 миллионов рамочных маркеров
- Новые рамочные маркеры кодируют до 22 бит информации с коррекций до трех неверных.
- Поддержка форматов RGB24, RGB32, RGB565, YUV12
- Переменная ширина границы маркера
- Высокая переносимость, потому как не используются сторонние библиотеки
- Надежная работа даже при неравномерном освещении
- Работает под windows, Linux, MacOS, iPhone



Распознавание отрезков и кривых

Преобразование Хафа (Hough transform)

Метод обнаружения прямых и кривых линий на полутоновых или цветных изображениях. Метод позволяет указать параметры семейства кривых и обеспечивает поиск на изображении множества кривых заданного семейства. Мы рассмотрим его применение для поиска на изображении прямолинейных отрезков и дуг окружностей.

В алгоритме преобразования Хафа используется аккумуляторный массив, размерность которого соответствует количеству неизвестных параметров в уравнении семейства искомых кривых. Например, при обнаружении прямых, описываемых уравнением $y=m*x+b$, для каждой прямой необходимо вычислить значения двух параметров m и b . При этом в массиве в элементах $A[M,B]$ накапливаются значения, указывающие на вероятность наличия на изображении прямой $y=m*x+b$, где M и B соответствуют дискретным значениям m и b .

Массив A используется в алгоритме Хафа для проверки каждого пиксела изображения и его окрестности. Определяется присутствует ли в данном пикселе выраженный край. Если присутствует, то вычисляются параметры искомой кривой, проходящей через данный пиксел. После оценки параметров прямой в данном пикселе они дискретизируются для получения соответствующих значений M и B , и значение массива $A[M,B]$ увеличивается. В некоторых реализациях увеличение выполняется на единицу, в других на величину мощности края в обработанном пикселе. После обработки всех пикселов выполняется поиск локальных максимумов в аккумуляторном массиве. Точки локальных максимумов соответствуют параметрам наиболее вероятных прямых на изображении.

Преобразование Хафа

- Более подробно о преобразовании Хафа можно прочесть, например, здесь: <http://habrahabr.ru/blogs/algorithm/102948/> или здесь: <http://cgm.computergraphics.ru/content/view/107>
- Пример реализации – библиотека **AForge.NET** <http://www.aforge.net/>

Недостатки метода

- Скорость
- Низкая точность

SURF (Speeded Up Robust Features)

- Впервые представлен на ECCV 2006 Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool (ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf)
- Принцип: поиск особых точек изображения и создание их дескрипторов, инвариантных к масштабу и вращению с помощью матрицы Гессе.
Для каждой особой точки считается градиент максимального изменения яркости и масштаб, взятый из масштабного коэффициента матрицы Гессе.

[Больше математики](#)

Недостатки

- Не выделяет объект из фона
- Плохо работает на предметах без ярко выраженной текстуры
- Скорость

SURF (Speeded Up Robust Features)

SURF - Homepage http://www.vision.ee.ethz.ch/~surf/download_ac.html

- Закрытые исходные тексты
- Для некоммерческого использования
- C++

OpenSURF

ASSURF - Homepage <http://code.google.com/p/in-spirit/wiki/ASSURF>

- ActionScript library
- С открытыми исходными текстами

OpenSURF – Homepage <http://www.chrisevansdev.com/computer-vision-opensurf.html>

- С открытыми исходными текстами
- C++, C#

OpenASURF – Homepage <http://theveganrobot.com/OpenASURF/>

- Реализация OpenSURF для Android

SURF (Speeded Up Robust Features)

P-SURF – Homepage <http://processingsurf.altervista.org/>

- Реализация для MacOSX

javasurf – Homepage <http://code.google.com/p/javasurf/>

- Платформонезависимая реализация на java, основанная на P-SURF.

Так же есть реализации с использованием OpenCV, GPU, CUDA.

Ссылки на них можно найти здесь: <http://en.wikipedia.org/wiki/SURF>

Адаптация OpenSURFcs под silverlight4

Распознавание лиц

FaceLight - Homepage <http://facelight.codeplex.com/>

- Silverlight
- Лицензия Ms-PL
- Принцип: фильтрация по цвету кожи в цветовой модели YCbCr
- [Демонстрация](#)

Luxand FaceSDK – Homepage <http://www.luxand.com/facesdk/index.php>

- Коммерческая
- C++, C#, VB и Borland Delphi
- Заявлена также работа под Linux и MacOS.
- Определяет отдельные части лица (брови, глаза, и т.п.), а также повороты.
- [Демонстрация Распознавания лиц с использованием преобразования Хафа](#)

Спасибо за внимание!