Введение в разработку под Kinect

WTF?!

Версии

Kinect for Windows

- **Коммерческая лицензия**
- Оптимизирован для использования с компьютерами под управлением Windows
- Оптимизирован для работы на близких дистанциях

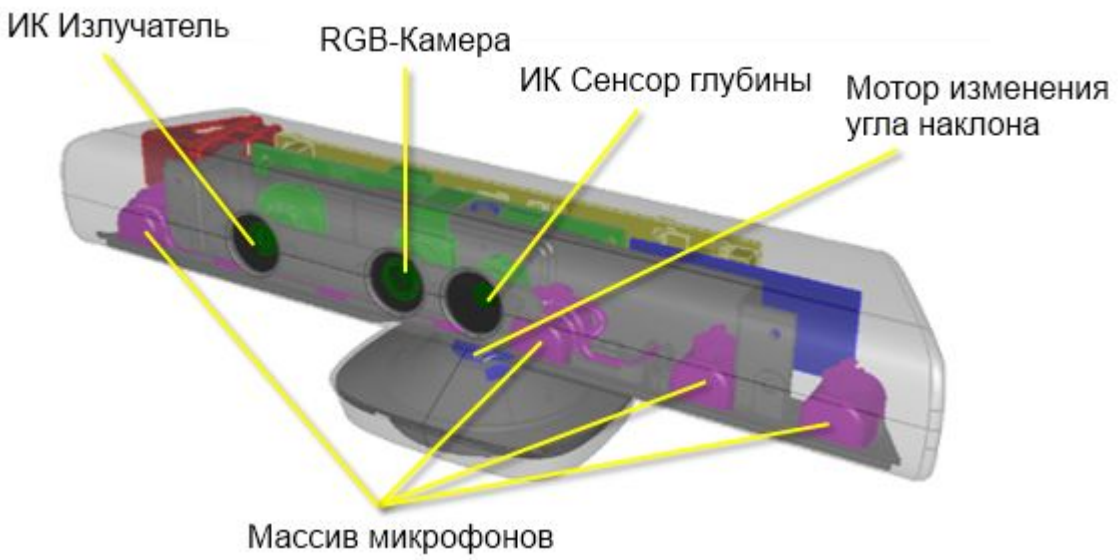
Kinect for Xbox 360

- Может использоваться только для девелоперских целей
- Оптимизирован для использования в Xbox 360

Устройство Kinect

- RGB камера
- 3D камера глубины

  (+ определение пользователя)
- Массив из 4х микрофонов
- Мотор изменения наклона девайса



Kinect SDKs

- OpenKinect SDK (Community)
- OpenNI Kinect SDK (PrimeSense)
- Kinect For Windows SDK (Microsoft)

Инициализация девайса

- Доступ к устройству осуществляется через коллекцию KinectSensor.KinectSensors. Пример получения доступа к первому устройству:

```
var kinect = KinectSensor.KinectSensors[0];
```

- Запуск устройства:

```
kinect.Start();
```

RGB-камера

RGB камера ИК-сенсор ИК-проектор

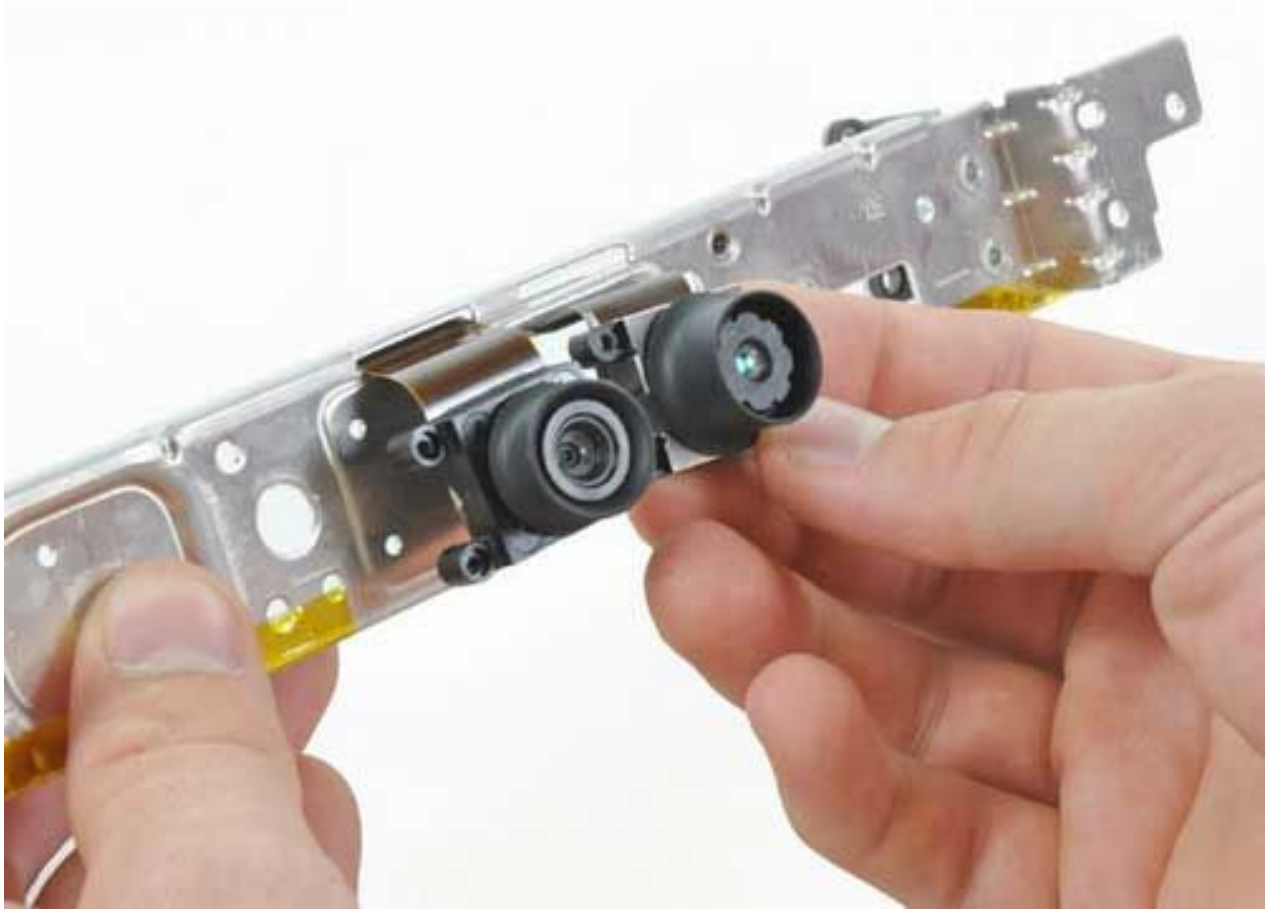
RGB-камера

- Поддерживаемые разрешения
  - RGB 640x480 (30 FPS)
  - RGB 1280x960 (12 FPS)
  - YUV 640x480 (15 FPS)
- Данные с камеры поступают в виде одномерного байтового массива, содержащего по 4 элемента на каждый пиксель
- Данные в массиве хранятся в формате BGRA

RGB-камера

Демонстрация

Камера глубины



Камера глубины

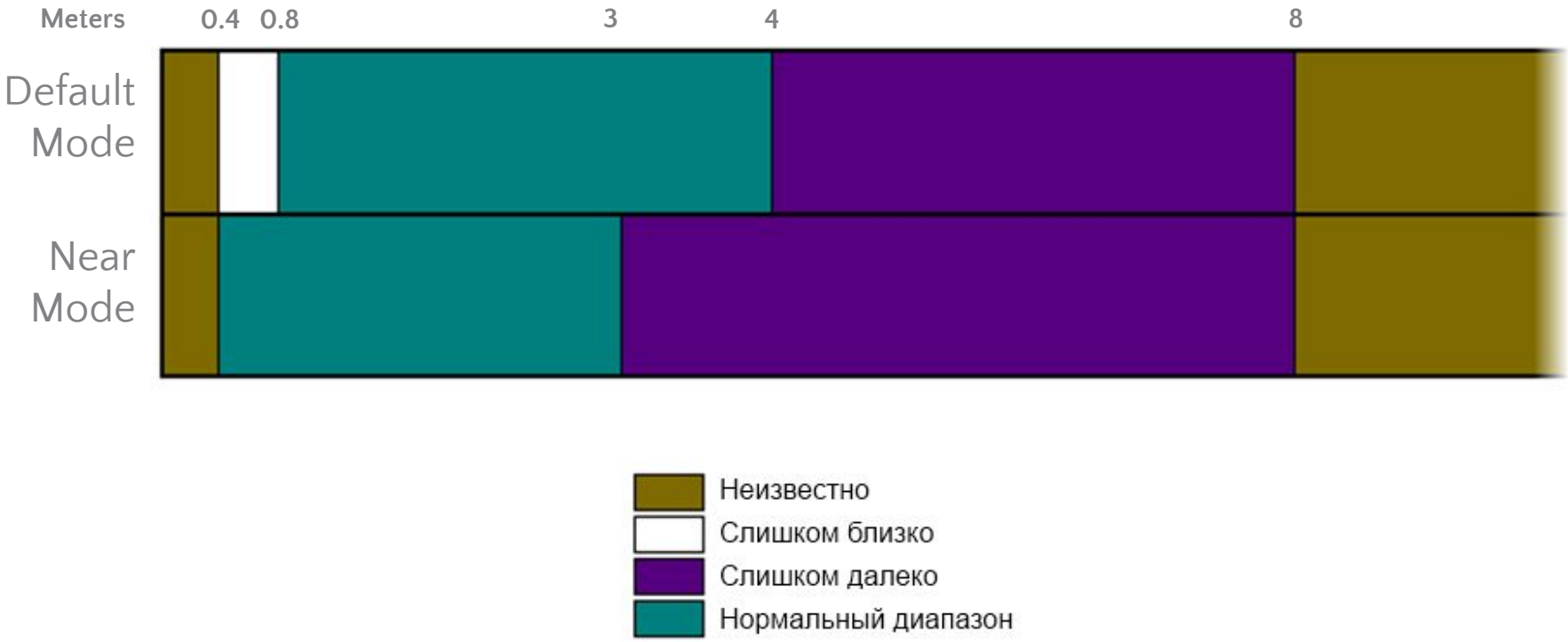
- Датчик глубины состоит из инфракрасного проектора, объединенного с монохромной КМОП-матрицей (CMOS APS Matrix), что позволяет датчику Kinect получать трёхмерное изображение при любом естественном освещении.

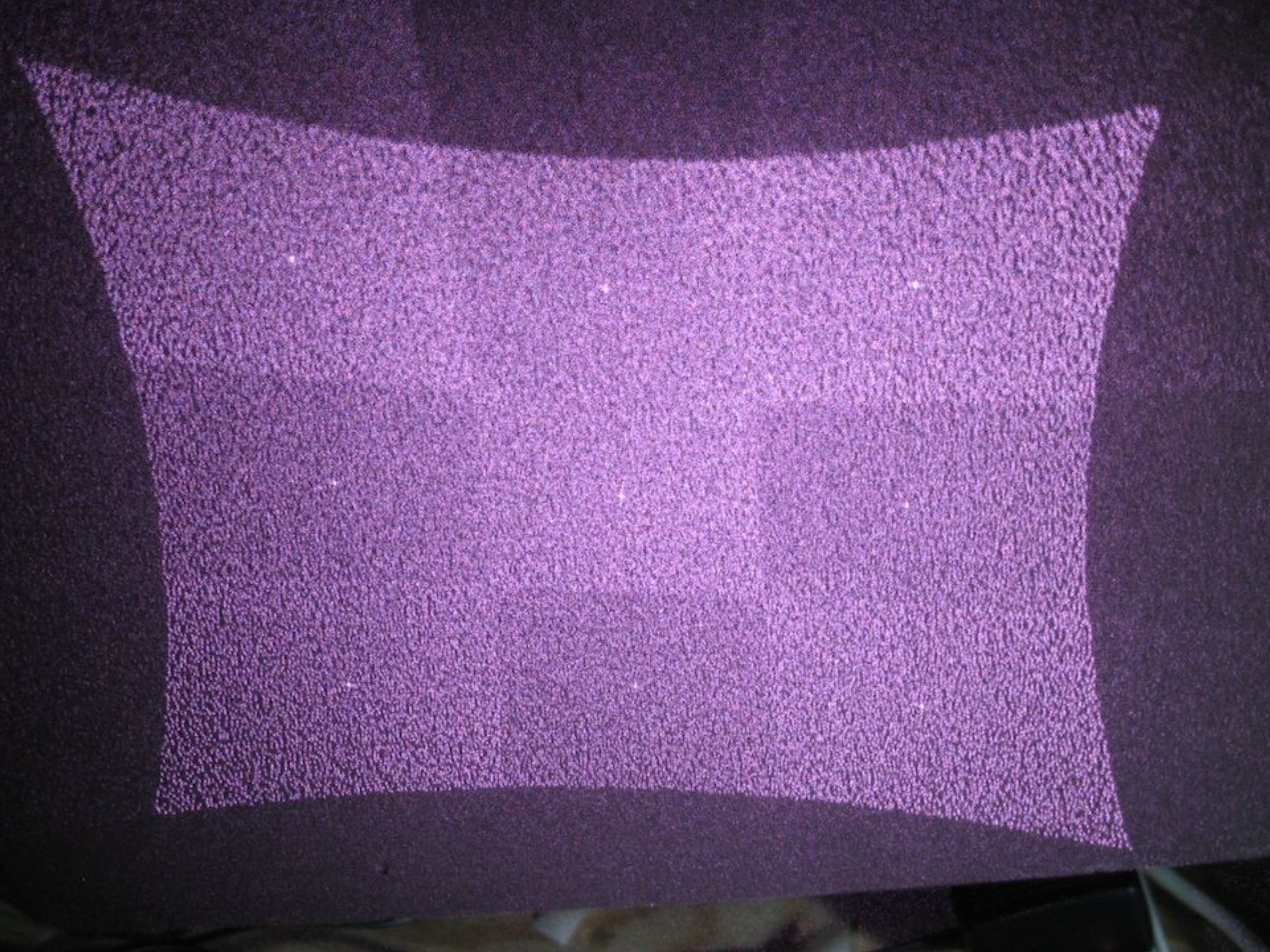
- Поддерживаемые разрешения
  - 640x480 (30 FPS)
  - 320x240 (30 FPS)
  - 80x60 (30 FPS)

- Поддерживаемые дистанции
  - 80см – 4м (DepthRange.Default - Kinect for xBox 360)
  - 40см – 3м (DepthRange.Near - Kinect for PC, Kinect SDK v1.5)

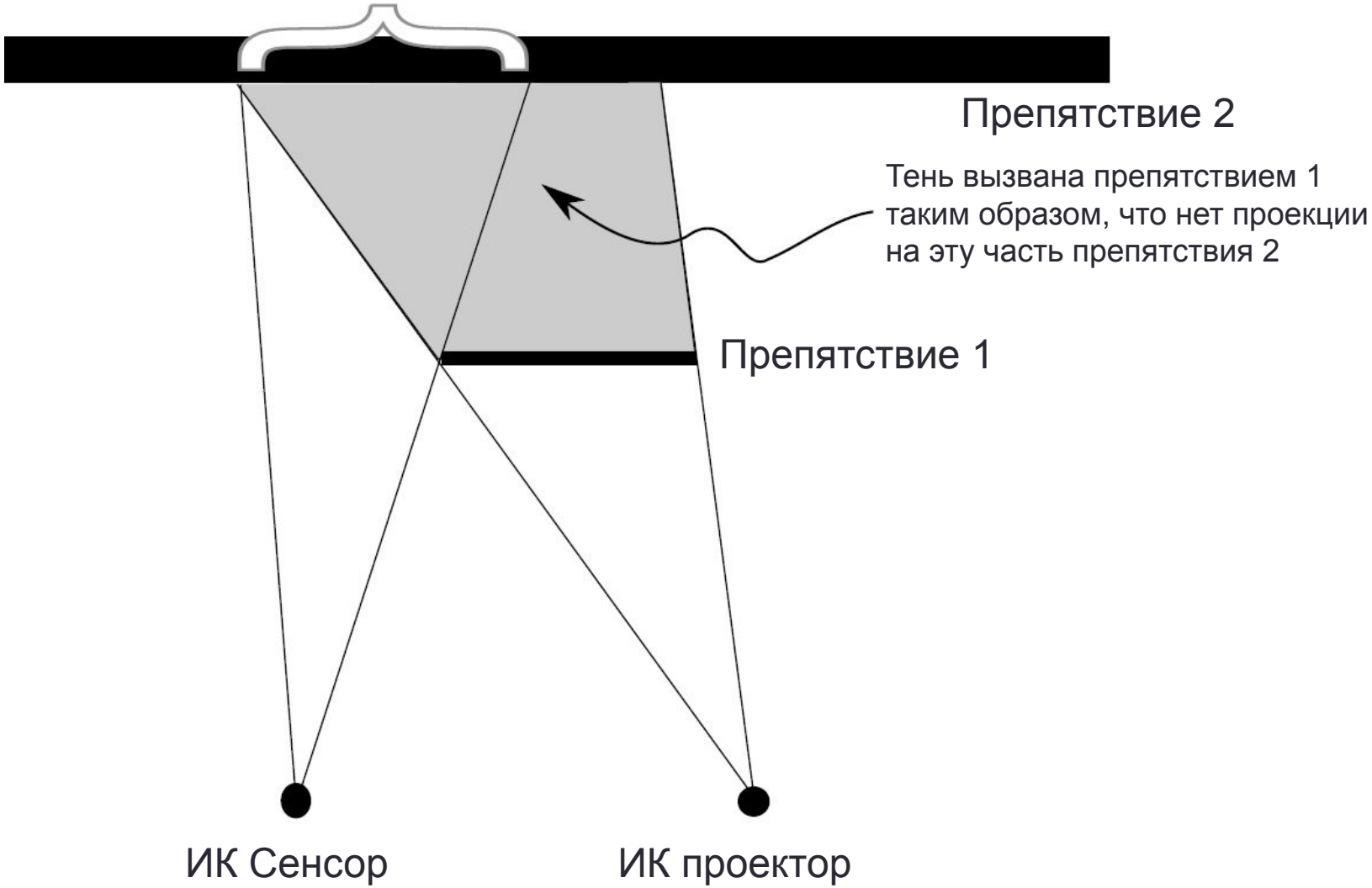
- Данные с камеры глубины поступают в виде одномерного массива типа short[]

Дистанции


Meters
0.4
0.8
3
4
8
Default Mode
Near Mode
Неизвестно
Слишком близко
Слишком далеко
Нормальный диапазон



Черные тени – что это?



Индекс игрока

- Массив данных глубины содержит информацию как по удаленности каждой точки, так и индекс игрока

- Получение индекса игрока:

```
var player = depthMatrix[i] & DepthImageFrame.PlayerIndexBitmask;//7
```

- Получение глубины:

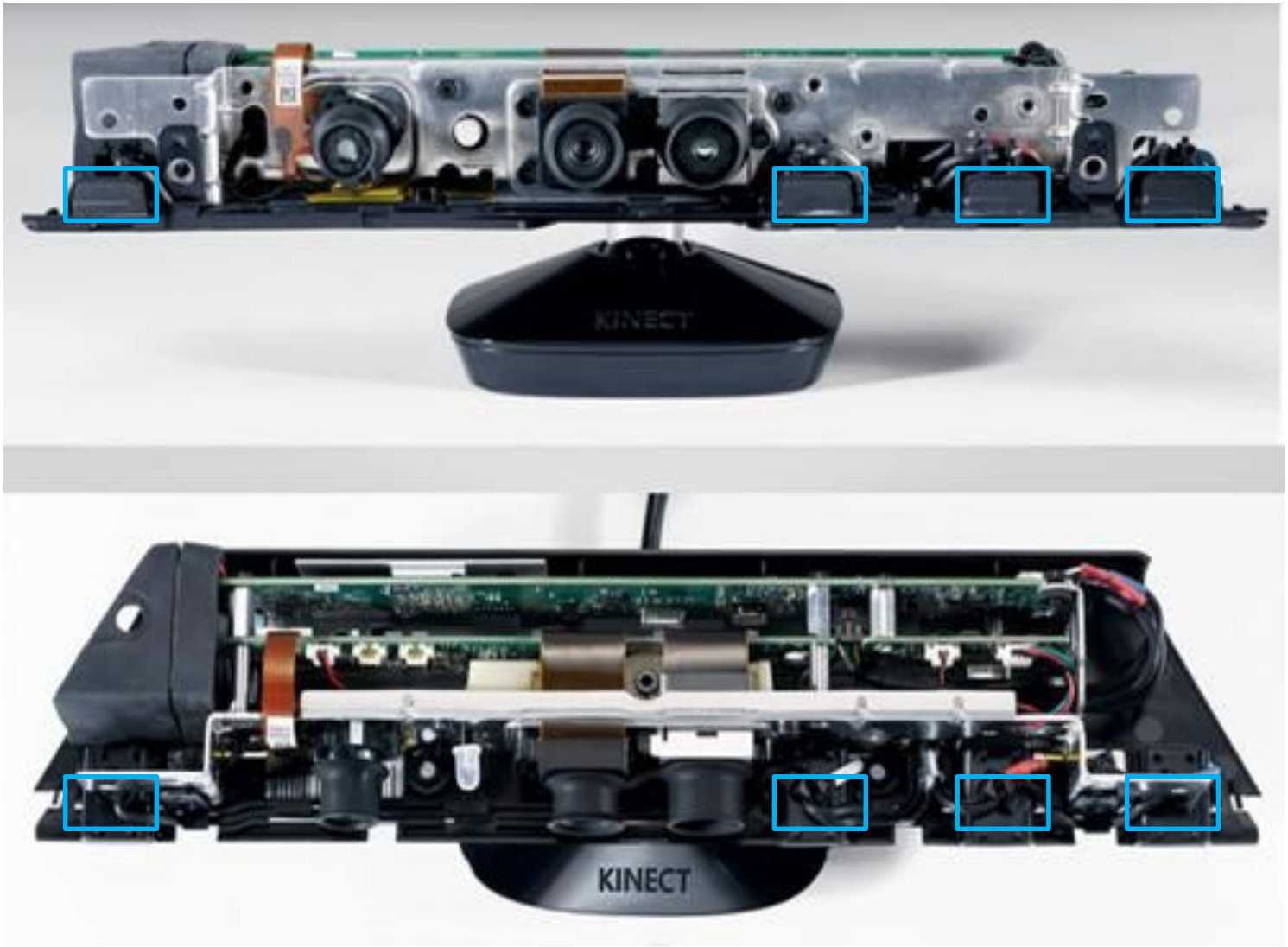
```
var realDepth = depthMatrix[i] >> DepthImageFrame.PlayerIndexBitmaskWidth;
//3
```

3003	0	0	0	0	1	0	1	1	1	0	1	1	1	0	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
375	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	1

Камера глубины

Демонстрация

Массив микрофонов



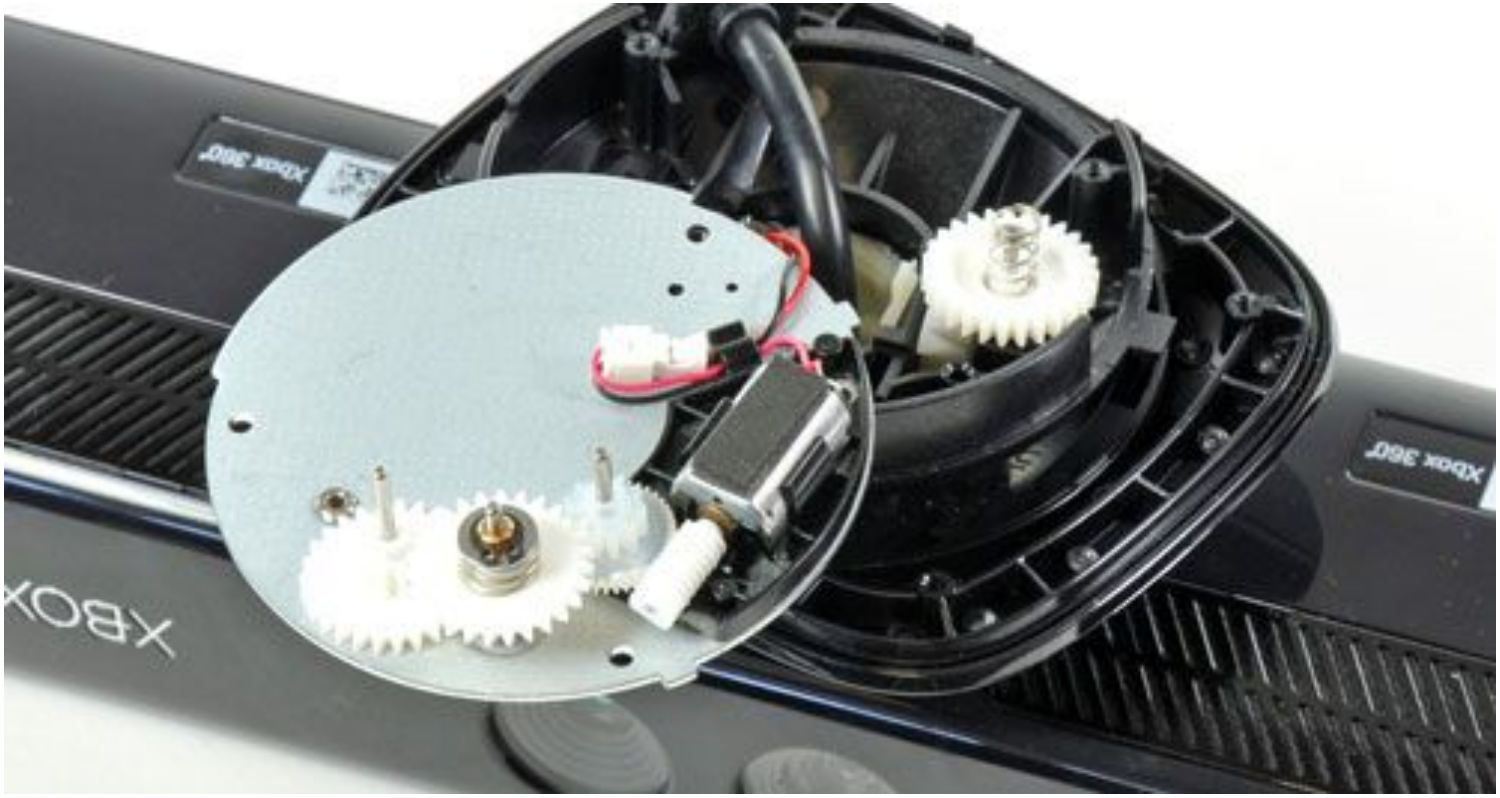
Массив микрофонов

- Микрофонная решетка позволяет производить локализацию источника звука и подавление шумов, что дает возможность говорить без наушников и микрофона.
- Поддерживается 16-битный звук с частотой дискретизации 16 kHz
- Sound Source Angle – угол и степень уверенности того, где находится источник звука
- Beam Angle – угол, использующийся при записи аудио, можно трактовать как направленный микрофон
- Максимальный угол определения звука ±50

Массив микрофонов

Демонстрация

Мотор изменения угла наклона Kinect



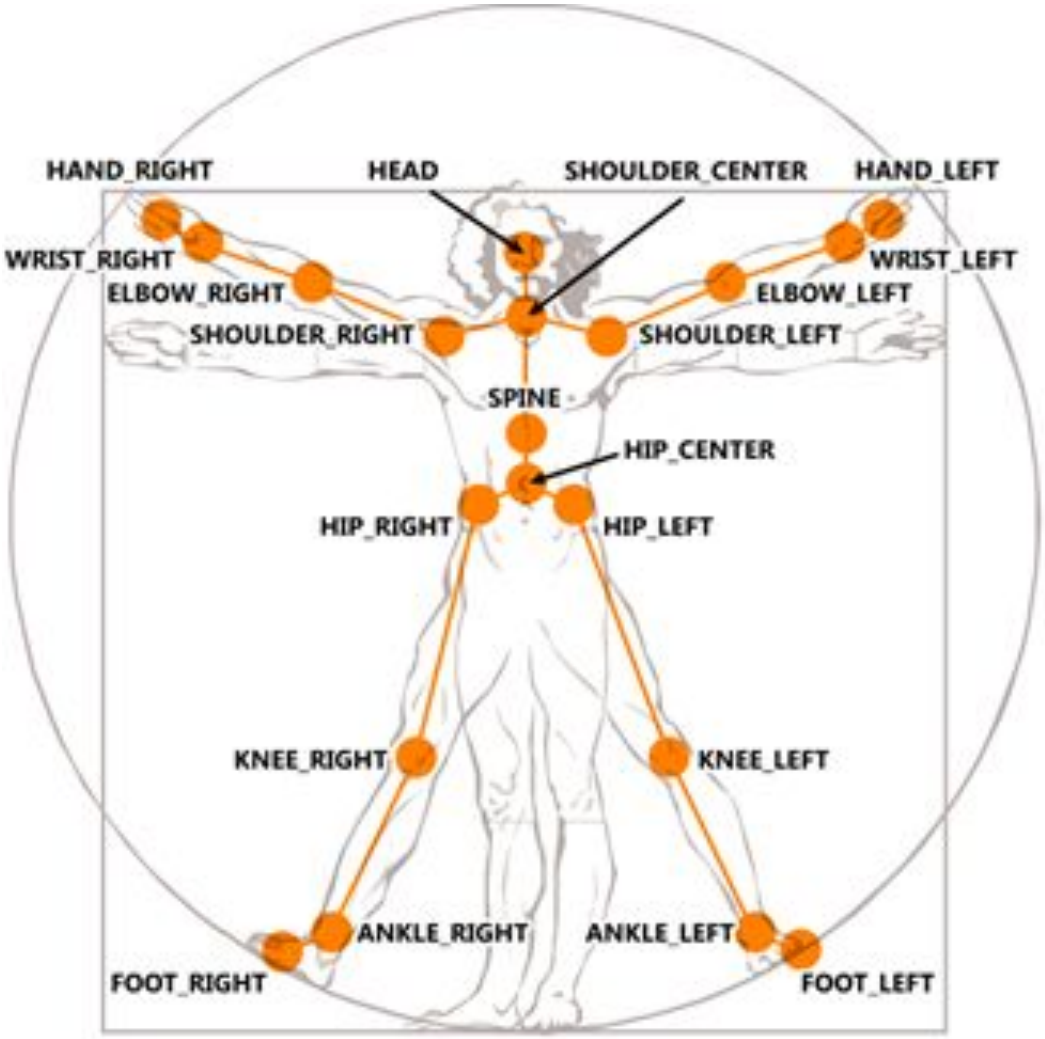
Мотор изменения угла наклона Kinect

- Служит для изменения градуса наклона устройства
- Минимальный угол наклона -27 градусов; максимальный +27
- Не существует прямого доступа к мотору через методы. Угол наклона можно изменить только через свойство kinect.ElevationAngle типа int

Мотор изменения угла наклона Kinect

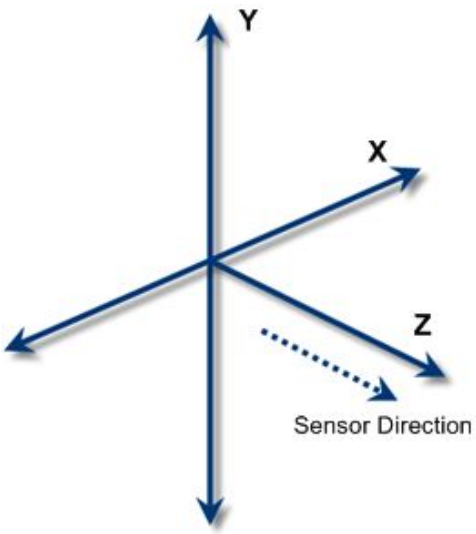
Демонстрация

SkeletonFrame – собери меня полностью



Скелет человека и Kinect

- Скелет состоит из:
  - 20 суставов (Joint) для обычного режима
  - 10 суставов для сидячего режима
- Возможна поддержка одновременно 2х активных игроков
- Может следить одновременно за 6 игроками
- Данные, полученные из SkeletonFrame являются массивом типа Skeleton[], каждый элемент которого содержит коллекцию суставов Joints с 3D-координатами своего положения в прост-ве



Параметры сглаживания скелета

- Smoothing(сглаживание) – default 0.5
- Correction(коррекция) – default 0.5
- Prediction(прогнозирование) – default 0.5
- JitterRadius(радиус дрожания) – default 0.05
- MaxDeviationRadius(максимальный радиус отклонения) – default 0.05

Позиционирование

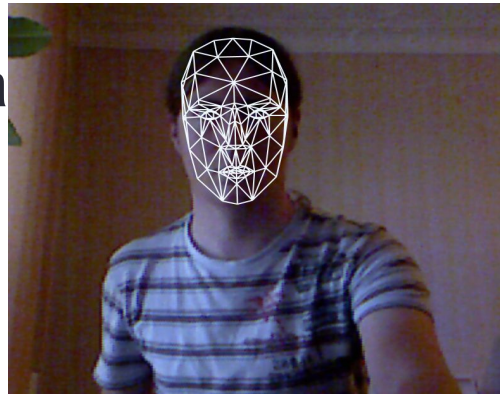
- Класс DepthImageFrame содержит:
- `DepthImagePoint MapFromSkeletonPoint(SkeletonPoint skeletonPoint)` – позволяет получить координаты пикселя глубины из сустава скелета
- `ColorImagePoint MapToColorImagePoint(int depthX, int depthY, ColorImageFormat colorImageFormat)` – позволяет получить координаты пикселя цвета из координат пикселя глубины
- `SkeletonPoint MapToSkeletonPoint (int depthX, int depthY)` – позволяет получить координаты сустава скелета из координат пикселя глубины
- Актуально для преобразований координат между разными системами координат.

SkeletonFrame

Демонстрация

FaceTracking API

- В основе алгоритма распознавания лежит алгоритм Active appearance model (AAM – активная модель внешности/активная видовая модель)
- При работе с Face Tracking API, разработчик посылает структуру данных, содержащую информацию с RGBA-изображением и массив данных глубин. В результате обработки этих данных, разработчик получает на выходе коллекцию 2D-координат точек элементов лица
- При сопоставлении лица и распознанных точек лица, используется Candide-3 маска



FaceTracking API

Демонстрация

Цена вопроса

Kinect for Xbox 360 – $150 (~6 тыс. р.)
Kinect for PC – $250 (undefined тыс. р.)

Комплект поставки:

- девайс
- кабель питания
- руководство пользователя
- игра (Kinect Adventures)

Источники

- http://kinectforwindows.org – портал для Kinect-разработчиков
- http://channel9.msdn.com/coding4fun/kinect – канал, посвященный Kinect
- http://channel9.msdn.com/Series/KinectQuickstart – краткое видео руководство
- http://blogs.msdn.com/b/kinectforwindows/ – блог разработчиков Kinect
- http://msdn.microsoft.com/en-us/library/hh855348 – Kinect for Windows Programming Guide

???

Вопросы