



New iPhone Camera Hack!

Mobile Act NAGOYA #2 - 2016/9/30

自己紹介

@temoki



- PL : エンジニア = 7 : 3
- iOS : Android = 9 : 1
- カメラ/写真 アプリ
- 雰囲気メガネ アプリ

新しい iPhone 買いましたか？

7 Plus + 艶々 JET BLACK

- 9月9日 16:01 予約受付開始
- 9月9日 16:07 予約できました
- 9月30日 まだ出荷準備中... 🥺

今回の  の個人的な目玉は...

もちろんデュアルカメラ！

二つのカメラが一つになって撮影します。

- より明るくなった広角カメラ F1.8 (28mm相当)
- 新しく加わった望遠カメラ F2.8 (56mm相当)

デュアルで何が嬉しいのか？

-  初の光学ズーム 2x
 -  ~~デジタルズームと合わせて10×~~ (嬉しい)
- 二つのセンサー情報をフル活用したシーン認識
 -  最適な写真を撮影
- ポートレートモード ※での被写界深度エフェクト
 -  一眼レフカメラのようなボケを表現

※ パブリックベータ配信中の iOS 10.1 から利用可能

そして色再現

- 広色域キャプチャー
 - ➡ 従来の sRGB より広い色空間の P3 で撮影
- 広色域ディスプレイ
 - ➡ True Tone ディスプレイで P3 色空間を再現

過去最強と言わざるをえない！



iOS 10 Camera API

Camera API Diff

- **Dual Camera & Camera Discovery**
- **New Photo Capture API**
 - **RAW Photo**
 - **Live Photos**
- **Wide Color**

Dual Camera & Camera Discovery

Dual Camera & Camera Discovery

- `AVCaptureDeviceDiscoverySession` - NEW!

```
let discoverySession = AVCaptureDeviceDiscoverySession(  
    deviceTypes: [.builtInDuoCamera],  
    mediaType: AVMediaTypeVideo,  
    position: .back)  
  
let devices: [AVCaptureDevice]? = discoverySession?.devices
```

Dual Camera & Camera Discovery

- `AVCaptureDeviceType` - *NEW!*
 - `.builtInWideAngleCamera`
 - `.builtInTelephotoCamera`
 - `.builtInDuoCamera`

同時キャプチャーできる！？

できたら被写界深度エフェクトみたいに

視差から立体認識してゴニヨゴニヨ...

できません 😭

.builtInDuoCamera

- 二つのカメラで最適化された一枚の写真だけを出力
- 撮影に細かいコントロールはできない (全てお任せ)
- 後述の RAW フォーマット撮影も不可能

**.builtInTelephotoCamera +
.builtInTelephotoCamera**

```
let captureSession = AVCaptureSession()
captureSession.addInput(wideAngleCameraDevice)
captureSession.addInput(telephotoCameraDevice)
```

 **Terminating app due to uncaught exception !!!**

Multiple audio/video AVCaptureInputs are not currently supported

New Photo Capture API

New Photo Capture API

- `AVCaptureStillImageOutput` - *Deprecated*
- `AVCapturePhotoOutput` - *NEW!*
 - **RAW Photo Capture**
 - Live Photo Capture

RAW って？

- イメージセンサーから得られた未加工のデータ
- このデータから下記 **現像** 処理を経て保存される
 - デモザイクという色付け処理
 - 色・明るさなどの自動レタッチ処理
 - JPEG 等のフォーマットに圧縮

JPEG と RAW

- JPEG
 - プロ (= Apple) が焼き上げたケーキのようなもの
 - 再加工には限界がある
 - 誰でも・お手軽・コンパクト
- RAW
 - ケーキの材料一式のようなもの
 - 後からいくらでも加工ができる
 - 技術要・大変・大きい

RAW撮影ができるデバイス

- iPhone 7, 7 Plus
- iPhone 6s, 6s Plus
- iPhone SE
- iPad Pro 9.7 inch

※ ただし iSight (Back) カメラのみ

RAW撮影ができるアプリ

- **NG** 標準カメラアプリ
- **OK** Adobe Lightroom アプリ、など

RAW撮影ができるAPI

- **NG** UIImagePickerController
- **OK** AVFoundation , AVCapturePhotoOutput

API で RAW 撮影

- `AVCapturePhotoOutput`

```
let types = photoOutput.availableRawPhotoPixelFormatTypes
let type = types.first!.uint32Value
let settings = AVCapturePhotoSettings(rawPixelFormatType: type)
photoOutput.capturePhoto(with: settings, delegate: self)
```

API で RAW 撮影後

- `AVCapturePhotoCaptureDelegate`
- `rawSampleBuffer: CMSampleBuffer?` から RAW データにアクセス可能

```
func capture(_ captureOutput: AVCapturePhotoOutput,  
            didFinishProcessingRawPhotoSampleBuffer rawSampleBuffer: CMSampleBuffer,  
            previewPhotoSampleBuffer: CMSampleBuffer?,  
            resolvedSettings: AVCaptureResolvedPhotoSettings,  
            bracketSettings: AVCaptureBracketedStillImageSettings?,  
            error: Error?) {  
}  
}
```

RAW データをそのまま触る

- `CVPixelBuffer` にバイトデータとして含まれる
- iPhone 7 Plus だと “rgg4” というフォーマット
 - `kCVPixelFormatType_14Bayer_RGGB`
 - Bayer 14-bit Little-Endian, packed in 16-bits, ordered R G R G... alternating with G B G B...

```
// CVPixelBuffer?  
let pixelBuffer = CMSampleBufferGetImageBuffer(rawSampleBuffer!)
```

RAW データを保存する

- Adobe DNG フォーマットで保存
- DNG は Adobe が標準化を目指す RAW のフォーマット
- **D**igital - **N**e**G**ative

```
// Data?  
let data = AVCapturePhotoOutput.dngPhotoDataRepresentation(  
    forRawSampleBuffer: rawSampleBuffer!,  
    previewPhotoSampleBuffer: previewPhotoSampleBuffer)
```

RAW データを現像する

- `import CoreImage.framework`
- RAW データから `CIFilter` オブジェクトを作成
- RAW 現像用のオプションを指定
- `CIFilter` から `CIImage` を生成

```
let rawFilter = CIFilter(imageURL: rawURL, options: nil)

// Noise Reduction
let nrKey = kCIInputLuminanceNoiseReductionAmountKey
if let nr = rawFilter?.value(forKey: nrKey) {
    rawFilter.setValue(nr.doubleValue + 0.1, forKey: nrKey)
}

let image: CIImage? = rawFilter?.outputImage
// -> CGImage, UIImage, JPEG, ...
```

まとめ

