AVCapturePhotoOutput—Beyond the Basics
Addendum to Session 501
Session 511

Brad Ford Apple
Live Photos
Live Photos

RAW + DNG
Live Photos

RAW + DNG

Wide Color
What You’ll Learn

AVCapturePhotoOutput—Beyond the Basics
What You’ll Learn

AVCapturePhotoOutput—Beyond the Basics

• Scene Monitoring
What You’ll Learn

AVCapturePhotoOutput—Beyond the Basics

• Scene Monitoring
• Resource Preparation and Reclamation
What You’ll Learn

AVCapturePhotoOutput—Beyond the Basics
• Scene Monitoring
• Resource Preparation and Reclamation
Camera Privacy Policy Changes
AVCapturePhotoOutput Design Features
AVCapturePhotoOutput Design Features

Functional programming model
AVCapturePhotoOutput Design Features

Functional programming model
Photo settings encapsulation
AVCapturePhotoOutput Design Features

Functional programming model

Photo settings encapsulation

A delegate-style interface for tracking the progress of photo capture requests
AVCapturePhotoOutput Design Features

Functional programming model
Photo settings encapsulation
A delegate-style interface for tracking the progress of photo capture requests
Resolving of photo settings to an immutable object
Scene Monitoring in AVCapturePhotoOutput
AVCapturePhotoOutput Scene Monitoring
AVCapturePhotoOutput Scene Monitoring
AVCapturePhotoOutput Scene Monitoring
Scene Monitoring in AVCapturePhotoOutput
Scene Monitoring in AVCapturePhotoOutput

Flash scene
Scene Monitoring in AVCapturePhotoOutput

Flash scene
- True Tone flash for rear camera
Scene Monitoring in AVCapturePhotoOutput

Flash scene

- True Tone flash for rear camera
- Retina flash for front camera
Scene Monitoring in AVCapturePhotoOutput
Scene Monitoring in AVCapturePhotoOutput

Still Image Stabilization scene
Scene Monitoring in AVCapturePhotoOutput

Still Image Stabilization scene
- Multiple image fusion capture
Scene Monitoring in AVCapturePhotoOutput

Still Image Stabilization scene
- Multiple image fusion capture
- Reduces low light blur
Scene Monitoring in AVCapturePhotoOutput

Still Image Stabilization scene

• Multiple image fusion capture
• Reduces low light blur
Scene Monitoring in AVCapturePhotoOutput

Still Image Stabilization scene

- Multiple image fusion capture
- Reduces low light blur

Time

- Bright light exposure duration
- Dark light exposure duration
Applicable Light Ranges for Flash Capture

Lux Levels

With SIS

Without SIS

Darker

Lighter
Applicable Light Ranges for Flash Capture

With SIS | Without SIS

Lux Levels

Darker | Lighter
Applicable Light Ranges for Flash Capture

Lux Levels

Darker | Lighter

With SIS | Without SIS
public var photoSettingsForSceneMonitoring: AVCapturePhotoSettings?
public var photoSettingsForSceneMonitoring: AVCapturePhotoSettings?

public var isStillImageStabilizationScene: Bool { get }

public var isFlashScene: Bool { get }
func setUpSceneMonitoring()
{
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring

    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                           options: .new, context: nil)
}
// Scene monitoring support

func setUpSceneMonitoring()
{
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring

    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                            options: .new, context: nil)
}
// Scene monitoring support

func setUpSceneMonitoring() {
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring

    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                            options: .new, context: nil)
}

/ Scene monitoring support

func setUpSceneMonitoring()
{
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring

    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                             options: .new, context: nil)
}
Scene Monitoring Defaults
Scene Monitoring Defaults

photoSettingsForSceneMonitoring is nullable
Scene Monitoring Defaults

photoSettingsForSceneMonitoring is nullable

```
isStillImageStabilizationScene = false
```
Scene Monitoring Defaults

photoSettingsForSceneMonitoring is nullable

isStillImageStabilizationScene = false
isFlashScene = false
Scene Monitoring Defaults

photoSettingsForSceneMonitoring is nullable

isStillImageStabilizationScene = false
isFlashScene = false

If flashMode = .off, isFlashScene is always false
Scene Monitoring Recommendations
Scene Monitoring Recommendations

Monitor what you intend to capture
Resource Preparation and Reclamation
AVCaptureSession Data Flow

AVCaptureDevice (Camera)

AVCaptureDeviceInput

AVCaptureConnection

AVCaptureVideoPreviewLayer

AVCaptureSession

.startRunning()

AVCapturePhotoOutput

AVCaptureMovieFileOutput
AVCaptureSession Data Flow

1. AVCaptureDevice (Camera)
2. AVCaptureDeviceInput
3. AVCaptureConnection
4. AVCaptureVideoPreviewLayer
5. AVCaptureSession
6. AVCapturePhotoOutput
7. AVCaptureMovieFileOutput
 AVCaptureSession Data Flow

AVCaptureDevice (Camera) -> AVCaptureDeviceInput

<table>
<thead>
<tr>
<th>AVCaptureConnection</th>
<th>AVCaptureConnection</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>AVCaptureVideoPreviewLayer</th>
<th>AVCaptureSession</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>AVCapturePhotoOutput</th>
<th>AVCaptureMovieFileOutput</th>
</tr>
</thead>
</table>

photoOutput.capturePhoto(with:, delegate:)

(Camera)
AVCaptureSession Data Flow

photoOutput.capturePhoto(with:, delegate:)
So Many Features, So Many Resources
So Many Features, So Many Resources

Uncompressed 420
So Many Features, So Many Resources

- Uncompressed 420
- Processed
So Many Features, So Many Resources

- Uncompressed 420
- Processed
- Flash
So Many Features, So Many Resources

- Uncompressed 420
- Processed
- Flash
- Still Image Stabilization
So Many Features, So Many Resources

- Uncompressed 420
- Processed
- Flash
- Still Image Stabilization

RAW
So Many Features, So Many Resources

- Uncompressed 420
- Processed
- Flash
- Still Image Stabilization

RAW

RAW + JPEG
So Many Features, So Many Resources

- Uncompressed 420
- Processed
- Flash
- Still Image Stabilization
- RAW
- RAW + JPEG
- Exposure Bracket
So Many Features, So Many Resources
So Many Features, So Many Resources

RAW + JPEG Exposure Bracket
How Much to Prepare?

Over-preparing
How Much to Prepare?

Over-preparing

• Bakes a cake every day of the year
How Much to Prepare?

Over-preparing

• Bakes a cake every day of the year

Under-preparing
How Much to Prepare?

Over-preparing
• Bakes a cake every day of the year

Under-preparing
• Misses the moment
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!)
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!
)
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!
)
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!
)

public var preparedPhotoSettingsArray: [AVCapturePhotoSettings] { get }
setPreparedPhotoSettingsArray Specifics
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
Reclaims unneeded resources if any
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
Reclaims unneeded resources if any
Reclaims everything if you pass an empty array
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
Reclaims unneeded resources if any
Reclaims everything if you pass an empty array
Calls you back when all resources are prepared
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
Reclaims unneeded resources if any
Reclaims everything if you pass an empty array
Calls you back when all resources are prepared
Returns error if resources couldn’t be prepared
setPreparedPhotoSettingsArray Specifics

Prepares resources for all the types of capture in your array of settings
Reclaims unneeded resources if any
Reclaims everything if you pass an empty array
Calls you back when all resources are prepared
Returns error if resources couldn’t be prepared

preparedPhotoSettingsArray default value is [AVCapturePhotoSettings()]
preparedPhotoSettingsArray Persistence
preparedPhotoSettingsArray Persistence

preparedPhotoSettingsArray persists across
preparedPhotoSettingsArray Persistence

preparedPhotoSettingsArray persists across
• AVCaptureSession start/stopRunning
preparedPhotoSettingsArray Persistence

preparedPhotoSettingsArray persists across

- AVCaptureSession start/stopRunning
- AVCaptureSession begin/commitConfiguration
preparedPhotoSettingsArray Persistence

preparedPhotoSettingsArray persists across
- AVCaptureSession start/stopRunning
- AVCaptureSession begin/commitConfiguration

preparedPhotoSettingsArray participates in AVCaptureSession begin/commitConfiguration deferred work semantics
Stopped AVCaptureSession Behavior
Stopped AVCaptureSession Behavior

setPreparedPhotoSettingsArray calls completion handler when you start running
Stopped AVCaptureSession Behavior

setPreparedPhotoSettingsArray calls completion handler when you start running
Multiple setPreparedPhotoSettingsArray calls cancel previous calls
Prepare Recommendations
Prepare Recommendations
Prepare Recommendations

Prepare
 Prepare before running
Prepare Recommendations

- Prepare
- Prepare before running
- Re-prepare when your UI changes
Non-Preparable Features
Non-Preparable Features

```swift
public var isHighResolutionCaptureEnabled: Bool
```
Non-Preparable Features

```swift
public var isHighResolutionCaptureEnabled: Bool

public var isLivePhotoCaptureEnabled: Bool
```
Non-Preparable Features

public var isHighResolutionCaptureEnabled: Bool
public var isLivePhotoCaptureEnabled: Bool
public var isLivePhotoAutoTrimmingEnabled: Bool
Camera Privacy Policy Changes
Privacy Review
Privacy Review

Photos and videos are personal, sensitive data
Privacy Review

Photos and videos are personal, sensitive data.
Beginning in iOS 7, users are notified when an app uses the Camera or Microphone.
Privacy Review

Photos and videos are personal, sensitive data

Beginning in iOS 7, users are notified when an app uses the Camera or Microphone

“AVCam” Would Like to Access the Camera

Don’t Allow | OK

“AVCam” Would Like to Access the Microphone

Don’t Allow | OK
Privacy Changes

Starting in iOS 10, apps must provide a reason for accessing sensitive data.
Privacy Changes

Starting in iOS 10, apps must provide a reason for accessing sensitive data.
Privacy Changes

Starting in iOS 10, apps must provide a reason for accessing sensitive data
Privacy Changes

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Launch screen interface file base name</td>
<td>String</td>
<td>Launch Screen</td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>Privacy - Camera Usage Description</td>
<td>String</td>
<td>to take photos and video</td>
</tr>
<tr>
<td>Privacy - Contacts Usage Description</td>
<td>String</td>
<td>APPL</td>
</tr>
<tr>
<td>Privacy - Health Share Usage Description</td>
<td>String</td>
<td>Main</td>
</tr>
<tr>
<td>Privacy - Health Update Usage Description</td>
<td>String</td>
<td>to record Live Photos and movies</td>
</tr>
<tr>
<td>Privacy - HomeKit Usage Description</td>
<td>String</td>
<td>5.0</td>
</tr>
<tr>
<td>Privacy - Location Always Usage Description</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Privacy - Location Usage Description</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Privacy - Location When In Use Usage Description</td>
<td>Array</td>
<td>(1 item)</td>
</tr>
<tr>
<td>Privacy - Media Library Usage Description</td>
<td>Array</td>
<td>(4 items)</td>
</tr>
<tr>
<td>Privacy - Microphone Usage Description</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>com.example.apple-samplecode.$(PRODUCT_NAME).rnc1034ident</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>???</td>
</tr>
<tr>
<td>Application requires iPhone environment</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>Supported interface orientations</td>
<td>Array</td>
<td>(4 items)</td>
</tr>
<tr>
<td>Privacy - Photo Library Usage Description</td>
<td>String</td>
<td>to save photos and videos</td>
</tr>
</tbody>
</table>
// Add to Info.plist

<plist version="1.0">
  <dict>
    <key>NSCameraUsageDescription</key>
    <string>to take photos and video</string>
    <key>NSMicrophoneUsageDescription</key>
    <string>to record Live Photos and movies</string>
    <key>NSPhotoLibraryUsageDescription</key>
    <string>to save photos and videos</string>
  </dict>
</plist>
Summary
Summary

AVCapturePhotoOutput allows fine control of scene monitoring behavior
Summary

AVCapturePhotoOutput allows fine control of scene monitoring behavior
AVCapturePhotoOutput allows on-demand resource allocation and reclamation
Summary

AVCapturePhotoOutput allows fine control of scene monitoring behavior
AVCapturePhotoOutput allows on-demand resource allocation and reclamation
Capture clients must provide a reason for camera, mic, and photos use
More Information

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advances in iOS Photography</td>
<td>Pacific Heights</td>
<td>Tuesday 11:00AM</td>
</tr>
<tr>
<td>Live Photo Editing and RAW Processing with Core Image</td>
<td>Nob Hill</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Working With Wide Color</td>
<td>Mission</td>
<td>Thursday 1:40PM</td>
</tr>
<tr>
<td>Lab</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Photo Capture Lab</td>
<td>Graphics, Games, and Media Lab D</td>
<td>Tuesday 1:00PM</td>
</tr>
<tr>
<td>PhotoKit Lab</td>
<td>Graphics, Games, and Media Lab D</td>
<td>Tuesday 4:00PM</td>
</tr>
<tr>
<td>Color Lab</td>
<td>Frameworks Lab A</td>
<td>Wednesday 1:00PM</td>
</tr>
<tr>
<td>Photo Capture Lab</td>
<td>Graphics, Games, and Media Lab D</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Live Photo &amp; Core Image Lab</td>
<td>Graphics, Games, and Media Lab D</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>Live Photo &amp; Core Image Lab</td>
<td>Graphics, Games, and Media Lab D</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>Color Lab</td>
<td>Graphics, Games, and Media Lab C</td>
<td>Friday 4:00PM</td>
</tr>
</tbody>
</table>