AVFoundation + Core Image in Practice

Yiming Tang

#32 CocoaHeads Shanghai September 2017

About Me

Yiming Tang (唐毅明)

iOS Engineer @baixing.com

@yiming_t

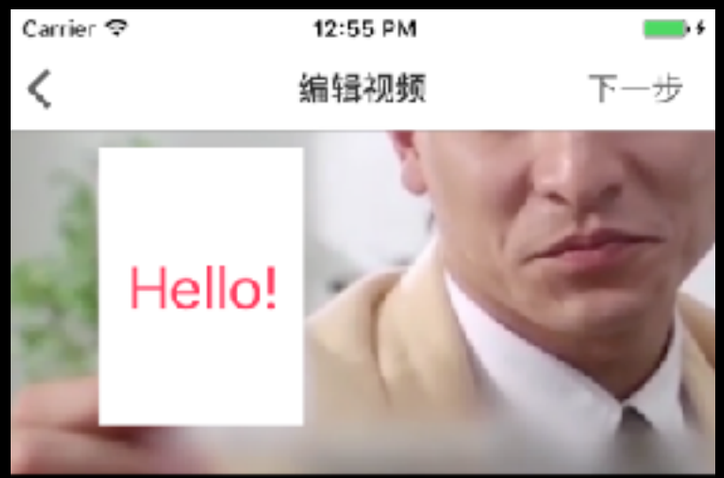
yimingtang

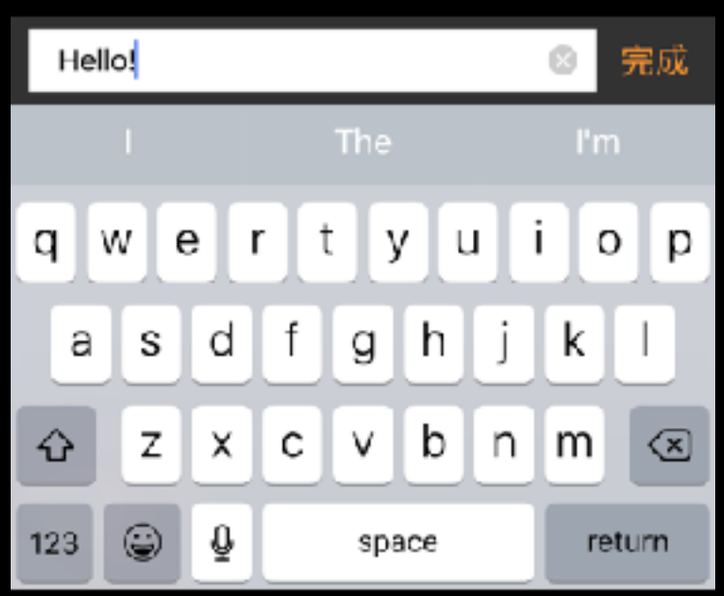
Requirements

a list of templates

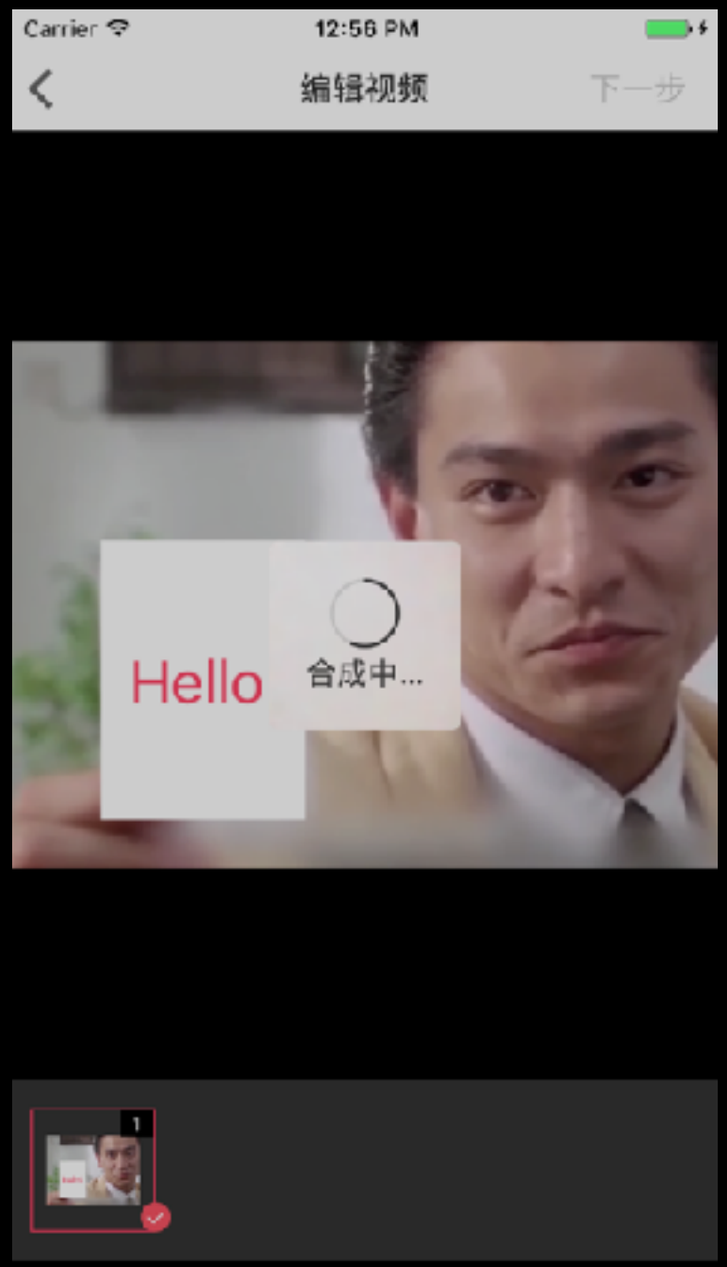
template details

Requirements



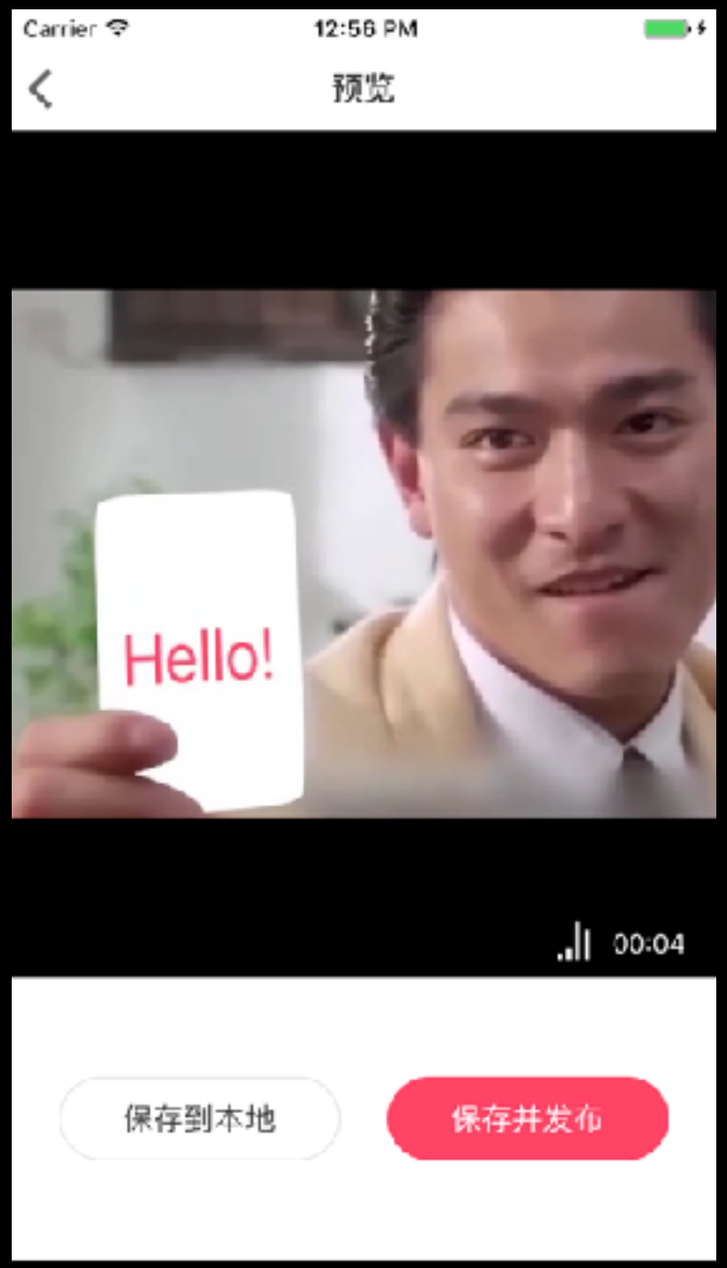


editor



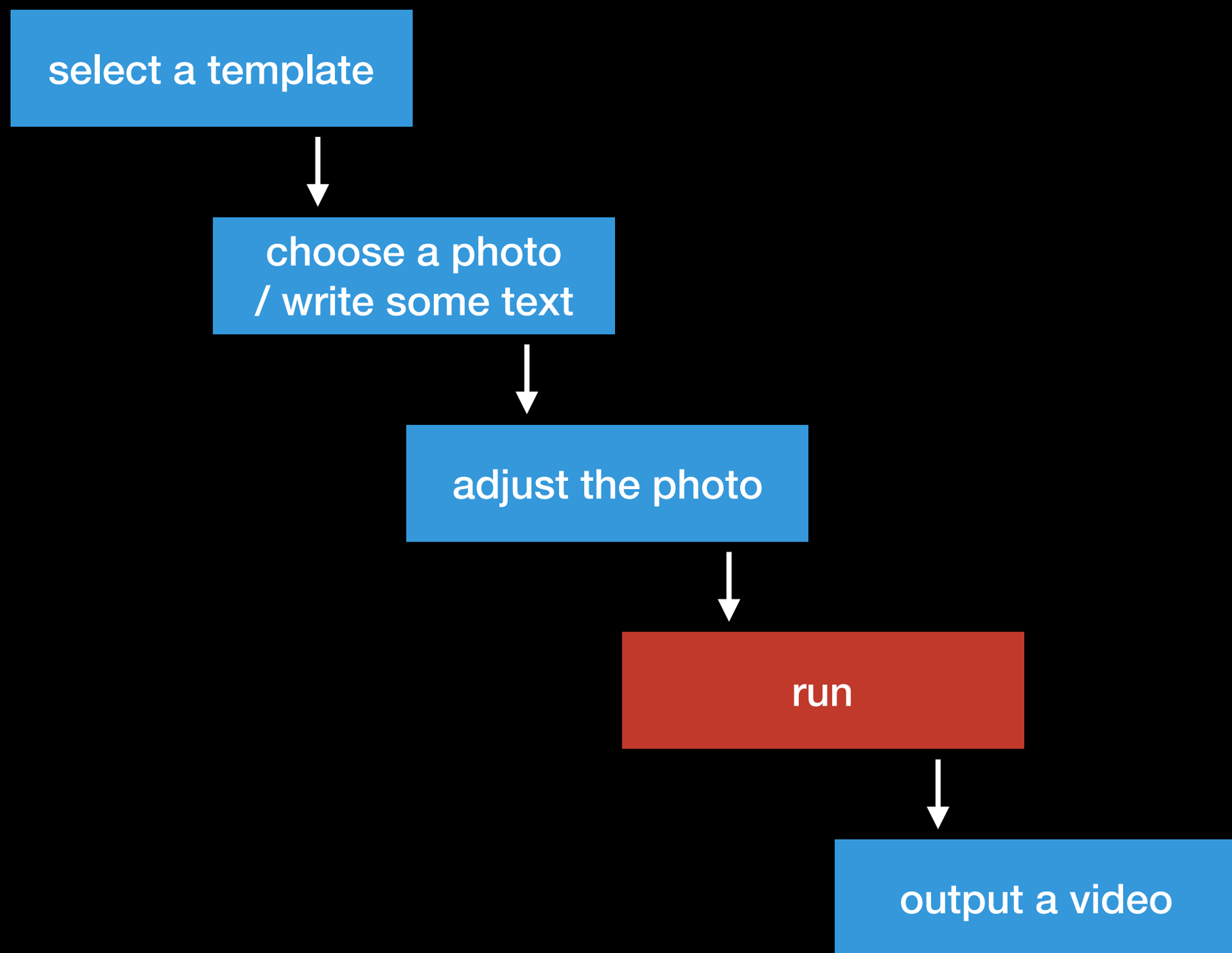
running

Requirements



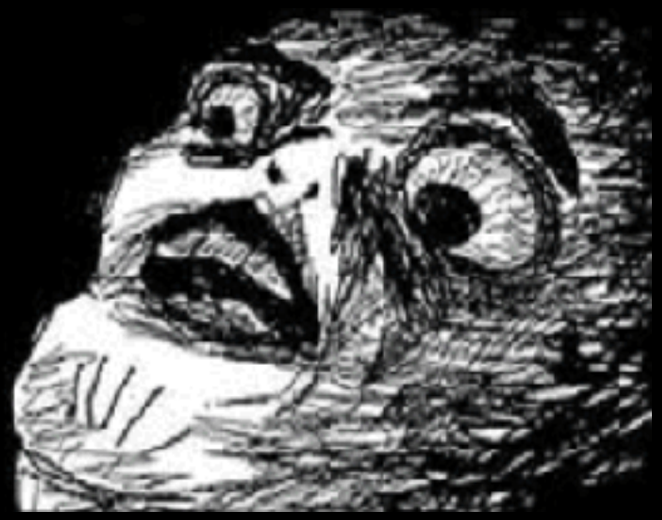
the final result

Requirements



Live Demo

It’s ***hard*** at first glance.



Narrow down
the problem.

What is a video?

“Film, also called a movie, motion picture, cinematography or photoplay is a series of still images that when shown on a screen create an illusion of motion images (due to the phi phenomenon).”

– *Wikipedia: Film*

Images + Time = Video

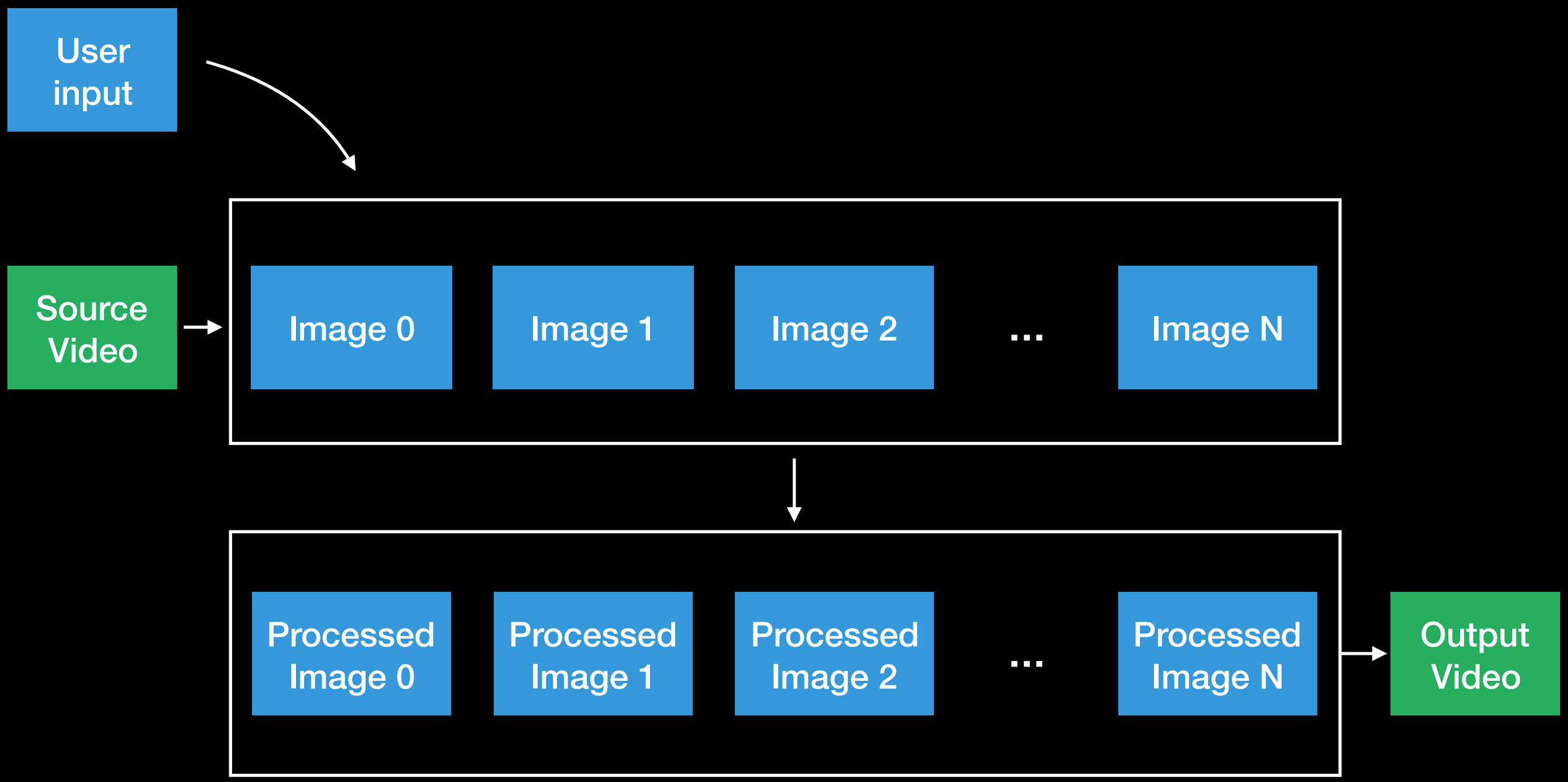
Video Problem

↓

Image Problem

Solution

- Get all images from the original video
- Process every image with user input if necessary
- Create a video from those processed images

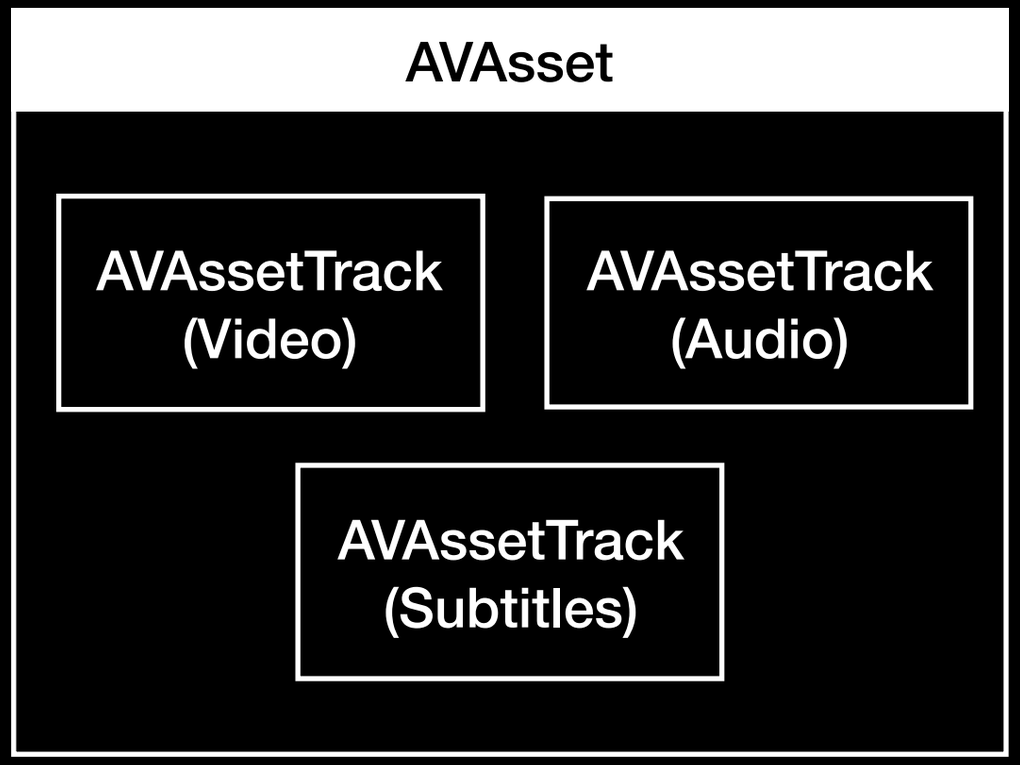

User input
Source Video
Image 0
Image 1
Image 2
...
Image N
Processed Image 0
Processed Image 1
Processed Image 2
...
Processed Image N
Output Video

How to get images from a video?

AVFoundation

Key Concepts

- AVAsset
- AVAssetTrack



Creating an Asset Object

```
NSURL *url = <#A URL that identifies an movie file#>;
AVAsset *anAsset = [AVAsset assetWithURL:maskAssetURL];
```

Preparing an Asset for Use

```
NSURL *url = <#A URL that identifies an audiovisual asset such as a movie file#>;
AVURLAsset *asset = [[AVURLAsset alloc] initWithURL:url options:nil];
NSArray *keys = @[@"tracks"];

[asset loadValuesAsynchronouslyForKeys:keys completionHandler:^{
    NSError *error = nil;
    AVKeyValueStatus tracksStatus = [asset statusOfValueForKey:@"tracks" error:&error];
    switch (tracksStatus) {
        case AVKeyValueStatusLoaded: {
            // Continue dealing with asset
            break;
        }
        case AVKeyValueStatusFailed: {
            // Report error
            break;
        }
        case AVKeyValueStatusCancelled: {
            // Do whatever is appropriate for cancelation.
            break;
        }
        case AVKeyValueStatusLoading: {
            // Loading
            break;
        }
        case AVKeyValueStatusUnknown: {
            // Unkown
            break;
        }
    }
}];
```

Generating a Sequence of Images

- `AVAssetImageGenerator`
- `AVAssetReader`

AVAssetImageGenerator

```
- (instancetype)initWithAsset:(AVAsset *)asset NS_DESIGNATED_INITIALIZER;

- (nullable CGImageRef)copyCGImageAtTime:(CMTime)requestedTime actualTime:
(nullable CMTime *)actualTime error:(NSError * _Nullable * _Nullable)outError
CF_RETURNS_RETAINED;

/* error object indicates the reason for failure if the result is
AVAssetImageGeneratorFailed */
typedef void (^AVAssetImageGeneratorCompletionHandler)(CMTime requestedTime,
CGImageRef _Nullable image, CMTime actualTime, AVAssetImageGeneratorResult
result, NSError * _Nullable error);

- (void)generateCGImagesAsynchronouslyForTimes:(NSArray<NSValue *>
*)requestedTimes completionHandler:
(AVAssetImageGeneratorCompletionHandler)handler;

- (void)cancelAllCGImageGeneration;
```

`AVAssetImageGenerator`

is

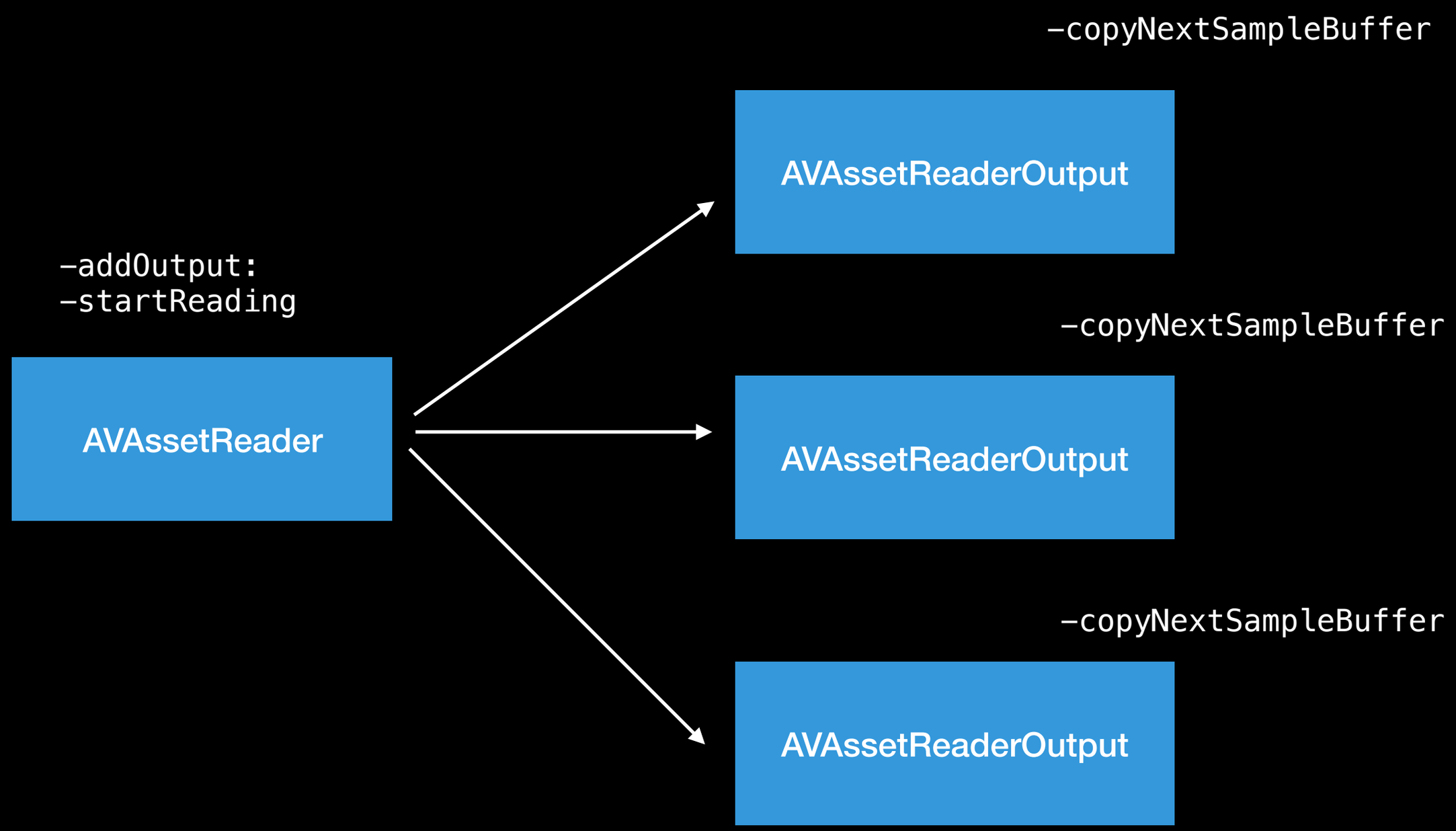
slow!

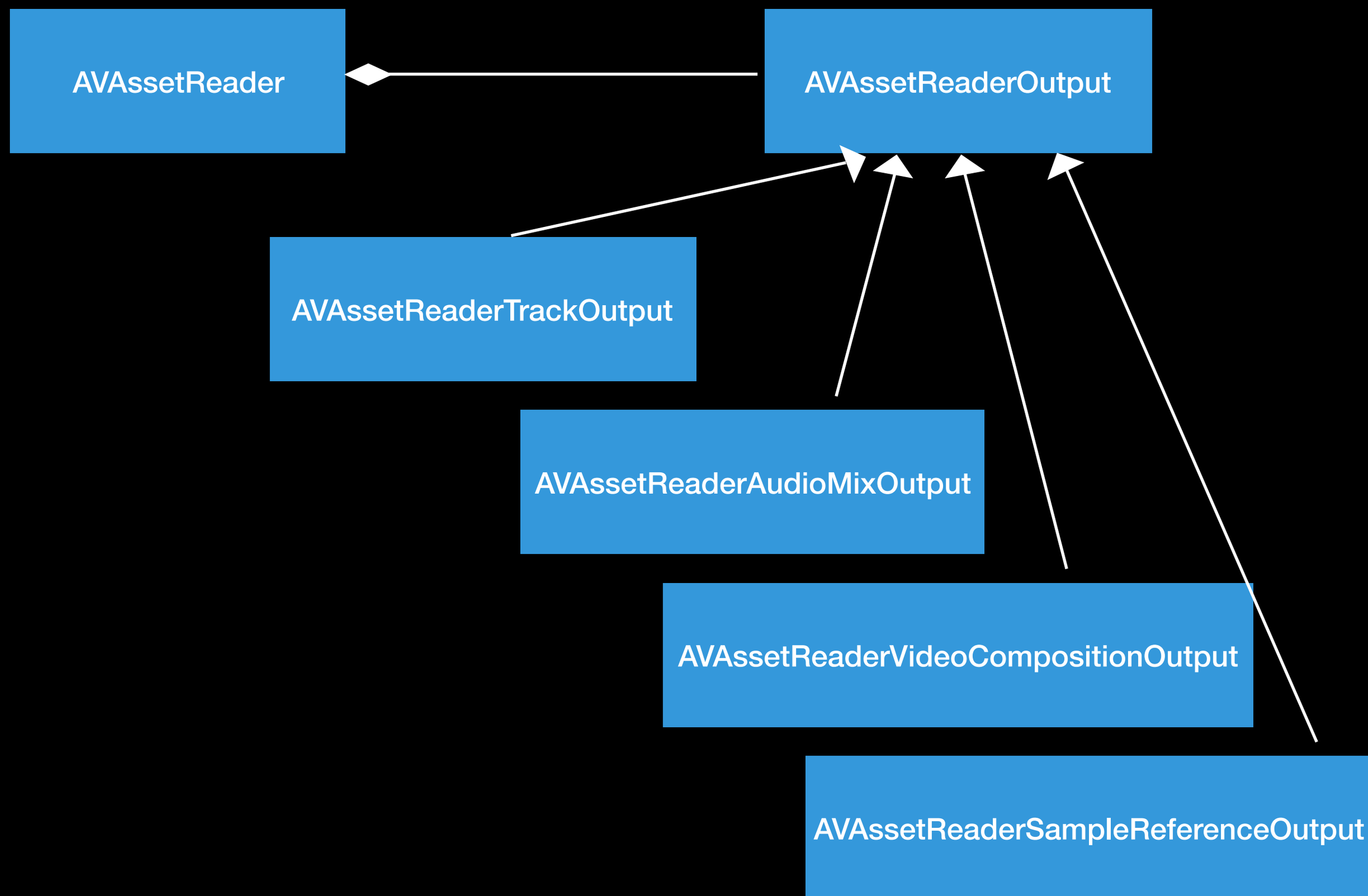
AVAssetReader

```
- (nullable instancetype)initWithAsset:(AVAsset *)asset error:(NSError *
_Nullable * _Nullable)outError NS_DESIGNATED_INITIALIZER;

@property (nonatomic, retain, readonly) AVAsset *asset;
@property (readonly) AVAssetReaderStatus status;
@property (nonatomic, readonly) NSArray<AVAssetReaderOutput *> *outputs;

- (BOOL)canAddOutput:(AVAssetReaderOutput *)output;
- (void)addOutput:(AVAssetReaderOutput *)output;
- (BOOL)startReading;
- (void)cancelReading;
```

AVAssetReaderOutput

```
- (nullable CMSampleBufferRef)copyNextSampleBuffer CF_RETURNS_RETAINED;
```


-copyNextSampleBuffer
AVAssetReaderOutput
-addOutput:
-startReading
-copyNextSampleBuffer
AVAssetReader
AVAssetReaderOutput
-copyNextSampleBuffer
AVAssetReaderOutput


AVAssetReader
AVAssetReaderOutput
AVAssetReaderTrackOutput
AVAssetReaderAudioMixOutput
AVAssetReaderVideoCompositionOutput
AVAssetReaderSampleReferenceOutput

Setting up Asset Reader

```
self.assetReader = [[AVAssetReader alloc] initWithAsset:self.asset error:error];
if (!self.assetReader || *error) {
    return NO;
}

AVAssetTrack *assetVideoTrack = [[self.asset tracksWithMediaType:AVMediaTypeVideo]
firstObject];

NSDictionary *decompressionVideoSettings = @{
    (id)kCVPixelBufferPixelFormatTypeKey : @(kCVPixelFormatType_32BGRA),
    (id)kCVPixelBufferIOSurfacePropertiesKey : [NSDictionary dictionary]
};

self.assetReaderVideoOutput = [AVAssetReaderTrackOutput
assetReaderTrackOutputWithTrack:assetVideoTrack
outputSettings:decompressionVideoSettings];

if ([self.assetReader canAddOutput:self.assetReaderVideoOutput]) {
    [self.assetReader addOutput:self.assetReaderVideoOutput];
}
```

Reading an Asset

```
// Start the asset reader up.
[self.sourceAssetReader startReading];

BOOL done = NO;
while (!done) {
    // Copy the next sample buffer from the reader output.
    CMSampleBufferRef sampleBuffer = [self.assetReaderOutput copyNextSampleBuffer];
    if (sampleBuffer) {
        // Do something with sampleBuffer here.
        CFRelease(sampleBuffer);
        sampleBuffer = NULL;
    } else {
        // Find out why the asset reader output couldn't copy another sample buffer.
        if (self.sourceAssetReader.status == AVAssetReaderStatusFailed) {
            NSError *failureError = self.sourceAssetReader.error;
            // Handle the error here.
        } else {
            // The asset reader output has read all of its samples.
            done = YES;
        }
    }
}
```

- CMSampleBufferRef -> CVImageBufferRef

```
> CMSampleBufferGetImageBuffer()
```

- CVImageBufferRef -> CIImage

```
> [CIImage imageWithCVPixelBuffer:]
```

We get all images!

Processing Images

original image
+
user input image
-> target image

original frame

+

user input

=

output frame

How?

Core Image

Basic Usage

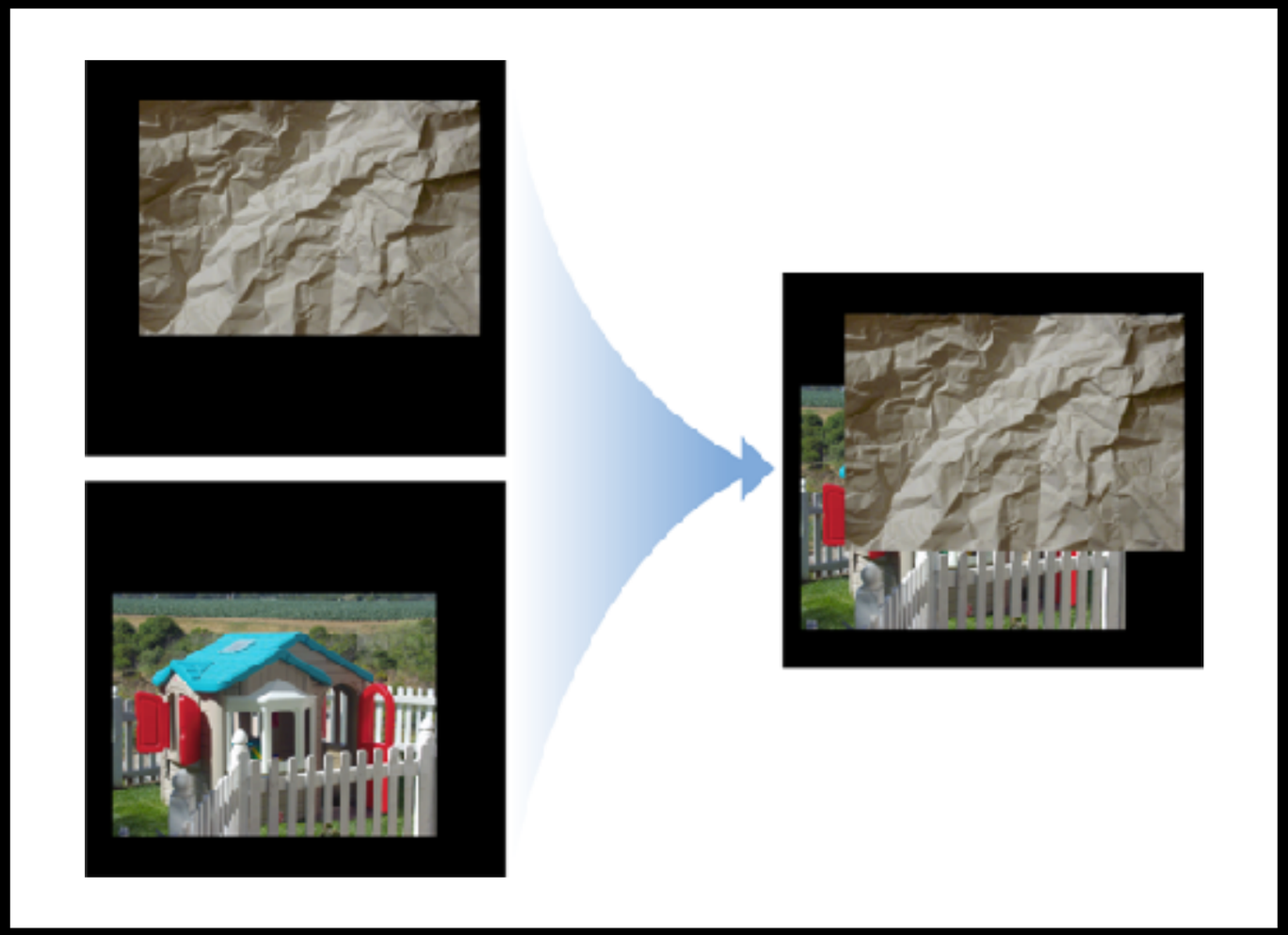
```
CIContext *context = [CIContext contextWithOptions:nil];
CIImage *image = <# An CIImage>;
CIFilter *filter = [CIFilter filterWithName:@"aFilterName"];
[filter setValue:image forKey:kCIInputImageKey];
// Set other key-value pairs

// Render
[context createCGImage:filter.outputImage fromRect:aRect];
```

Filters

iOS built-in Core Image filters are awesome!

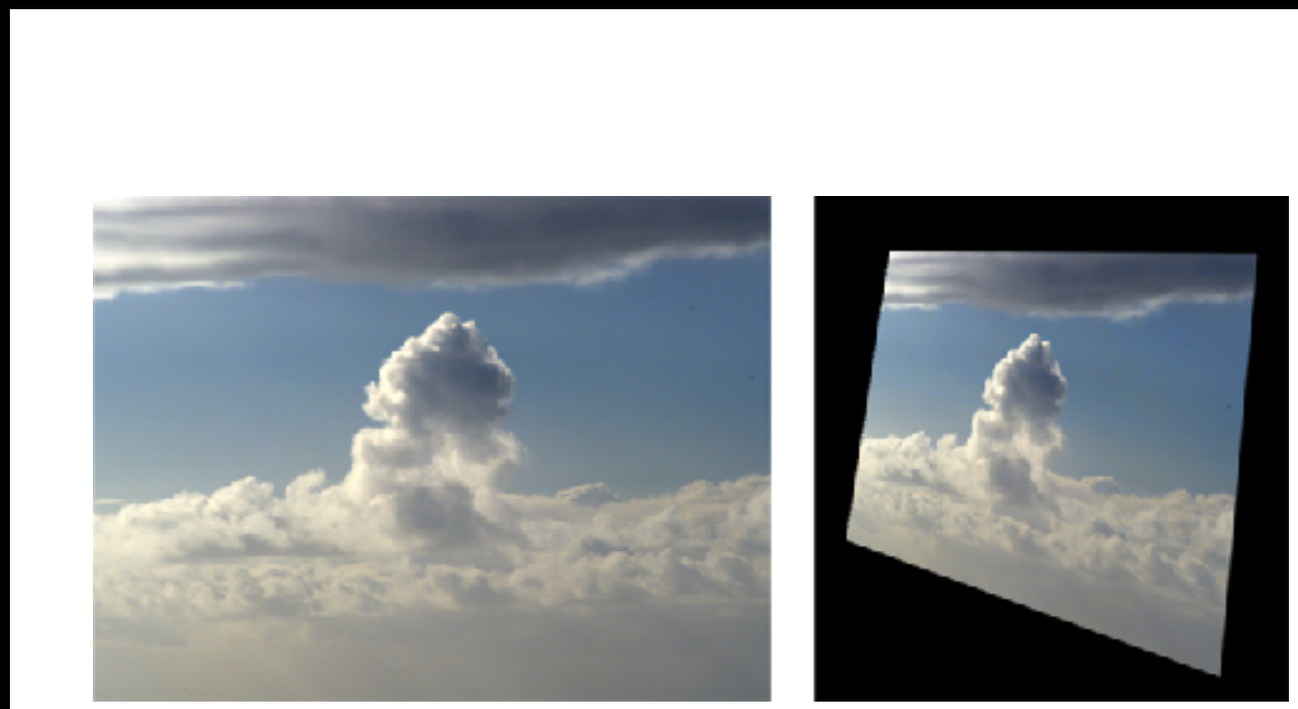
CISourceOverCompositing

- inputImage
- inputBackgroundImage



CIPerspectiveTransform

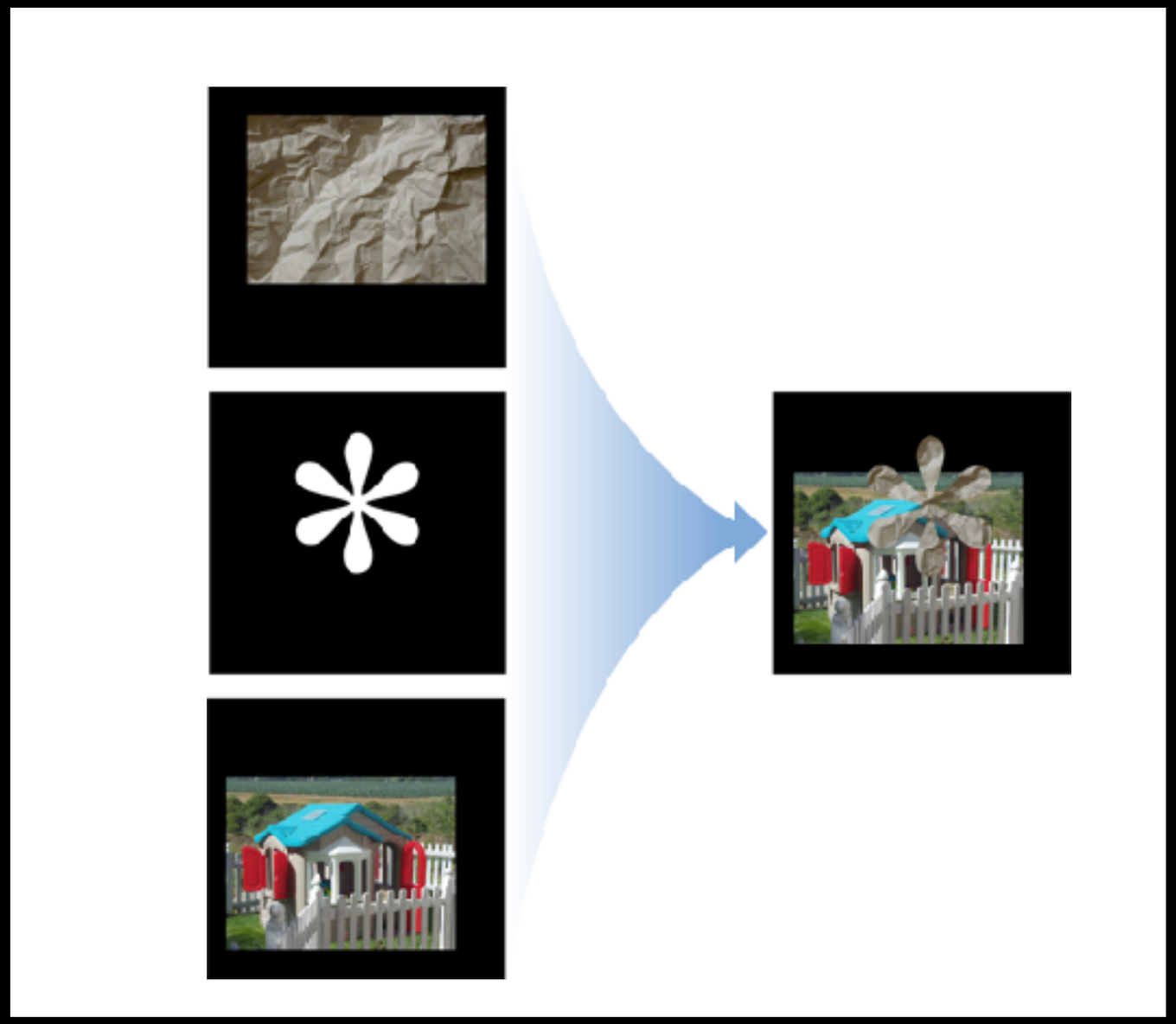
- inputImage
- inputTopLeft
- inputTopRight
- inputBottomRight
- inputBottomLeft



How to deal with “hands”?

CIBlendWithMask

- inputImage
- inputBackgroundImage
- inputMaskImage



A key frame mask

- How to get quadrilateral vertices?
- How to get masks?

After Effects + mocha AE

Creating a video from images

AVAssetWriter

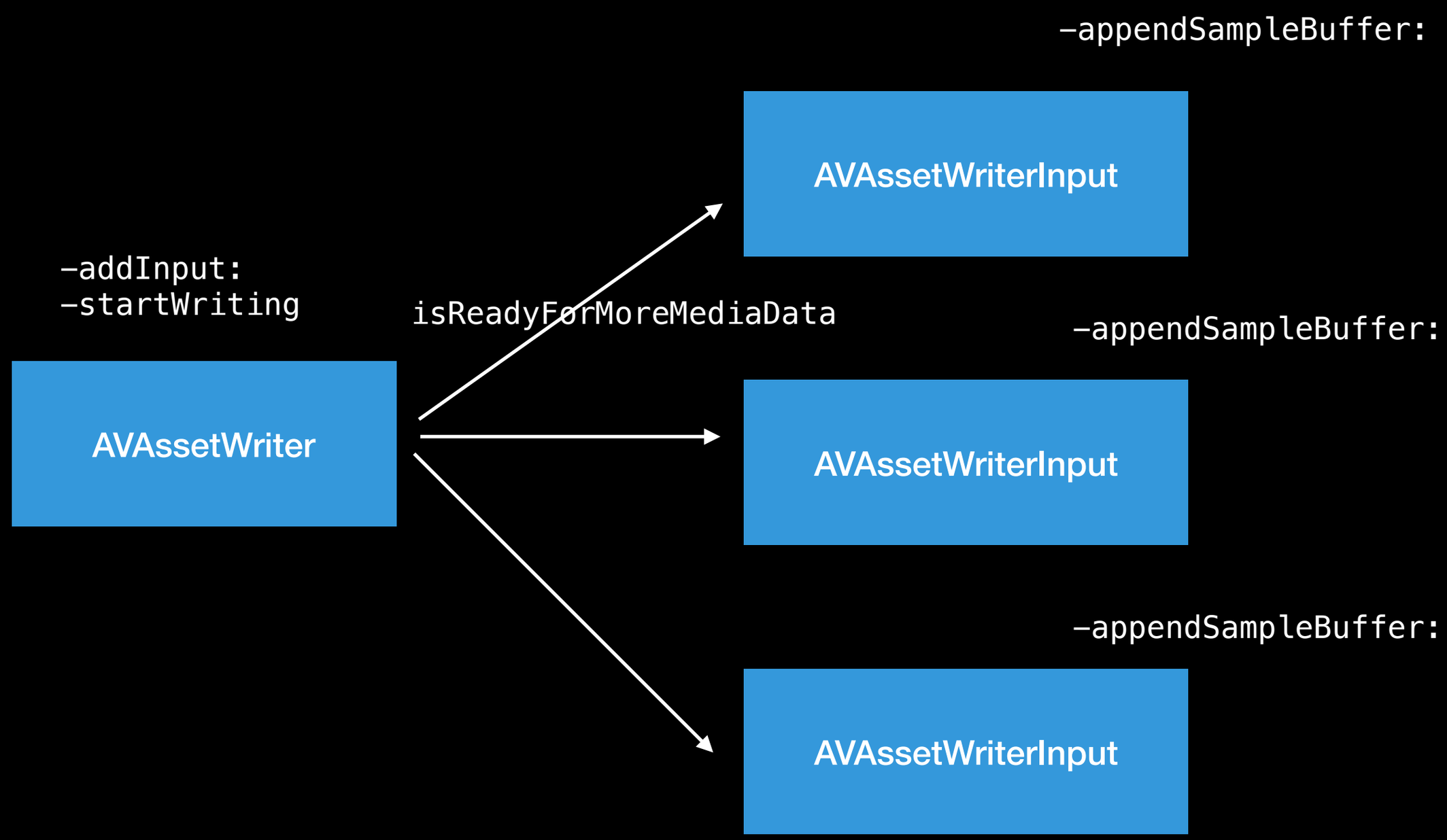
```
- (nullable instancetype)initWithURL:(NSURL *)outputURL fileType:(NSString
*)outputFileType error:(NSError * _Nullable * _Nullable)outError
NS_DESIGNATED_INITIALIZER;

@property (nonatomic, copy, readonly) NSURL *outputURL;
@property (readonly) AVAssetWriterStatus status;
@property (readonly, nullable) NSError *error;
@property (nonatomic, readonly) NSArray<AVAssetWriterInput *> *inputs;

- (BOOL)canAddInput:(AVAssetWriterInput *)input;
- (void)addInput:(AVAssetWriterInput *)input;

- (BOOL)startWriting;
- (void)startSessionAtSourceTime:(CMTime)startTime;
- (void)endSessionAtSourceTime:(CMTime)endTime;
- (void)cancelWriting;
- (void)finishWritingWithCompletionHandler:(void (^)(void))handler;
```

AVAssetWriterInput

```
- (instancetype)initWithMediaType:(NSString *)mediaType outputSettings:
(nullable NSDictionary<NSString *, id> *)outputSettings sourceFormatHint:
(nullable CMFormatDescriptionRef)sourceFormatHint NS_AVAILABLE(10_8, 6_0)
NS_DESIGNATED_INITIALIZER;

@property (nonatomic, readonly, nullable) NSDictionary<NSString *, id>
*outputSettings;
@property (nonatomic, readonly, getter=isReadyForMoreMediaData) BOOL
readyForMoreMediaData;

- (void)requestMediaDataWhenReadyOnQueue:(dispatch_queue_t)queue usingBlock:
(void (^)(void))block;
- (BOOL)appendSampleBuffer:(CMSampleBufferRef)sampleBuffer;
- (void)markAsFinished;
```

Sample Usage

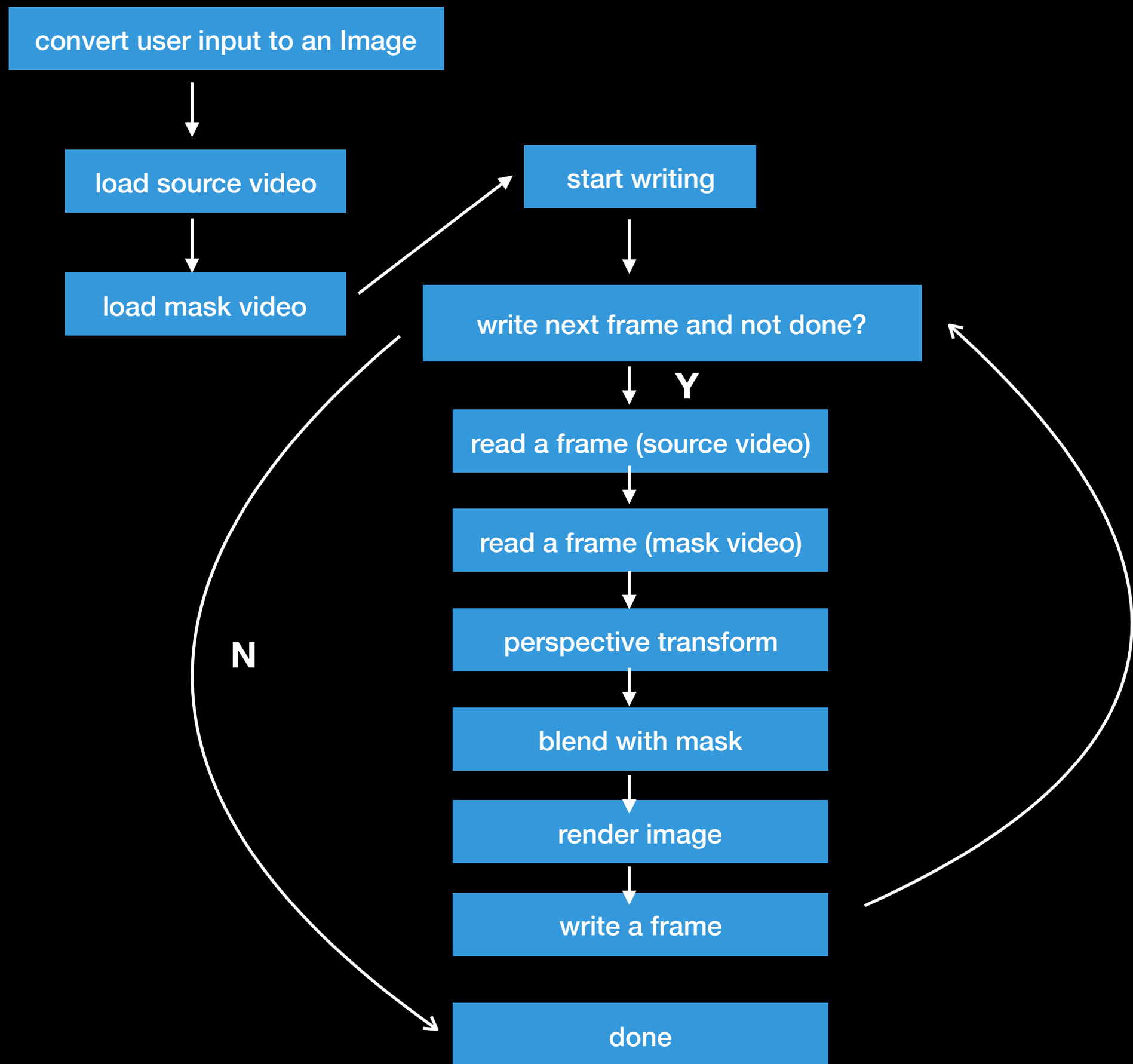
```
// Prepare the asset writer for writing.
[self.assetWriter startWriting];
// Start a sample-writing session.
[self.assetWriter startSessionAtSourceTime:kCMTimeZero];
// Specify the block to execute when the asset writer is ready for media data and the queue
to call it on.
[self.assetWriterInput requestMediaDataWhenReadyOnQueue:myInputSerialQueue usingBlock:^{
    while ([self.assetWriterInput isReadyForMoreMediaData]){
        // Get the next sample buffer.
        CMSampleBufferRef nextSampleBuffer = [self.assetReaderOutput copyNextSampleBuffer];
        if (nextSampleBuffer) {
            // If it exists, append the next sample buffer to the output file.
            [self.assetWriterInput appendSampleBuffer:nextSampleBuffer];
            CFRelease(nextSampleBuffer);
            nextSampleBuffer = NULL;
        } else {
            // Assume that lack of a next sample buffer means the sample buffer source is
out of samples and mark the input as finished.
            [self.assetWriterInput markAsFinished];
            break;
        }
    }
}];
```


-appendSampleBuffer:
AVAssetWriterInput
-addInput:
-startWriting
isReadyForMoreMediaData
-appendSampleBuffer:
AVAssetWriter
AVAssetWriterInput
-appendSampleBuffer:
AVAssetWriterInput

AVAssetWriterInputPixelBufferAdaptor

```
// We've got CIFilter output as a CIImage object
// Render it to CVPixelBuffer and write that to video

CVPixelBufferRef renderedOutputPixelBuffer = NULL;
CVReturn error = CVPixelBufferPoolCreatePixelBuffer(NULL,
self.assetWriterInputPixelBufferAdaptor.pixelBufferPool, &renderedOutputPixelBuffer);

if (!error) {
    if (filteredImage) {
        [self.ciContext render:filteredImage toCVPixelBuffer:renderedOutputPixelBuffer];
        [self.assetWriterInputPixelBufferAdaptor
appendPixelBuffer:renderedOutputPixelBuffer
withPresentationTime:CMSampleBufferGetOutputPresentationTimeStamp(sourceSampleBuffer)];
}
```

Wrap-up


convert user input to an Image
load source video
load mask video
start writing
write next frame and not done?
Y
N
read a frame (source video)
read a frame (mask video)
perspective transform
blend with mask
render image
write a frame
done

What I learned

- Don’t be afraid of new problems. Narrow down the problem: complicated -> simple
- Keep learning.

References

- **AVFoundation Programming Guide**: https://developer.apple.com/library/content/documentation/AudioVideo/Conceptual/AVFoundationPG/Articles/00_Introduction.html
- **Core Image Filter Reference:** https://developer.apple.com/library/content/documentation/GraphicsImaging/Reference/CoreImageFilterReference/index.html
- **Motion tracking overview and resources**: https://helpx.adobe.com/after-effects/using/tracking-stabilizing-motion-cs5.html

Q & A

Thank you!

More programming articles
WeChat：百姓网技术团队

Yiming Tang (唐毅明)

iOS Engineer @baixing.com

@yiming_t

yimingtang