

# Pursuing 60 fps scrolling

Tamás Zahola  
[github.com/tzahola](https://github.com/tzahola)

## Our task

Write a business card  
browser for iPad



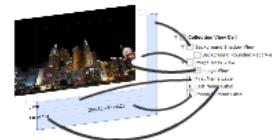
Thank you!  
Q&A

# Pursuing 60 fps scrolling

Tamás Zahola  
[github.com/tzahola](https://github.com/tzahola)

# Our task

Write a business card  
browser for iPad





```

- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView
    cellForItemAtIndexPath:(NSIndexPath *)indexPath {
    Cell* cell = [collectionView
        dequeueReusableCellWithReuseIdentifier:NSStringFromClass([Cell class])
        forIndexPath:indexPath];
    cell.data = self.data[indexPath.row];
    return cell;
}

@interface Cell : UICollectionViewCell
@property (weak, nonatomic) IBOutlet UIView *backgroundRoundedRectView;
@property (weak, nonatomic) IBOutlet UIView *backgroundShadowView;

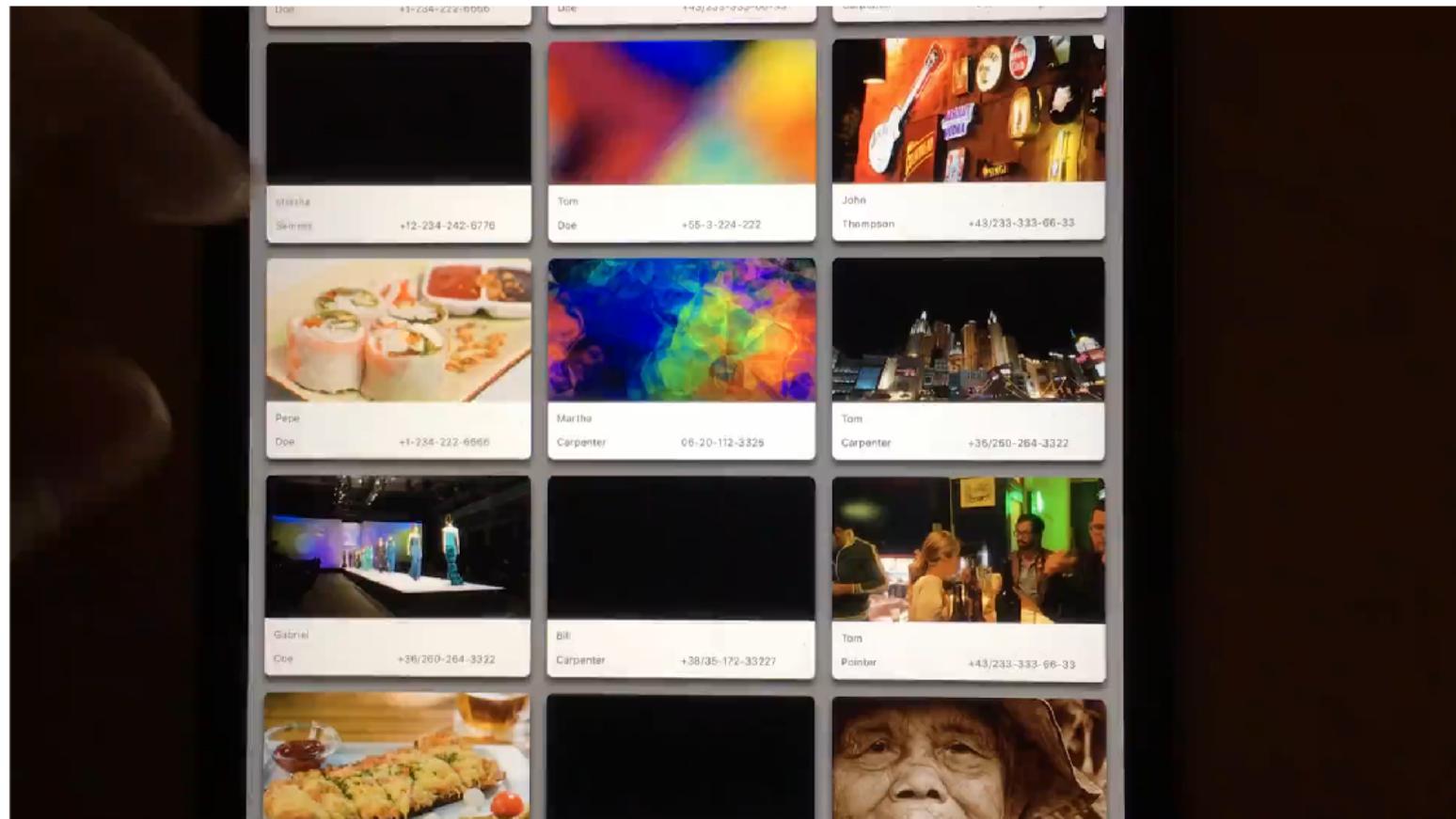
@property (weak, nonatomic) IBOutlet UIImageView *imageViewMaskView;
@property (weak, nonatomic) IBOutlet UIImageView *imageView;
@property (weak, nonatomic) IBOutlet UILabel *firstNameLabel;
@property (weak, nonatomic) IBOutlet UILabel *lastNameLabel;
@property (weak, nonatomic) IBOutlet UILabel *phoneNumberLabel;

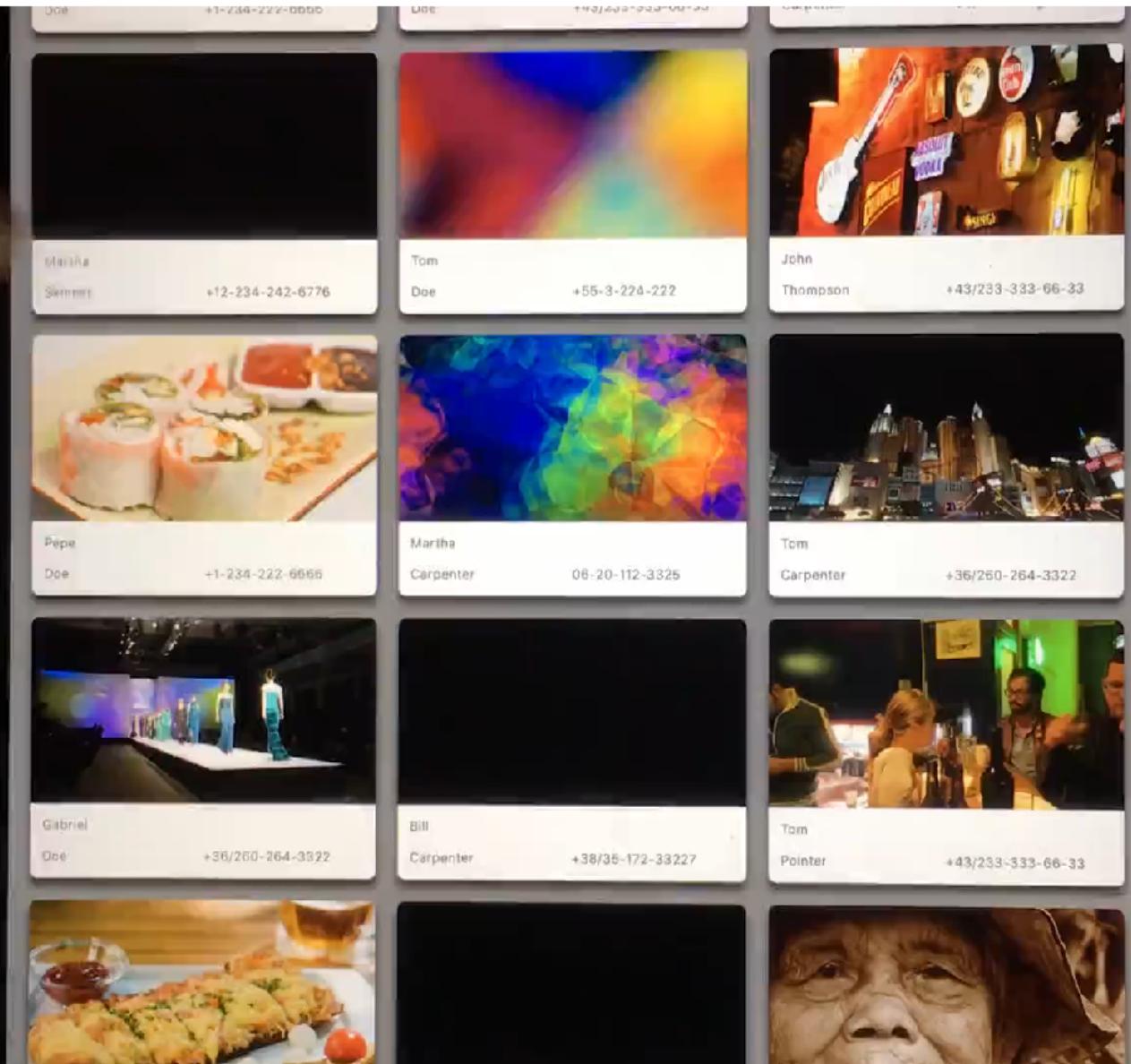
@property (nonatomic, strong) Data* data;
@end

@implementation Cell
- (void)setData:(Data *)data {
    if (_data == data) {
        return;
    }
    _data = data;
    self.firstNameLabel.text = data.firstName;
    self.lastNameLabel.text = data.lastName;
    self.phoneNumberLabel.text = data.phoneNumber;
    self.imageView.image = [UIImage imageWithData:[NSData
        dataWithContentsOfURL:data.imageURL]];
}
- (void)prepareForReuse {
    [super prepareForReuse];
    self.data = nil;
}
- (void)awakeFromNib {
    [super awakeFromNib];
    self.imageViewMaskView.layer.masksToBounds = YES;
    self.imageViewMaskView.layer.cornerRadius = 5;
    self.backgroundRoundedRectView.backgroundColor = [UIColor whiteColor];
    self.backgroundRoundedRectView.layer.masksToBounds = YES;
    self.backgroundRoundedRectView.layer.cornerRadius = 5;
    self.backgroundShadowView.layer.shadowColor = [UIColor blackColor].CGColor;
    self.backgroundShadowView.layer.shadowOffset = CGSizeMake(0, 5);
    self.backgroundShadowView.layer.shadowRadius = 3;
    self.backgroundShadowView.layer.shadowOpacity = 0.7;
}
@end

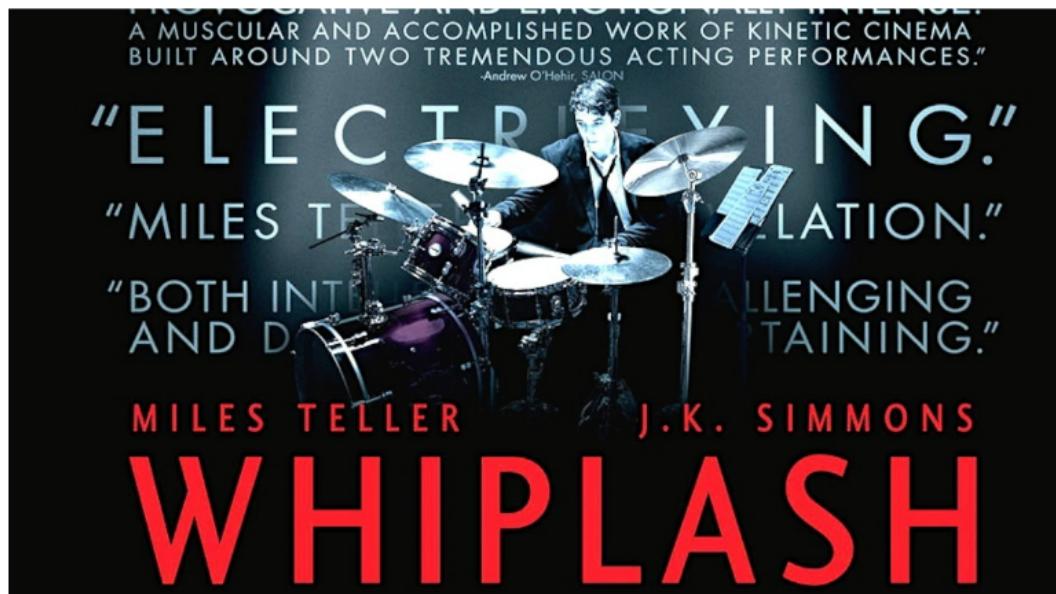
```

# The easy fix





# Is this good enough?



*"There are no two words in the English language more harmful than 'good job'."* – Terence Fletcher

*"There are no two words in software development more harmful than 'good enough'."* – Yours truly :P

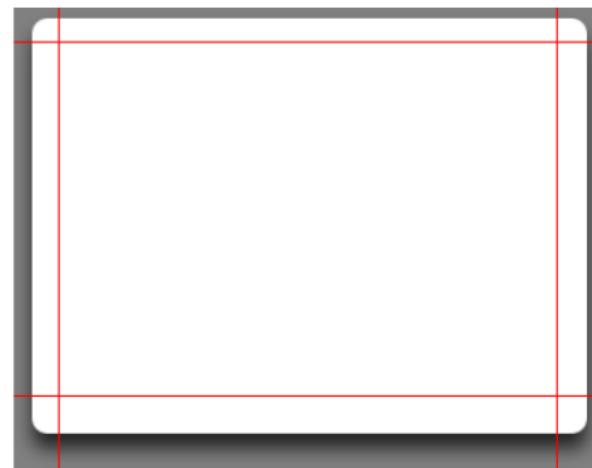
# **Let's see what we can do**

With **traditional tools**

(i.e. don't use a cannon to kill a fly. hint: cannon = AsyncDisplayKit)

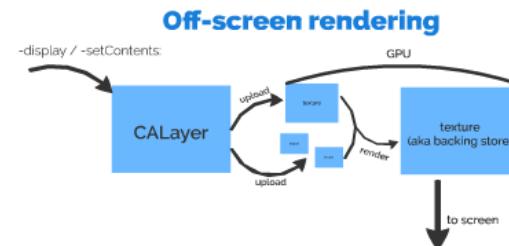
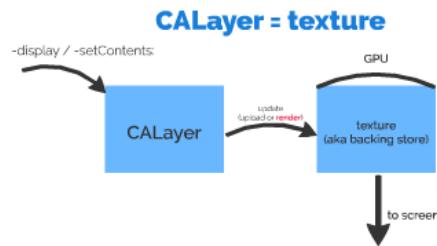
# Forget CALayer shadow

Even with `layer.shouldRasterize = YES` it cannot beat the good old 9-patch image.



The solid background makes alpha bending obsolete too.

# Eliminate off-screen rendering

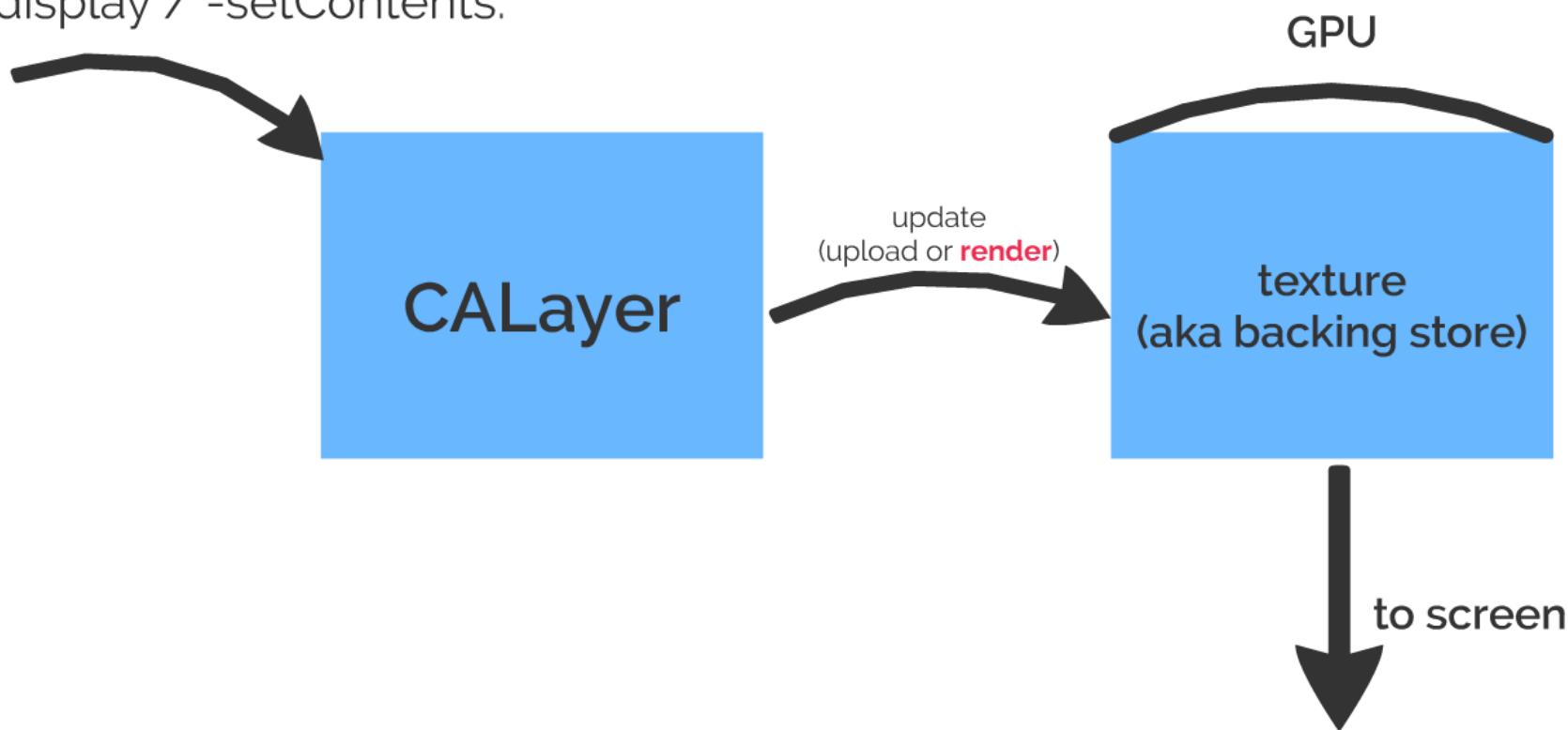


## What triggers off-screen rendering?

- layer.masksToBounds = YES
- layer.shouldRasterize = YES
- layer.shadow...
- layer.mask = ...

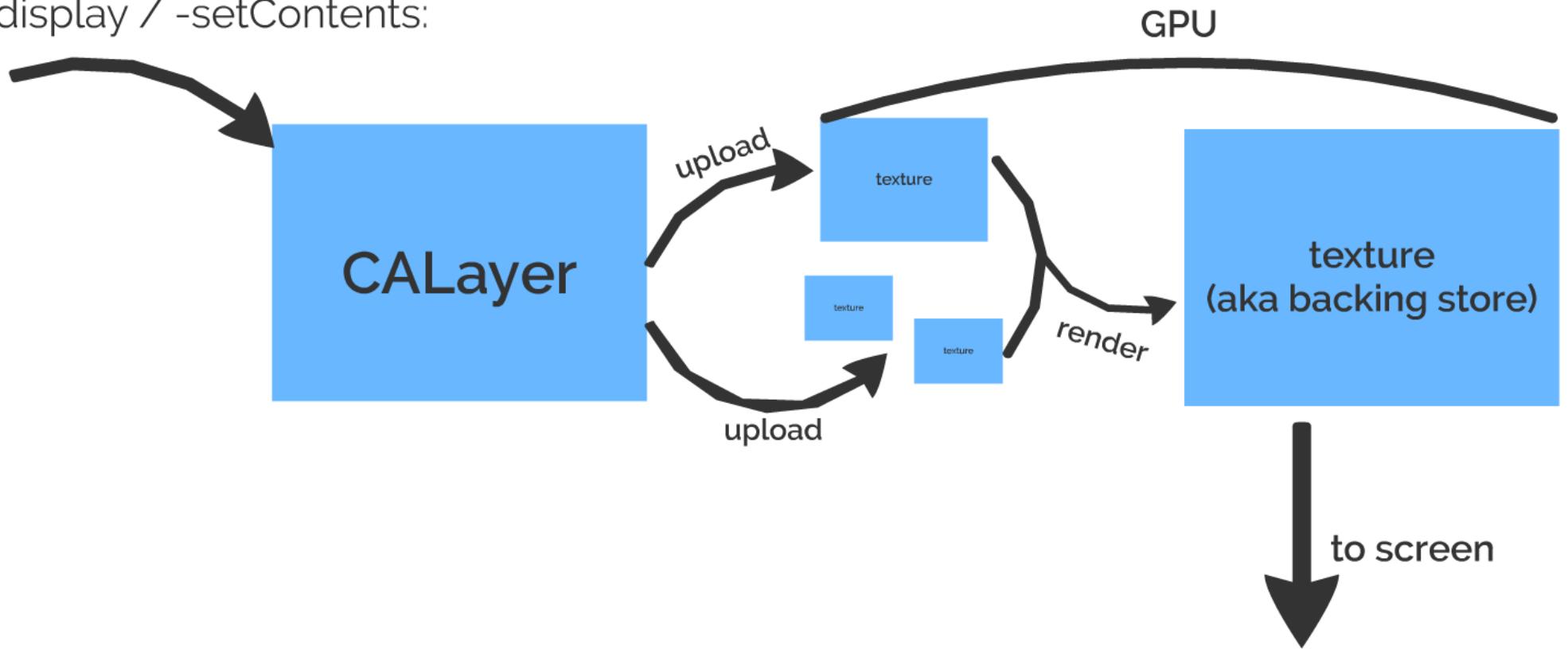
# CALayer = texture

-display / -setContents:



# Off-screen rendering

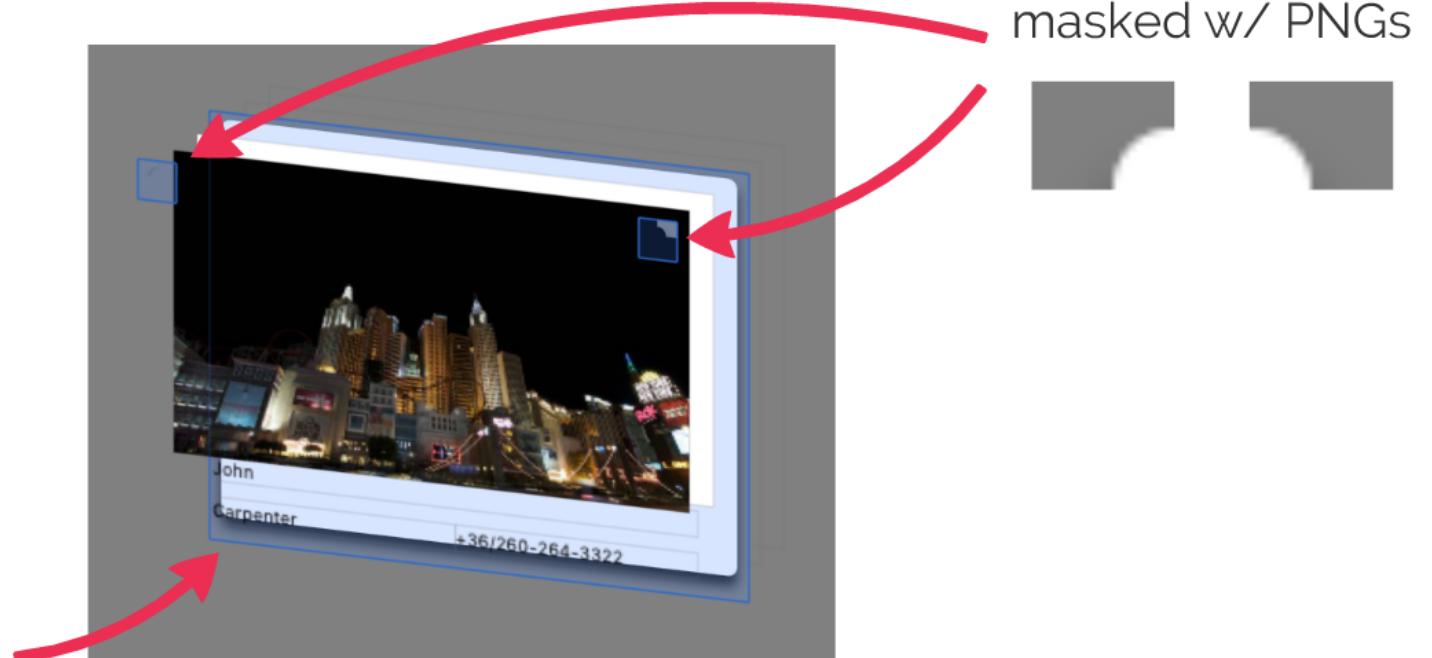
-display / -setContents:



## What triggers off-screen rendering?

- `layer.masksToBounds = YES`
- `layer.shouldRasterize = YES`
- `layer.shadow...`
- `layer.mask = ...`

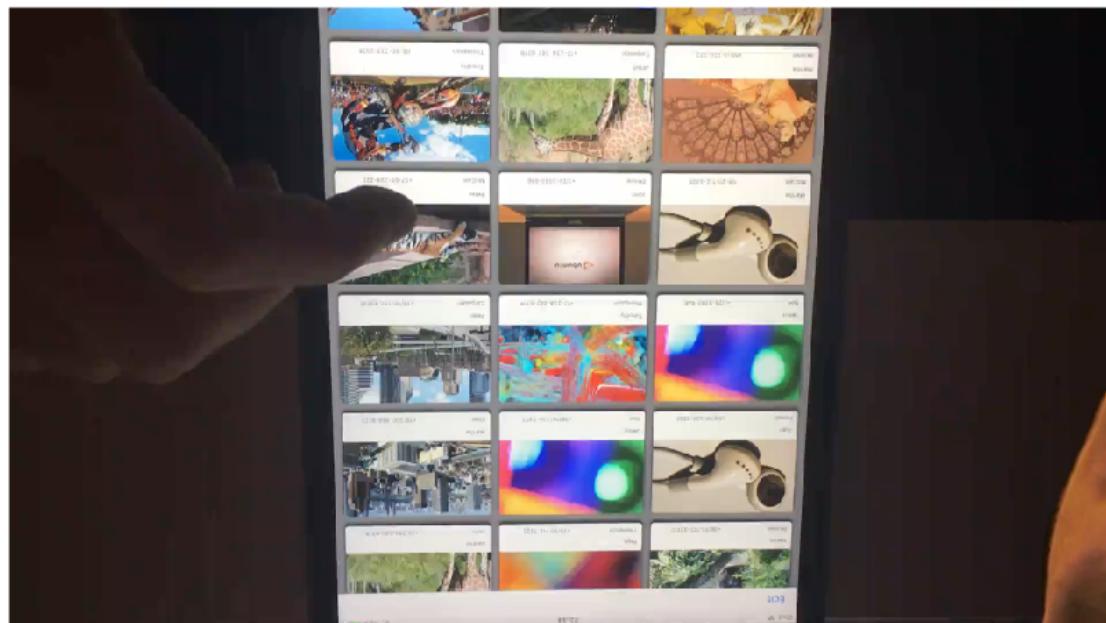
# Baked-in effects



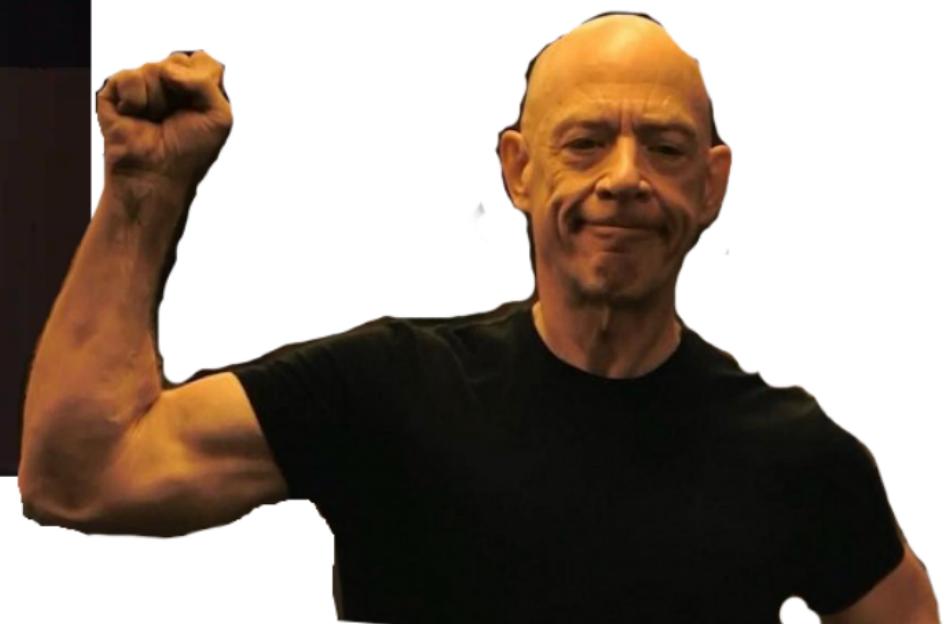
baked-in shadow & rounded-rect background  
via 9-patch image

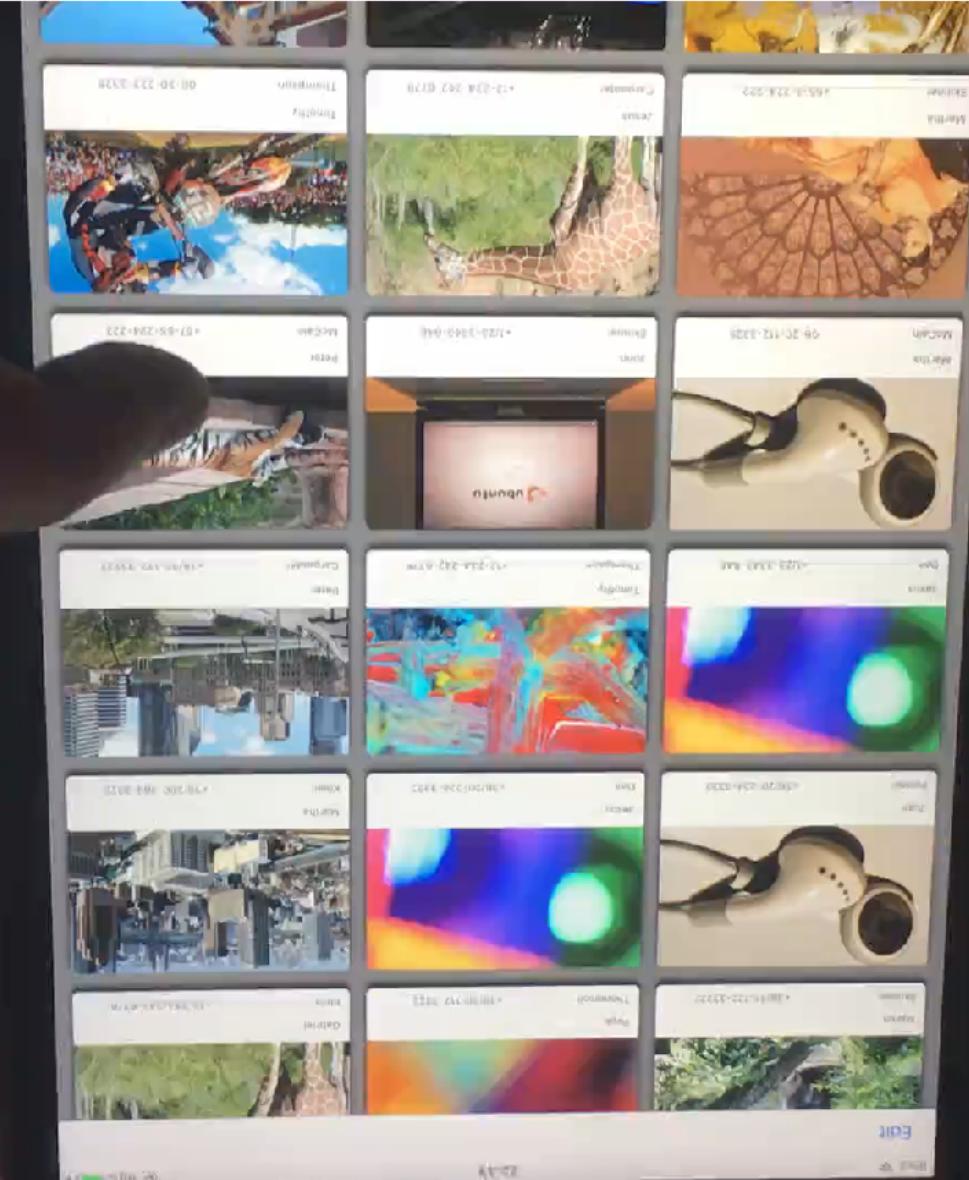
image corners  
masked w/ PNGs

# Are we finished yet?

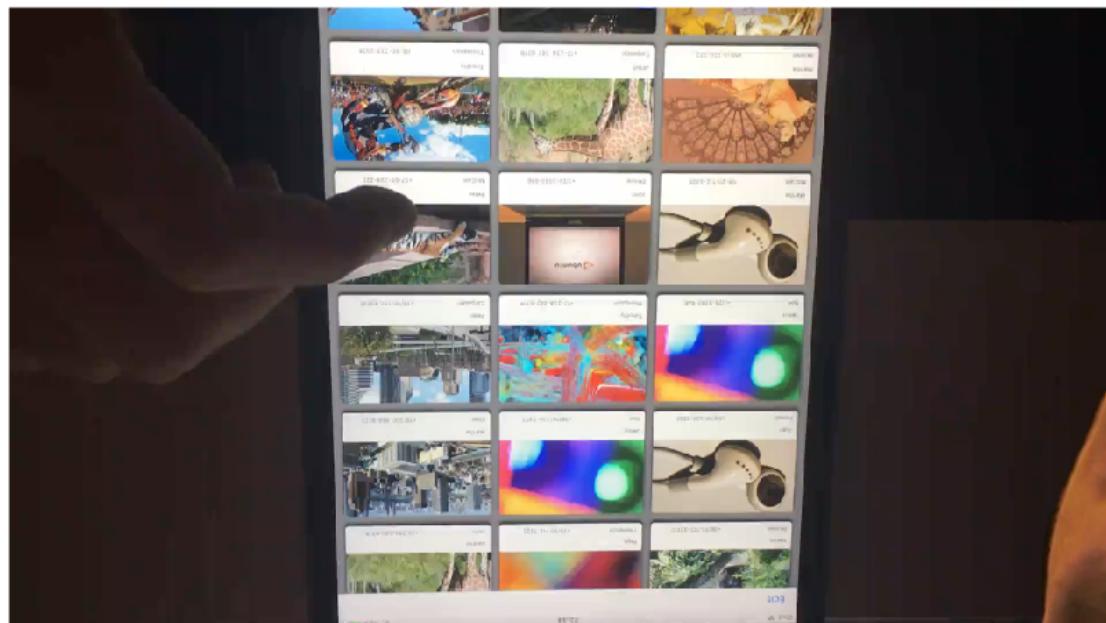


not quite...

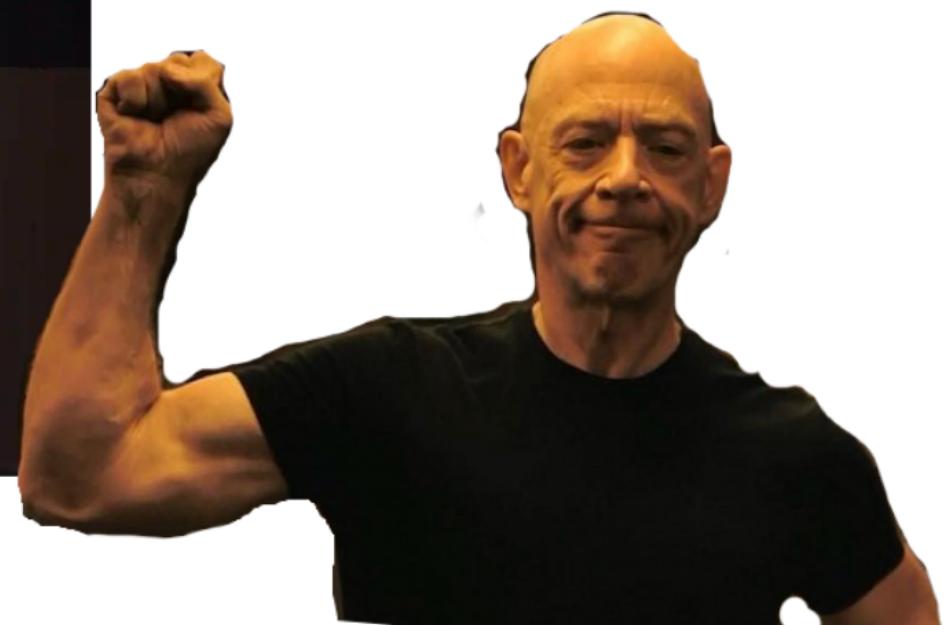




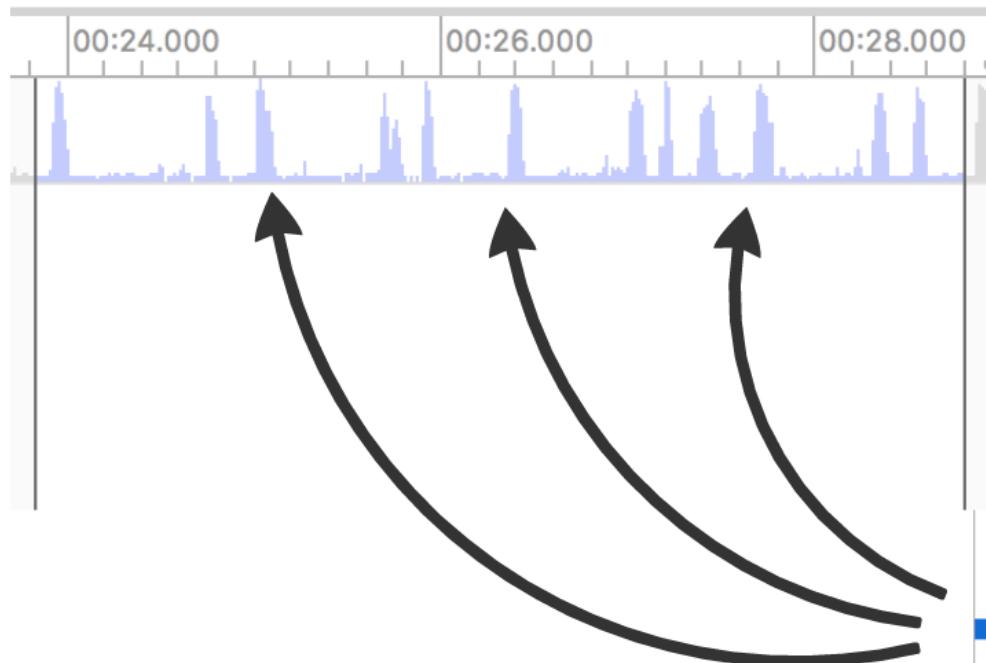
# Are we finished yet?



not quite...



# Let's profile!



But we didn't ask for re-layout!

20.0ms	100.0%	0.0	▼-[CollectionView collectionView:cellForItemAtIndexPath:] Scrolling60fps
17.0ms	85.0%	0.0	▼-[UICollectionView dequeueReusableCellWithReuseIdentifier:forIndexPath:] UIKit
17.0ms	85.0%	0.0	▼-[UICollectionView _dequeueReusableViewOfKind:withIdentifier:forIndexPath:viewCategory:] UIKit
10.0ms	50.0%	0.0	▼-[Cell prepareForReuse] Scrolling60fps
10.0ms	50.0%	0.0	▼-[Cell setData:] Scrolling60fps
8.0ms	40.0%	0.0	▼-[UILabel _setText:] UIKit
5.0ms	25.0%	0.0	▶-[UIView(AdditionalLayoutSupport) _invalidateIntrinsicContentSizeNeedingLayout:] UIKit
2.0ms	10.0%	0.0	▶-[UIView(Rendering) setNeedsDisplay] UIKit
1.0ms	5.0%	0.0	▶-[NSClassFromString Foundation
1.0ms	5.0%	0.0	▶-[Cell lastNameLabel] Scrolling60fps
1.0ms	5.0%	1.0	objc_msgSend libobjc.A.dylib
6.0ms	30.0%	0.0	▶+[UIView(Animation) performWithoutAnimation:] UIKit
1.0ms	5.0%	0.0	▶-[NSDictionaryM objectForKey:] CoreFoundation
3.0ms	15.0%	0.0	▼-[Cell setData:] Scrolling60fps
1.0ms	5.0%	0.0	▶-dispatch_async libdispatch.dylib
1.0ms	5.0%	0.0	▶-[UILabel _setText:] UIKit
1.0ms	5.0%	1.0	objc_msgSend libobjc.A.dylib

**UILabel :(**



**CATextLayer ;)**

# Still not quite right...

On the main thread???

2395.0ms	53.1%	0,0	▼Main Thread 0x929ef	⊕
953.0ms	21.1%	953,0	▼0x1846578d0	libz.1.dylib
894.0ms	19.8%	0,0	▼0x1846578d0	libz.1.dylib
894.0ms	19.8%	0,0	►png_read_IDAT_dataApple	ImageIO
59.0ms	1.3%	0,0	►inflate	libz.1.dylib
252.0ms	5.5%	252,0	►_platform_memmove	libsystem_platform.dylib
201.0ms	4.4%	201,0	►inflate	libz.1.dylib
84.0ms	1.8%	84,0	►0x184657760	libz.1.dylib
82.0ms	1.8%	82,0	►objc_msgSend	libobjc.A.dylib
43.0ms	0.9%	43,0	►mach_msg_trap	libsystem_kernel.dylib

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    UIImage* image = [UIImage imageWithData:[NSData dataWithContentsOfURL:data.imageURL]];
    dispatch_async(dispatch_get_main_queue(), ^{
        self.imageView.image = image;
    });
});
```

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    UIImage* image = [UIImage imageWithData:[NSData dataWithContentsOfURL:data.imageURL]]; 
```

Turns out when we wrote

,,

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    UIImage* image = [UIImage imageWithData:[NSData dataWithContentsOfURL:data.imageURL]];
    dispatch_async(dispatch_get_main_queue(), ^{
        self.imageView.image = image;
    });
});

dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    CGImageSourceRef source = CGImageSourceCreateWithURL((CFURLRef)data.imageURL,
    NULL);
    CGImageRef cgImage = CGImageSourceCreateImageAtIndex(source, 0,
        (CFDictionaryRef @{
            (NSString*)kCGImageSourceShouldCache : @YES,
            (NSString*)kCGImageSourceShouldCacheImmediately : @YES
        });
    CFRelease(source);
    UIImage* image = [UIImage imageWithCGImage:cgImage];
    CGImageRelease(cgImage);

    dispatch_async(dispatch_get_main_queue(), ^{
        self.imageView.image = image;
    });
});

dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    CGImageSourceRef source = CGImageSourceCreateWithURL((CFURLRef)data.imageURL, NULL);
    CGImageRef cgImage = CGImageSourceCreateImageAtIndex(source, 0,
        (CFDictionaryRef @{
            (NSString*)kCGImageSourceShouldCache : @YES,
            (NSString*)kCGImageSourceShouldCacheImmediately : @YES
        });
    CFRelease(source);

    dispatch_async(dispatch_get_main_queue(), ^{
        self.imageLayerView.layer.contents = cgImage;
        CGImageRelease(cgImage);
    });
});
```

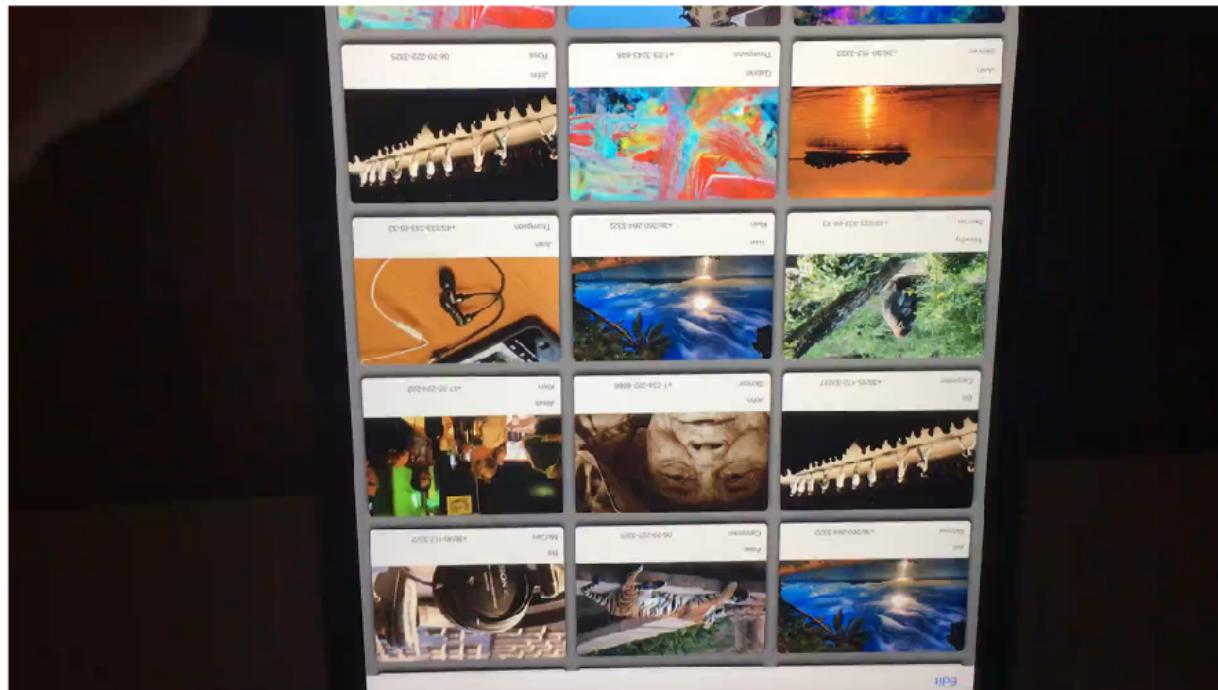
Turns out when we wrote  
this...

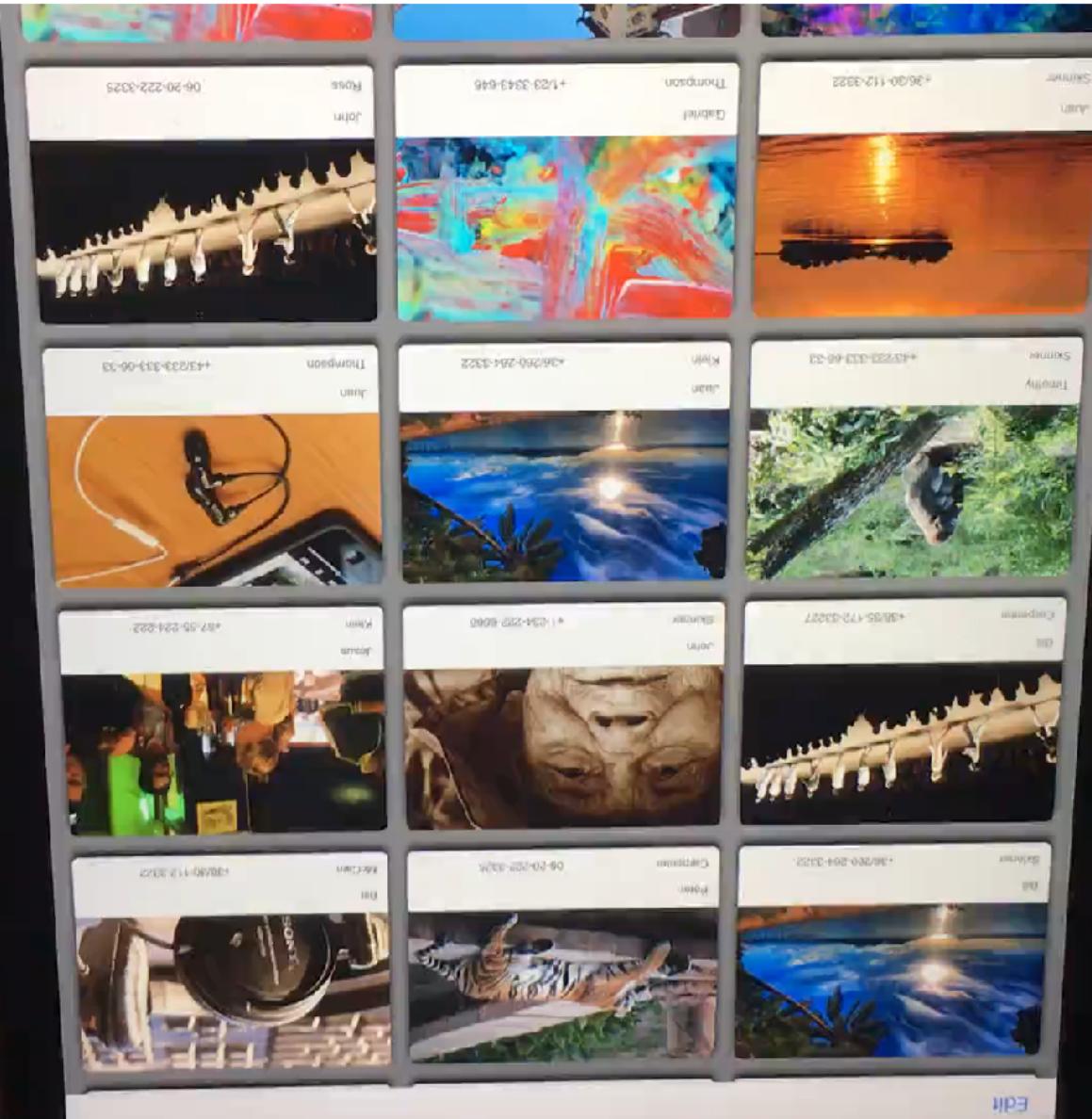
... we meant this.

or this.

```
});  
});
```

# Almost there

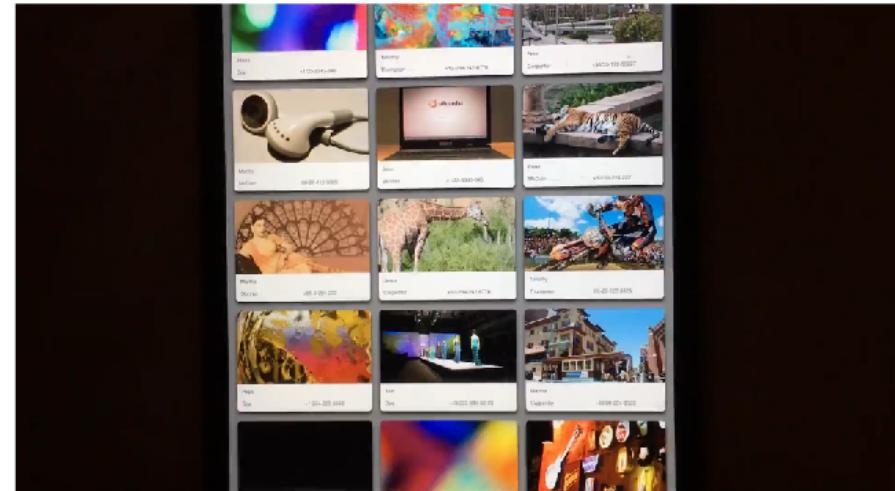
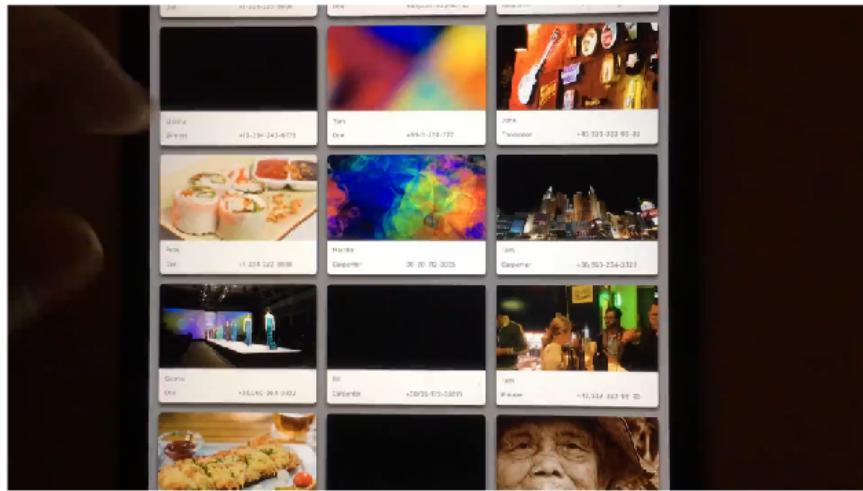


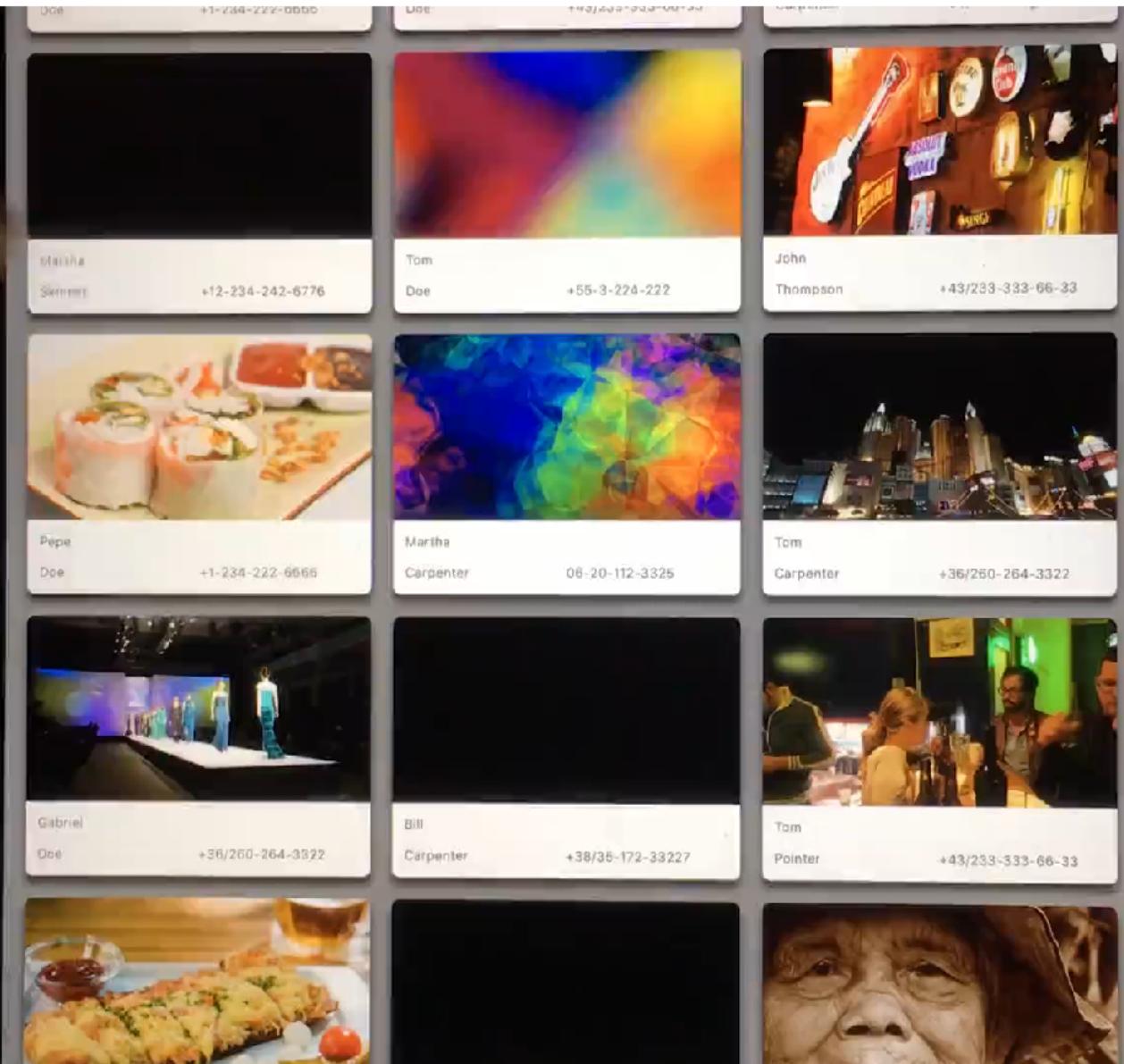


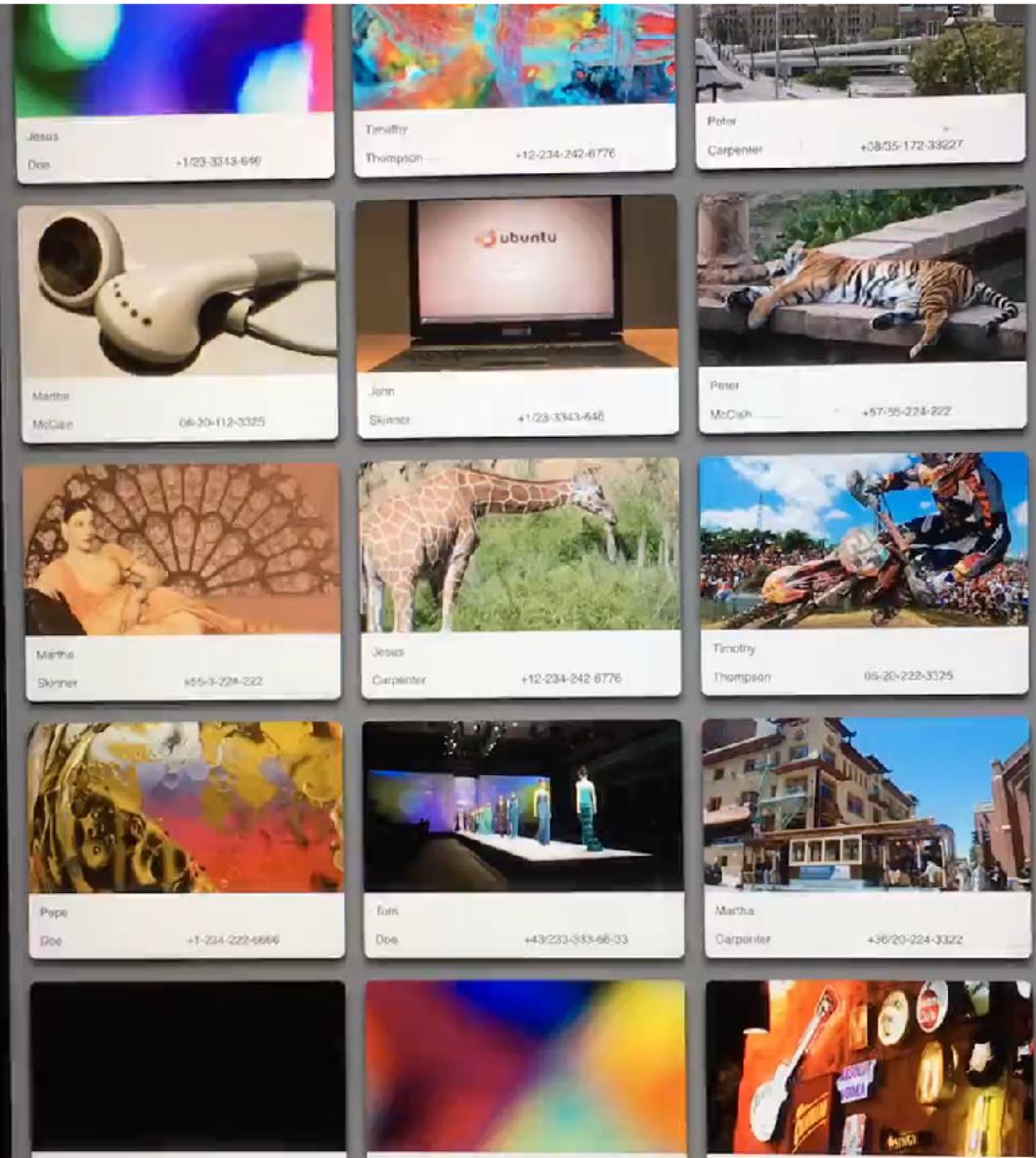
**PNG** → DEFlate (zip kind of) → **slow**

**JPEG** → DCT (MPEG kind of)  
hardware accelerated → **fast**

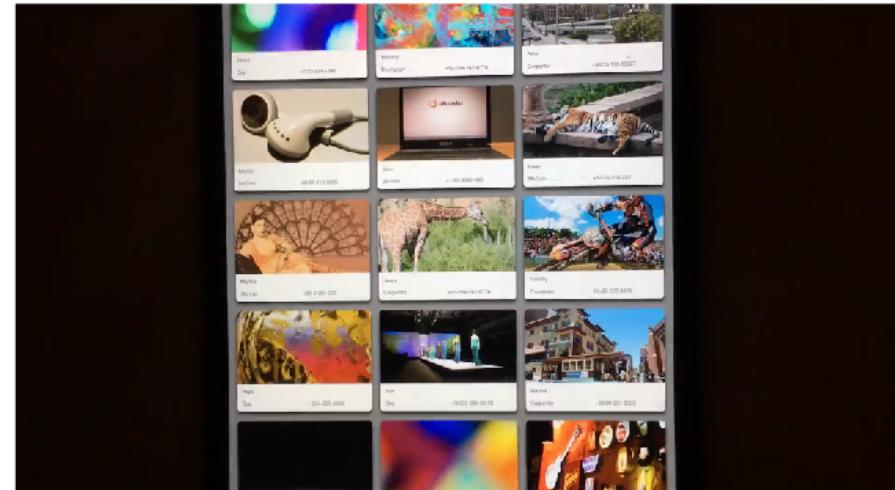
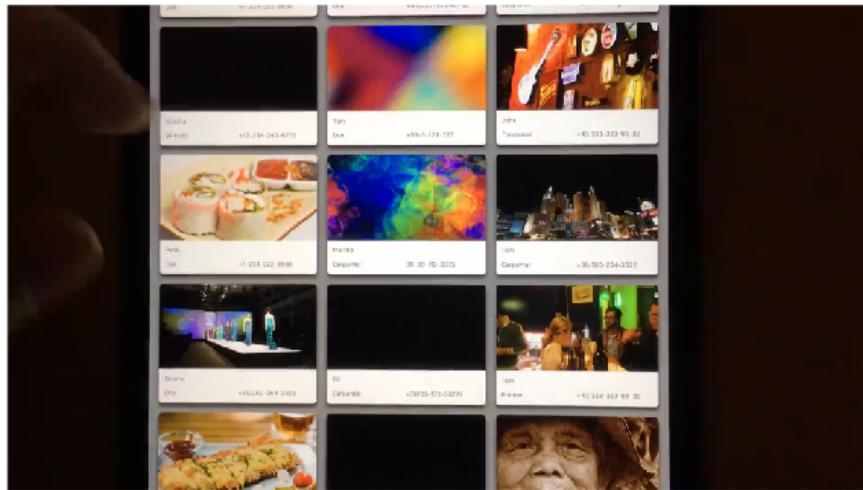
# Recap



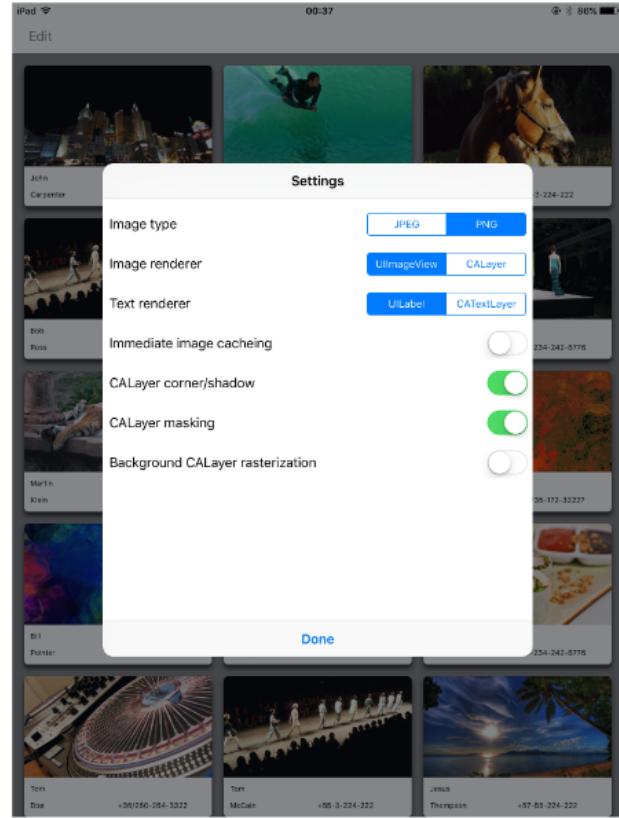




# Recap



# Try it yourself!



[github.com/tzahola/Scrolling60fps](https://github.com/tzahola/Scrolling60fps)

# **Thank you!**

## **Q&A**