Realidad Virtual y Aumentada Vuforia-Wall

Eduard Feicho

Overview

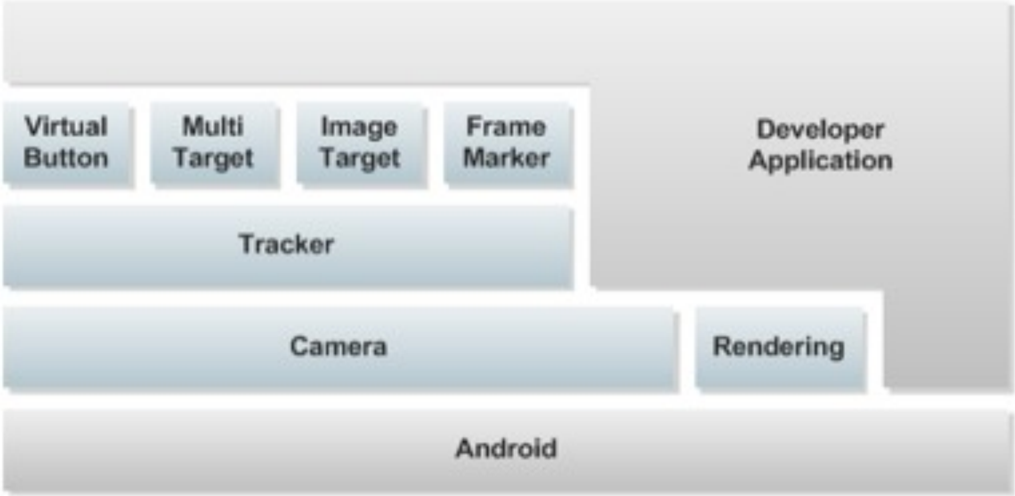
- Vuforia SDK (iPhone)
- Vuforia-Wall
- Demonstración

Vuforia SDK

- SDK de Qualcomm
  - muy nuevo (2010/2011)
  - Snapdragon procesadores (Nexus One, HTC Desire HD)
  - FastCV “iOnRoad increased app performance by 10-15% in 2 days”
  - https://developer.qualcomm.com/develop/mobile-technologies/augmented-reality

- Características
  - Codigos QR
  - Marcadores naturales y múltiples
  - Botones virtual
  - iPhone y Android

Vuforia SDK



- Camera
  - Matriz calibrado
  - Flash (lampo)
  - Enfocar

- Tracker
  - IMAGE_TRACKER/ MARKER_TRACKER
  - start/stop

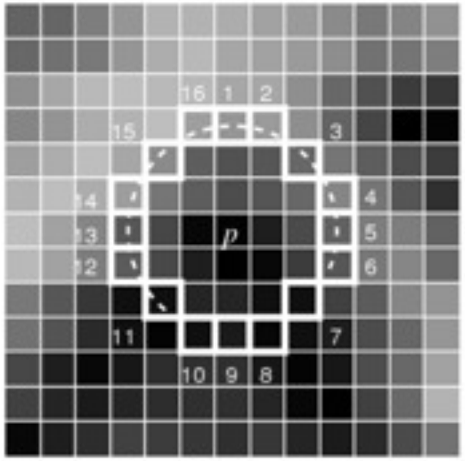
- Targets
  - ImageTarget de Trackable (+createVirtualButton, +getVirtualButton)
  - MultiTarget de Trackable (+addPart(Trackable), +setPartOffset(Matrix34F))
  - Frame

Natural Features/ Marcadores Naturales

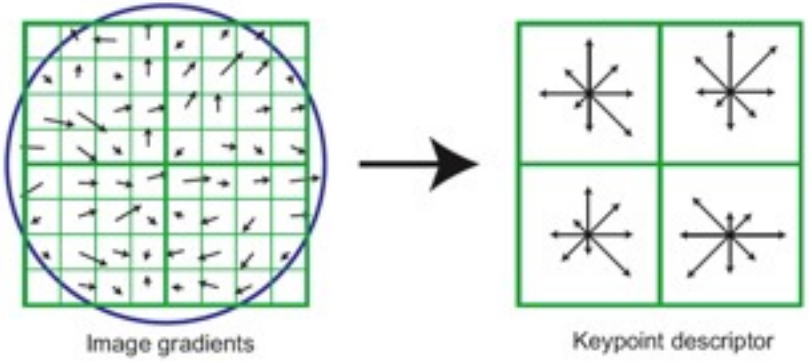
FAST-corner detector [Edward Rosten]

Features from Accelerated Segment Test

Al menos 12 pixeles adyacentes son más brillante o oscuro que p

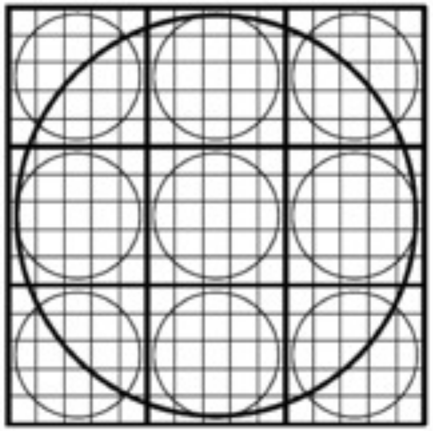


SIFT [David Lowe]



- Traslación
- Escale (DoG pirámide)
- Rotación (Lo más fuerte gradiente)
- Histograma de Gradientes (Affine Invariance)
- Cambia de luz constante+linear
  (gradiente+normalización del vector)

Natural Features



PhonySIFT [Daniel Wagner]

Features from Accelerated Segment Test

- FAST Corner Detector
- Features de las escalas pre-computado
- SIFT con histograma de dimensión 3x3x4
  (10% peor que de 128 dimensiones)

-> Daniel Wagner trabaja con Qualcomm R&D Vienna

Overview

- Vuforia SDK (iPhone)
- Vuforia-Wall
- Demonstración

Vuforia Wall

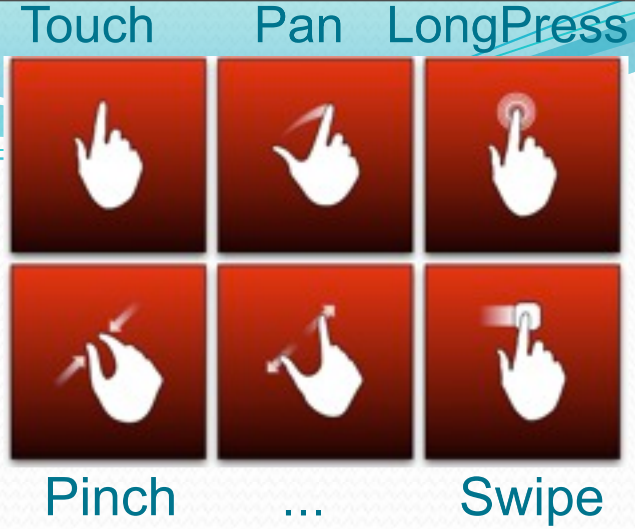
- Virtual Foto Wall
  - Fotos del viaje
  - Familia
  - Humorismo
- Características
  - Obtener fotos (camera, biblioteca interna)
  - Colocar, girar, escalar

EAGLView.mm

```
- (void)setup3dObjects; {
    [objects3D removeAllObjects];
    for (int i=0; i < [ImageWall sharedInstance].images.count; i++)
    {
          Plane3D *obj3D = [[Plane3D alloc] init];
          TouchImageView* imageView = [[ImageWall sharedInstance].images objectAtIndex:i];

          obj3D.dx = imageView.x;
          obj3D.dy = imageView.y;
          obj3D.rotation = imageView.rotation;
          obj3D.scale = imageView.scale;

          [obj3D setTextureWithImage:imageView.image];

          [objects3D addObject:obj3D];
          [obj3D release];
    }
}

- (void)postInitQCAR; {
    QCAR::setHint(QCAR::HINT_IMAGE_TARGET_MULTI_FRAME_ENABLED, 1);
    QCAR::setHint(QCAR::HINT_IMAGE_TARGET_MILLISECONDS_PER_MULTI_FRAME, 25);
    QCAR::setHint(QCAR::HINT_MAX_SIMULTANEOUS_IMAGE_TARGETS, 1);
}
```

EAGLView.mm

```
- (void)renderFrameQCAR;
{
    [self setFramebuffer];

    // Clear colour and depth buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Render video background and retrieve tracking state
    QCAR::State state = QCAR::Renderer::getInstance().begin();
    QCAR::Renderer::getInstance().drawVideoBackground();

    for (int i = 0; i < state.getNumActiveTrackables(); ++i) {
        // Get the trackable
        const QCAR::Trackable* trackable = state.getActiveTrackable(i);

        for (int j=0; j<objects3D.count; j++) {
            QCAR::Matrix44F modelViewMatrix = QCAR::Tool::convertPose2GLMatrix(trackable->getPose());
            Object3D *obj3D = [objects3D objectAtIndex:j];

            ...
        }
    }
    QCAR::Renderer::getInstance().end();

    [self presentFramebuffer];
}
```



Gestures

```
- (void)setup; {
    UIView.userInteractionEnabled = YES;
    UIView.multipleTouchEnabled = YES;
    UIRotationGestureRecognizer *rotationGesture = [[UIRotationGestureRecognizer alloc]
    initWithTarget:self action:@selector(handleRotationGesture:)];
    [self addGestureRecognizer:rotationGesture];
}

- (IBAction)handleRotationGesture:(UIRotationGestureRecognizer *)sender; {
    drotation = [sender rotation] * 50.0;
    [self updateImageTransform];
    if (sender.state == UIGestureRecognizerStateEnded) {
        rotation = rotation + drotation;
    }
    drotation = 0.0;
}

- (void)updateImageTransform; {
    float x_new = x + dx;       float y_new = y + dy;       float rotation_new = rotation + drotation;
    float scale_new = scale * dscale;

    CGAffineTransform t_translate = CGAffineTransformMakeTranslation(x_new, y_new);
    CGAffineTransform t_rotation  = CGAffineTransformMakeRotation(rotation_new / 180.0 * 3.14);
    CGAffineTransform t_scale     = CGAffineTransformMakeScale(scale_new, scale_new);
    self.transform = CGAffineTransformConcat(CGAffineTransformConcat(t_translate, t_rotation),
    t_scale);
}
```

Problemas/Especialidades

- ARC (Automatic Reference Counting) demasiado nuevo
  - borrar retain/release/NSAutoReleasePool
  - cambiar unas “properties” de “retain” a “weak”/”strong”

- Texturas cuadráticas

- Ya había gestiones en ARParentViewController -> borrar:

  -
    ```
    [parentView addSubview: overlayViewController.view];
    ```

  -
    ```
    - (void) touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
    - (void) touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
    - (void) touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
    - (void) touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
    ```

Overview

- Vuforia SDK (iPhone)
- Vuforia-Wall
- Demonstración