

Visión por computador con imágenes de rango

Lección 10.1

Dr. Pablo Alvarado Moya

MP6127 Visión por Computadora
Programa de Maestría en Electrónica
Énfasis en Procesamiento Digital de Señales
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica

I Cuatrimestre 2013

- ¿Cómo inferir información 3D de imágenes 2D?
- Métodos
 - Estimación de pose
 - Visión estéreo
 - Estructura desde movimiento
 - Imágenes de rango

- ¿Cómo inferir información 3D de imágenes 2D?
- Métodos
 - Estimación de pose ✓
 - Visión estéreo ✓
 - Estructura desde movimiento ✓
 - Imágenes de rango ←

Imágenes de rango

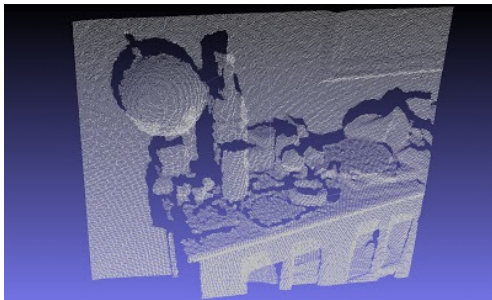
- Proveen información de profundidad para cada pixel
- Dispositivos actuales capturan imagen correspondiente
- Grises o RGB (RGB-D)



Cornell Activity Dataset

Imágenes de rango

- Recientemente productos comerciales de precio accesible proveen imágenes de rango a 30 fps o más
- (PrimeSense, Microsoft Kinect, Asus Xtion, Optex, Mesa Imaging, ...)
- Utilizan luz estructurada o sensores ToF (*"time of flight"*)



- Tecnología disponible a partir de año 2000
- Rango desde 1 metro hasta decenas de metros
- Resolución de profundidad ≈ 1 cm en versiones de 1 m
- Fuente de luz modulada o con obturador sincronizado
- Miden diferencia de fase entre luz emitida y recibida
- Ejemplos:
 - Creative* Interactive Gesture Camera (\approx US\$150, 320×240)
 - SoftKinetic DepthSense 325 (\approx US\$250, 320×240)
- Comparado a métodos de luz estructurada:
 - Mayor resolución espacial (x, y) en mapa de profundidad
 - Medición paralela en chip permite mayores tasas de cuadros por segundo

- En electrónica de consumo, mercado dominado por PrimeSense
- Microsoft adoptó en Kinect tecnología de PrimeSense para medición 3D
- Tecnología protegida: no todo detalle técnico conocido
- Usa cámara RGB, fuente IR, cámara IR
- Imagen de profundidad es de 640×480
- Se utiliza interpolación de puntos z calculados (≈ 4000 px)
- Rango z : 0,8 m–3,5 m, resolución 1 cm @2 m
- Resolución (x, y) : 3 mm @2 m

Patrón e imagen de puntos



Patrón de puntos



Patrón distorsionado

- Proceso de triangulación entre dos patrones se usa para estimar z
- Concepto funciona al tener la escena superficies suaves

Información adicional abre nuevas posibilidades de procesamiento

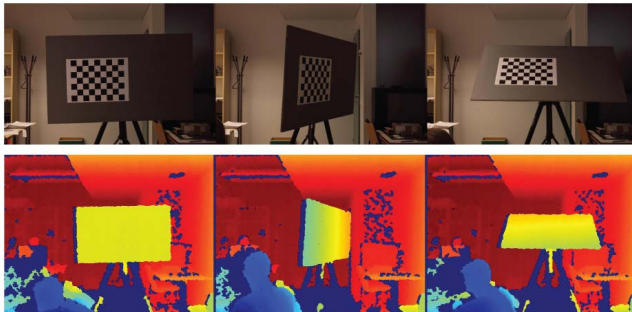
- Nuevos requisitos:
Calibración inter-cámaras (RGB + Profundidad)
- Algoritmos de segmentación espaciales
- Scanners 3D ([Kinect-Fusion](#))
- Algoritmos de detección de [pose humana](#)
- Reconocimiento de gestos
- Rastreo de movimiento
- Realidad aumentada (tele-conferencias)
- ...

Bibliotecas especializadas:

- `freenect` (OpenKinect)
- `openNI` y `NiTE`
- Intel SDK para “Interactive Gesture Camera” (Gestos)
- Cinder
- OpenFrameworks

- Problema de calibración: imagen de profundidad e imagen RGB deben relacionarse
- Además de distorsiones radial y tangencial, hay distorsión en z
- Artículo:
Herrera, D., Kannala, J. y Heikkilä, J. *Joint Depth and Color Camera Calibration with Distortion Correction*. PAMI Vol.34, No.10, Octubre 2012

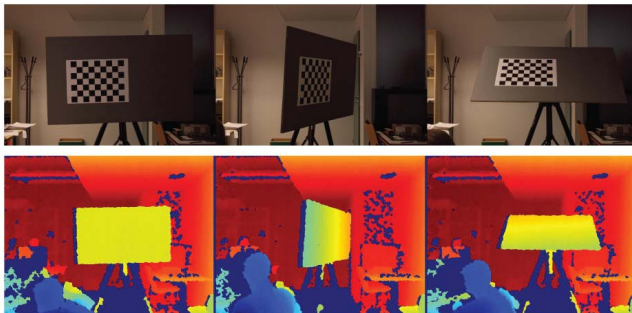
Calibración en sistema de rango



Herrera et.al.,2012

- Tablero utilizado para calibración de cámara(s) RGB
- Plano del tablero usado para calibración de cámara de rango.
- Múltiples puntos de vista necesarios

Detección de correspondencias



Herrera et.al.,2012

- En imagen(es) RGB se extraen esquinas de tablero
- En imagen de profundidad, usuario marca esquinas del plano
- Proceso de calibración se inicializa con método de Z. Zhang para cada cámara por separado

Mapeo de punto en sistema de cámara $\underline{\mathbf{x}}_c = [x_c, y_c, z_c]^T$ a punto en imagen $\underline{\mathbf{p}}_c = [u_c, v_c, 1]^T$ usa

- Punto normalizado

$$\underline{\mathbf{x}}_n = [x_n, y_n, 1]^T = [x_c/z_c, y_c/z_c, 1]^T, r^2 = x_n^2 + y_n^2$$

- Distorsión radial y tangencial

$$\rho = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) \quad \underline{\mathbf{x}}_g = \begin{bmatrix} 2k_3 x_n y_n + k_4 (r^2 + 2x_n^2) \\ k_3 (r^2 + 2y_n^2) + 2k_4 x_n y_n \\ 1 \end{bmatrix}$$

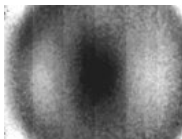
$$\underline{\mathbf{x}}_k = \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \begin{bmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{bmatrix} \underline{\mathbf{x}}_n + \underline{\mathbf{x}}_g$$

- Las coordenadas en la imagen son

$$\underline{\mathbf{p}}_c = \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_{cx} & 0 & u_{0c} \\ 0 & f_{cy} & v_{0c} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \mathbf{K} \underline{\mathbf{x}}_k$$

- 9 parámetros a optimizar son \mathbf{K} y $\underline{\mathbf{k}}_c = [k_1, k_2, k_3, k_4, k_5]^T$

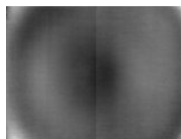
- Cámaras usualmente devuelven valor de disparidad, en vez de valor de distancia z
- Caso de Kinect usa “unidades de disparidad de Kinect” (kdu)
- Se plantea conversión desde imagen $\underline{p}_d = [u_d, v_d]^T$ a sistema de cámara $\underline{x}_d = [x_d, y_d, z_d]^T$ (relación inversa) usando mismo modelo anterior (se asume reversible).
- Sensor tiene un patrón de error sistemático constante $D_\delta(u, v)$, pero dependiente en amplitud de distancia:



0,56 m



1,24 m



Estimado

Herrera et.al.,

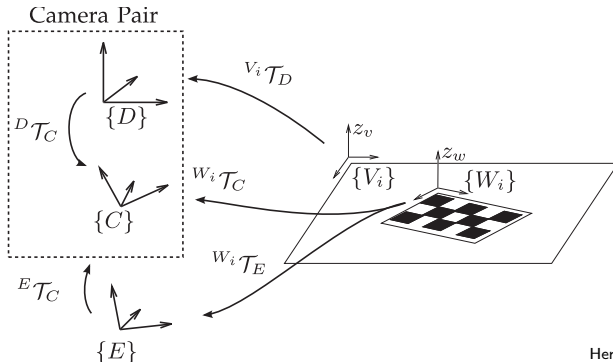
2012

- Conversión de disparidad $d \rightarrow z_d$:
1) corrección de distorsión y 2) inversión escalada:


$$d_k = d + D_\delta(u, v) \exp(\alpha_0 - \alpha_1 d)$$

$$z_d = \frac{1}{c_1 d_k + c_0}$$

Calibración extrínseca

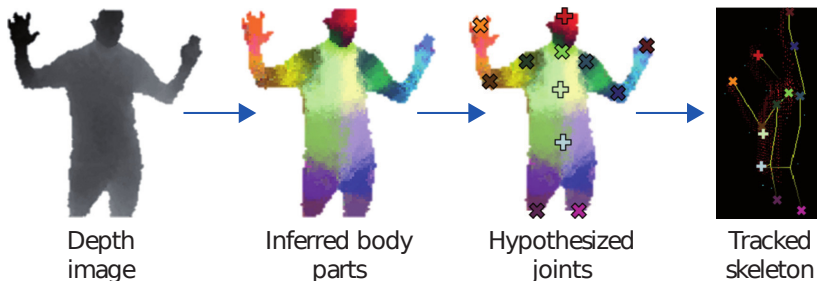


Herrera et.al, 2012

- Cámaras RGB se calibran (intrínseca y extrínsecamente) con método de Zhang
- Tablero no es visible en imagen de profundidad, solo el plano
- Relación entre plano y tablero ($\{V\}$ y $\{W\}$) requiere **coplanaridad** de ambos sistemas y múltiples imágenes
- \Rightarrow deben estimarse las normales del plano en cada imagen
- Métodos no lineales estiman parámetros de distorsión, incluyendo $D_{\delta}(u, v)$ que tiene 307 200 parámetros: 
- Se separa problema para optimizar $D_{\delta}(u, v)$ por aparte
- Propuesta mejora precisión a calibración de fábrica

- Información 3D permite estimación de pose en tiempo real
- Kinect de Microsoft aprovecha GPU para realizar estimación en 5 ms
- Artículo:
Shotton et.al. "*Real-Time Human Pose Recognition in arts from Single Depth Images*". Proc. IEEE CVPR, 2011

Concepto general



Z. Zhang, "MS Kinect Sensor and Its Effect" IEEE Multi-media. April-June 2012

- Solo profundidad evita problemas de iluminación, vestimenta, color de piel, etc.
- Segmentación de silueta y fondo se simplifica

- Concepto usa alrededor de 1 millón de imágenes de entrenamiento



- Generadas de diferentes formas:
 - Prueba: imágenes reales etiquetadas manualmente
 - Entrenamiento: captura de movimiento replicada
 - Poses: conducir, bailar, pelear, correr, navegar menus ...
 - 500 k cuadros, algunos cientos de secuencias
 - Variaciones (semi-)automáticas de rotación vertical, espejo derecha-izquierda, posición en escena, forma y tamaño de forma, pose de cámara ...

- Clases a reconocer: 31 partes del cuerpo

- (A:Arriba, B:aBajo, I:Izquierda, D:Derecha)
- Cabeza (AI,AD,BI,BD)
- Cuello
- Hombro (I,D)
- Brazo (AI,AD,BI,BD)
- Codo (I,D)
- Muñeca (I,D)
- Mano (I,D)
- Torso (AI,AD,BI,BD)
- Pierna (AI,AD,BI,BD)
- Rodilla (I,D)
- Tobillo (I,D)
- Pie (I,D)



Shotton et.al.,2011

Etiquetas de colocan cerca de **articulaciones**

Características para la clasificación

- Características por pixel
- Características de simple cálculo
- Paralelizables en GPU
- Comparación de profundidad:

$$f_{\theta}(I, \underline{\mathbf{x}}) = d_I \left(\underline{\mathbf{x}} + \frac{\underline{\mathbf{u}}}{d_I(\underline{\mathbf{x}})} \right) - d_I \left(\underline{\mathbf{x}} + \frac{\underline{\mathbf{v}}}{d_I(\underline{\mathbf{x}})} \right)$$



Shotton et.al.,2011

- Idea es clasificar cada pixel de forma independiente a las partes del cuerpo
- Se utilizan bosques de decisión aleatorios (*Randomized decision forests*)
- Cada bosque tiene T árboles de decisión, que tienen nodos de decisión y hojas
- Cada nodo de decisión tiene una característica $f_{\theta}(I, \underline{x})$ asignada y un umbral τ

- Dependiendo del valor de la característica con respecto al umbral se toma el camino izquierdo o derecho
- Cada hoja del árbol t almacena una distribución de probabilidades $P_t(c)$ para las etiquetas c .
- La probabilidad final se promedia entre los árboles del bosque:

$$P(c \mid I, \underline{\mathbf{x}}) = \frac{1}{T} \sum_{t=1}^T P_t(c \mid I, \underline{\mathbf{x}})$$

- Cada árbol se entrena con un conjunto diferente de imágenes sintéticas
- Se elije un subconjunto arbitrario de 2000 píxeles de cada imagen
- Cada árbol se entrena con el siguiente algoritmo voraz:
 - 1 Elija arbitrariamente un conjunto de candidatos de decisión $\phi = (\theta, \tau)$
 - 2 Reparta las muestras de entrenamiento $Q = \{(I, \underline{\mathbf{x}})\}$ en subconjuntos izquierdo y derecho para cada ϕ :

$$Q_l(\phi) = \{(I, \underline{\mathbf{x}}) \mid f_\theta(I, \underline{\mathbf{x}}) < \tau\}$$

$$Q_r(\phi) = Q \setminus Q_l(\phi)$$

- 3 Calcule el ϕ con la mayor ganancia de información:

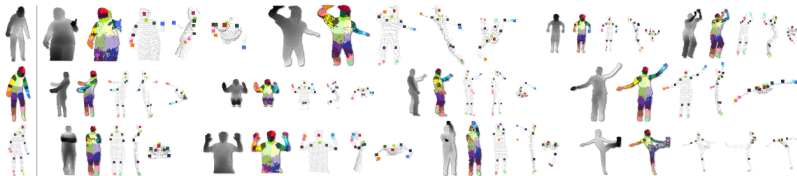
$$\phi^* = \arg \max_{\phi} G(\phi)$$

$$G(\phi) = H(Q) - \sum_{s \in l, r} \frac{Q_s(\phi)}{|Q|} H(Q_s(\phi))$$

- 4 La entropía de Shannon $H(Q)$ se calcula sobre el histograma normalizado de etiquetas de partes corporales $l_l(\underline{\mathbf{x}})$ para todo $(l, \underline{\mathbf{x}}) \in Q$
- 5 Si $G(\phi^*)$ supera un umbral y la profundidad del árbol es inferior a un máximo, se recurre en cada subconjunto $Q_l(\phi^*)$ y $Q_r(\phi^*)$
- Los autores reportan que con una aplicación distribuida, entrenar 3 árboles de profundidad 20, para 1 millón de imágenes, toma 1 día en un cluster de 1000 núcleos.

Posiciones de articulaciones

- Clasificación se hace para cada pixel
- ¿Cómo deducir dónde está la articulación?
- En artículo, los autores proponen desplazamiento de medias (mean-shift)
- En la realidad usan otro método (?)



Shotton et.al.,2011

Este documento ha sido elaborado con software libre incluyendo L^AT_EX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, LTI-Lib-2, GNU-Make, Kazam, Xournal y Subversion en GNU/Linux



Este trabajo se encuentra bajo una Licencia Creative Commons Atribución-NoComercial-LicenciarIgual 3.0 Unported. Para ver una copia de esta Licencia, visite <http://creativecommons.org/licenses/by-nc-sa/3.0/> o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

© 2013 Pablo Alvarado-Moya Escuela de Ingeniería Electrónica Instituto Tecnológico de Costa Rica