

Robótica Móvil un enfoque probabilístico

Técnicas de mapeo en 3D

Ignacio Mas

¿Por qué representar 3 dimensiones?

- Los mapas 2D se han usado exitosamente para tareas tales como navegación y localización.
- Los robots se mueven en un mundo de 3D.
- La evasión robusta de obstáculos y el planeamiento de trayectorias requieren modelos en 3D.
- ¿Cómo representar estructuras del entorno en 3D?

Representaciones más comunes

- Nube de puntos
- Grillas de vóxeles (pixel en 3D)
- Mapas de superficie
- Mallas
- ...

Nubes de puntos

- Pro:

- No discretizan los datos
- El área a mapear no esta limitada

- Contra:

- Uso de memoria no acotado
- No hay una representación directa de espacio libre o desconocido



Grillas de vóxeles 3D

- **Pro:**

- Representación volumétrica
- Tiempo de acceso constante
- Actualización probabilística



- **Contra:**

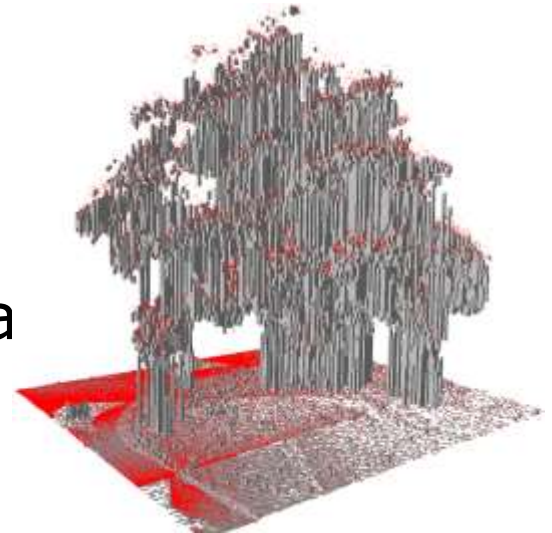
- Requerimiento de memoria: El mapa completo debe aloarse en memoria
- El tamaño del mapa debe saberse/suponerse
- Errores de discretización

Mapas 2.5D: “Mapas de elevación”

El promedio de todos los puntos escaneados caen en una celda

- **Pro:**

- Eficiente en cuanto a memoria
- Tiempo de acceso constante

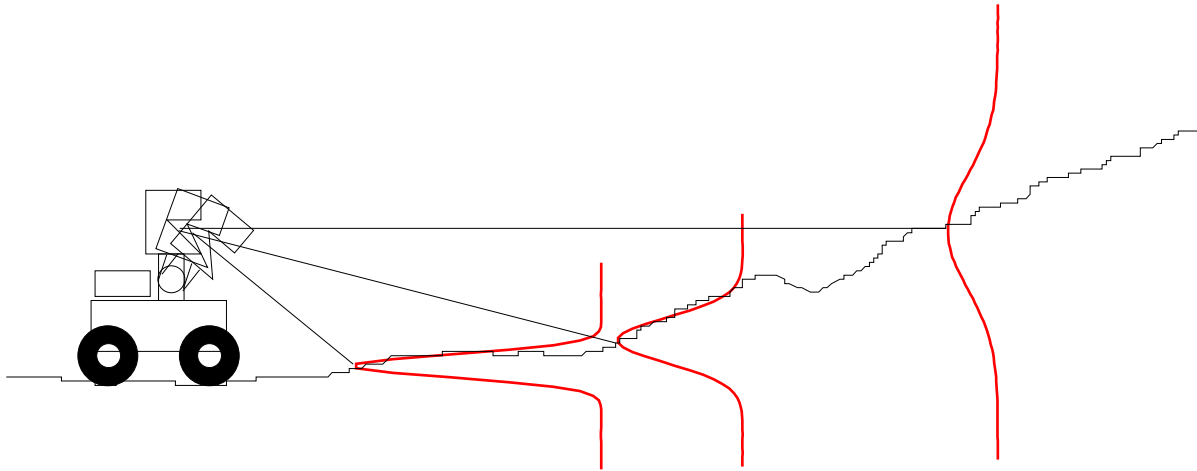


- **Contra:**

- No-probabilístico
- No hay distinción entre espacio libre y desconocido

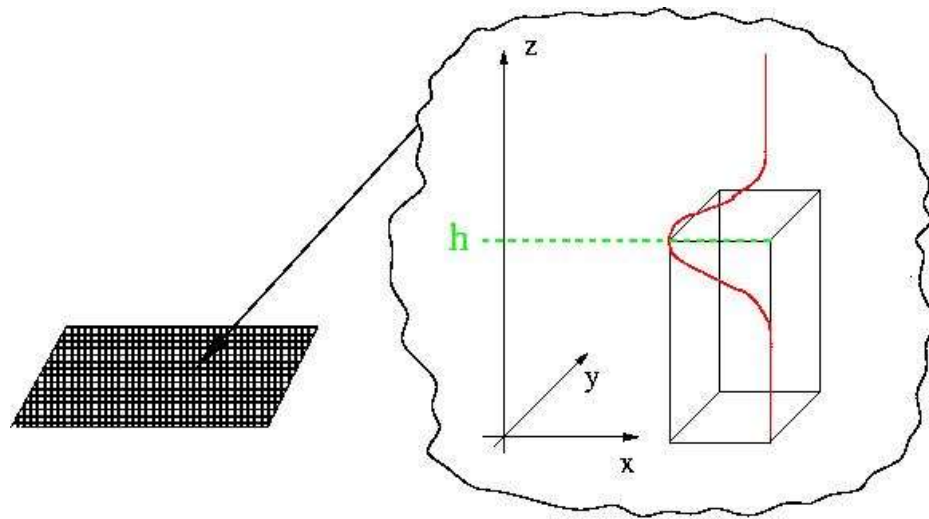
Mapas de elevación

- Grilla 2D que almacena una altura (elevación) estimada para cada celda
- En general, la incerteza aumenta con la distancia medida



Mapas de elevación

- Grilla 2D que almacena una altura (elevación) estimada para cada celda
- En general, la incerteza aumenta con la distancia medida
- Kalman para estimar la elevación

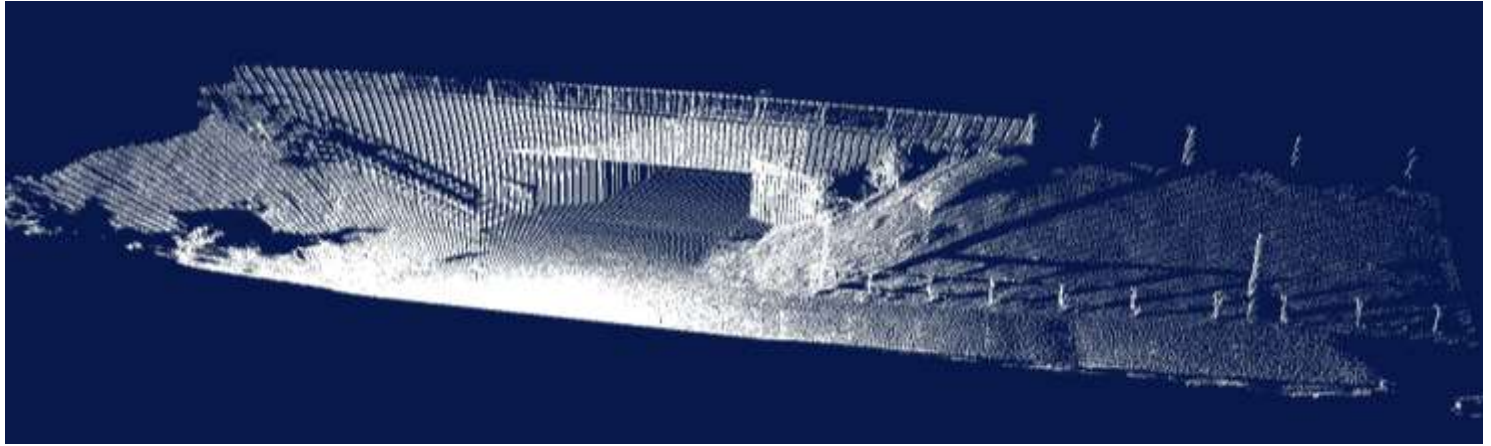


Mapas de elevación

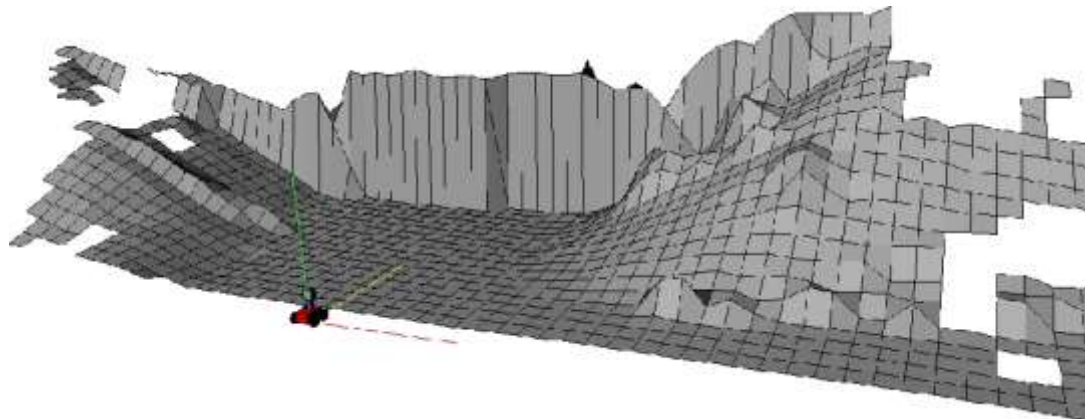
- **Pro:**
 - Representación 2.5D (vs. grilla 3D completa)
 - Tiempo de acceso constante
 - Estimación estadística de la altura
- **Contra:**
 - No contempla superficies verticales
 - Sólo un nivel es representado

Ejemplo de mapa de elevación

Datos
crudos:



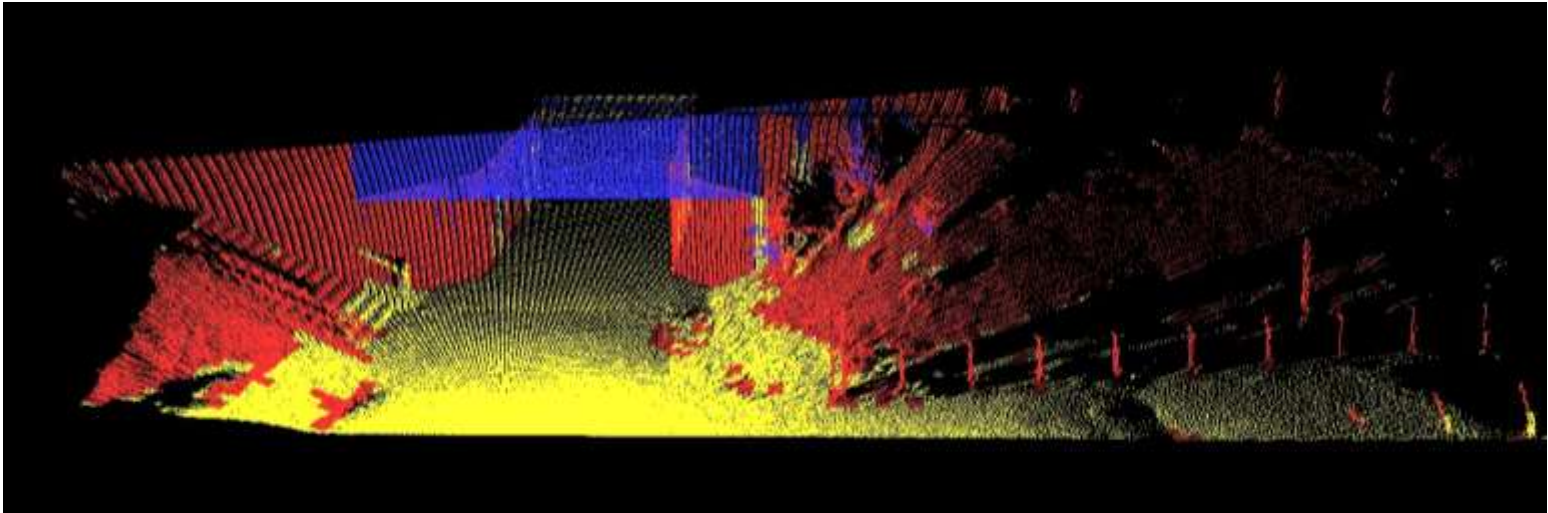
Mapa de
elevación:



Mapa de elevación extendido

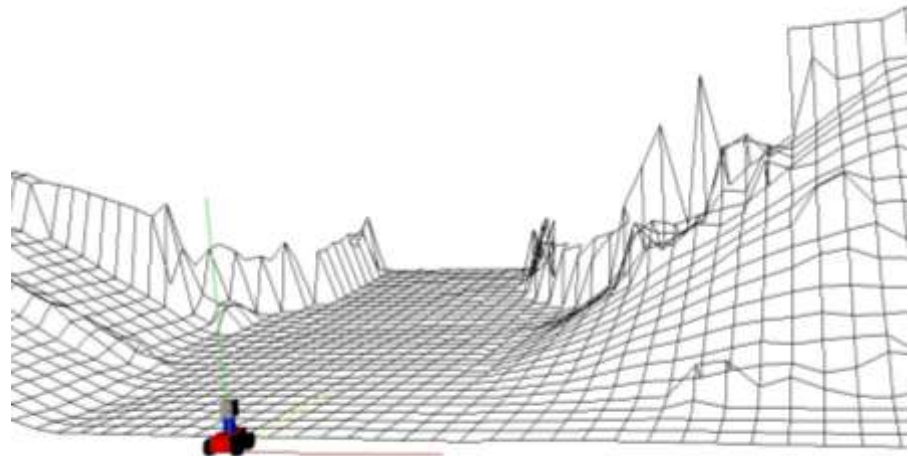
- Identificar:
 - Celdas que corresponden a estructuras verticales
 - Celdas que contienen espacios vacíos
- Chequear si la varianza de la altura de todas las mediciones de una celda es alta
- Chequear si el espacio vacío es mayor que la altura del robot ("gap cell")

Ejemplo: Mapa de elevación extendido



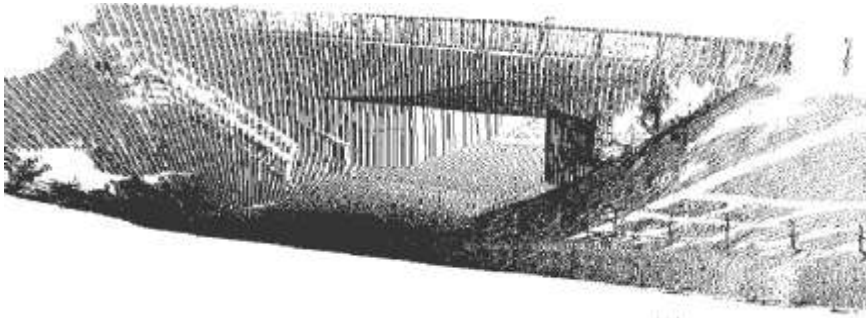
- Celdas con superficies verticales (rojo)
- Mediciones por sobre un espacio vacío grande (azul)
- Celdas vistas desde arriba (amarillo)

→ usar la brecha de celdas para determinar transitabilidad

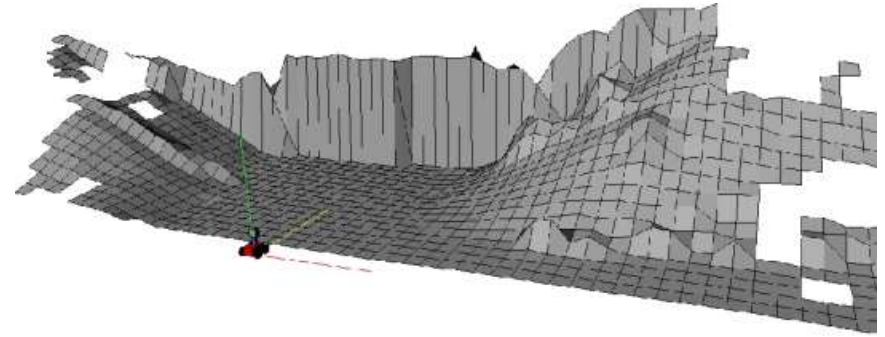


Mapa de elevación extendido

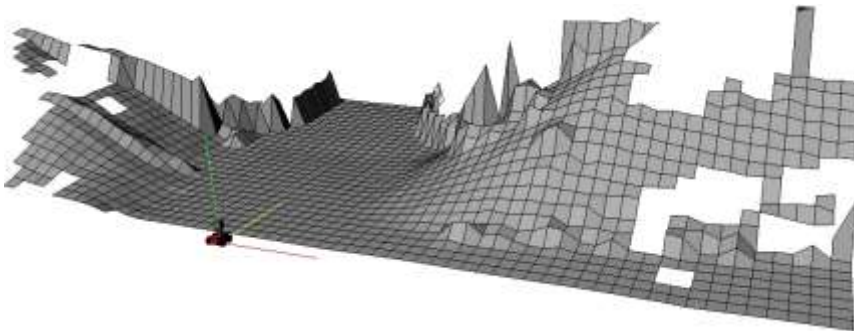
Tipos de mapa de terreno



Nube de puntos

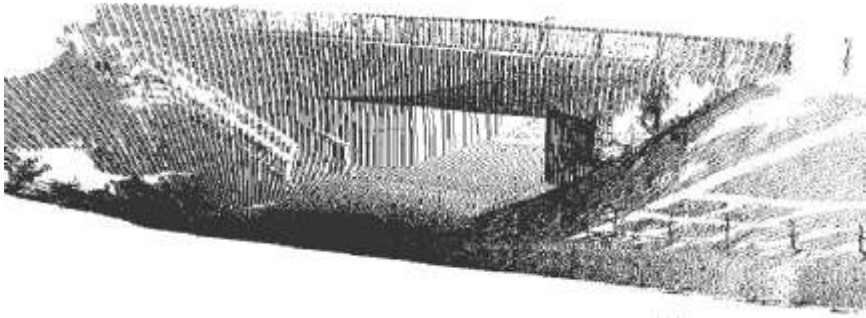


Mapa de elevación estándar

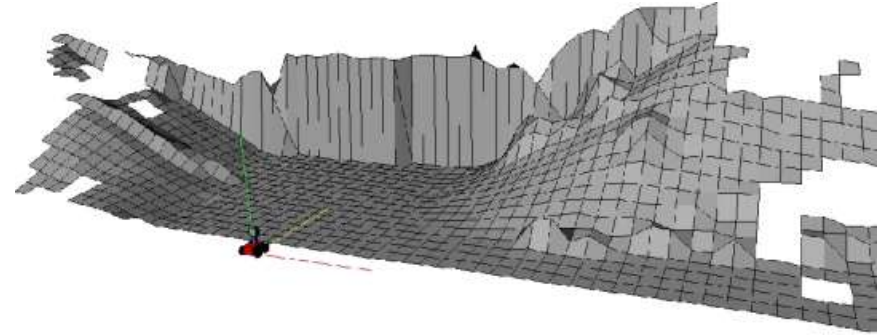


Mapa de elevación
extendido

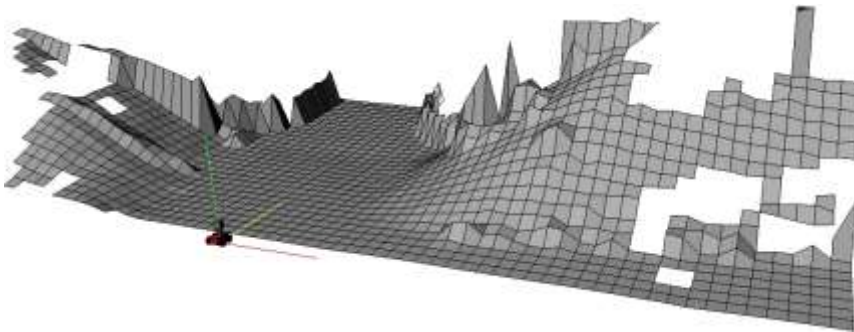
Tipos de mapa de terreno



Nube de puntos



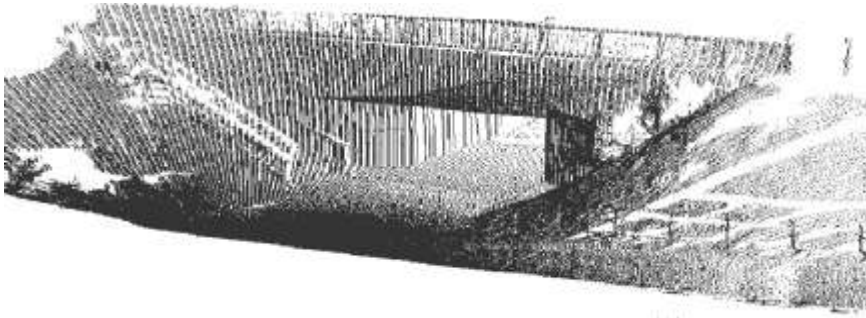
Mapa de elevación estándar



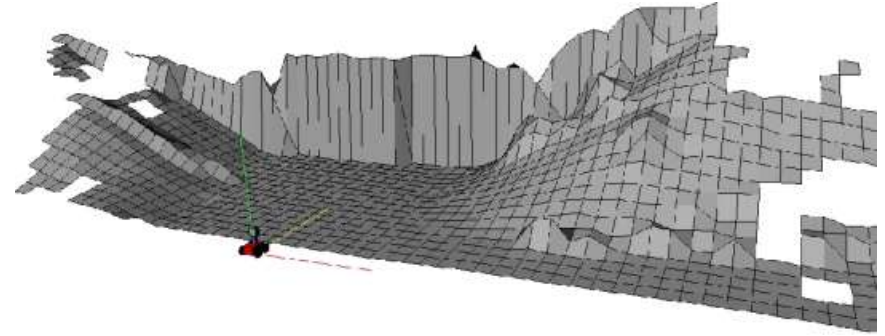
Mapa de elevación extendido

- + Planeamiento de paso bajo nivel es posible (celdas con espacios verticales)
- No puede pasar por abajo y por arriba del puente (solo un nivel por celda)

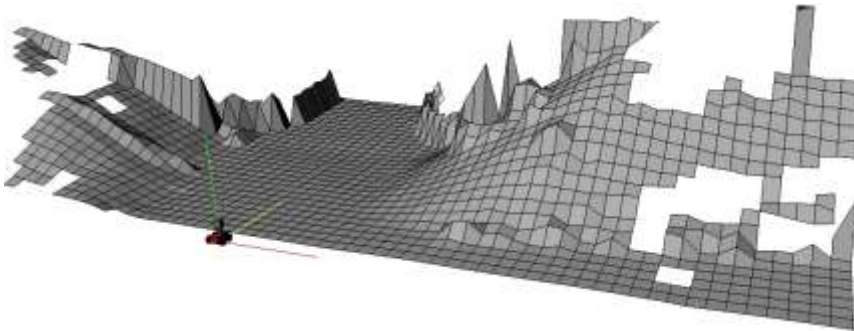
Tipos de mapa de terreno



Nube de puntos



Mapa de elevación estándar

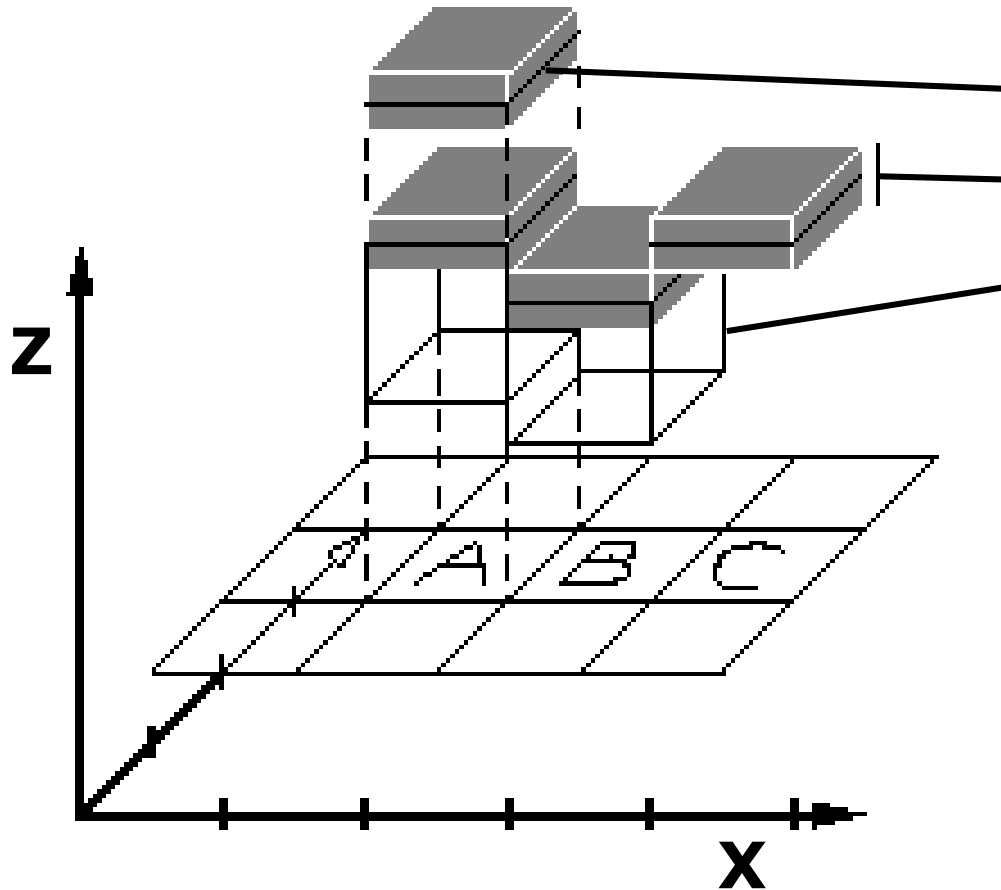


Mapa de elevación
extendido



**Mapa de superficie
multi-nivel**

Representación de mapas multi-nivel (MLS)



Cada celda 2D almacena varios paneles con valores:

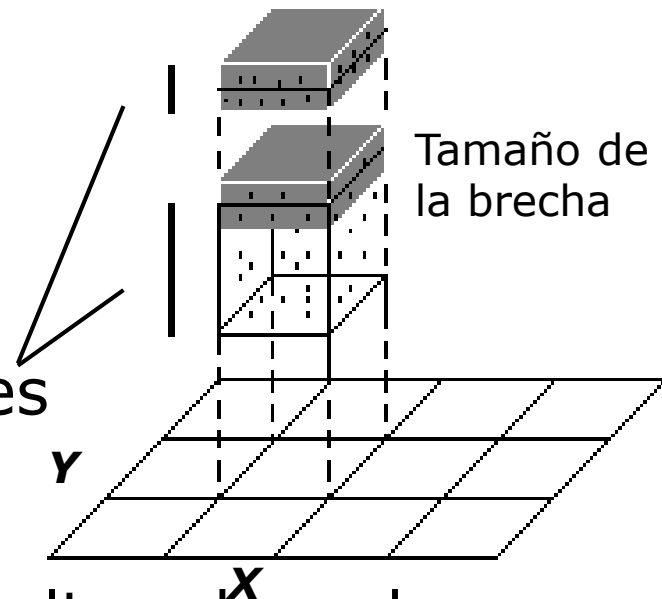
- La media de la altura μ
- La varianza de la altura σ
- Un valor de profundidad d

Notar:

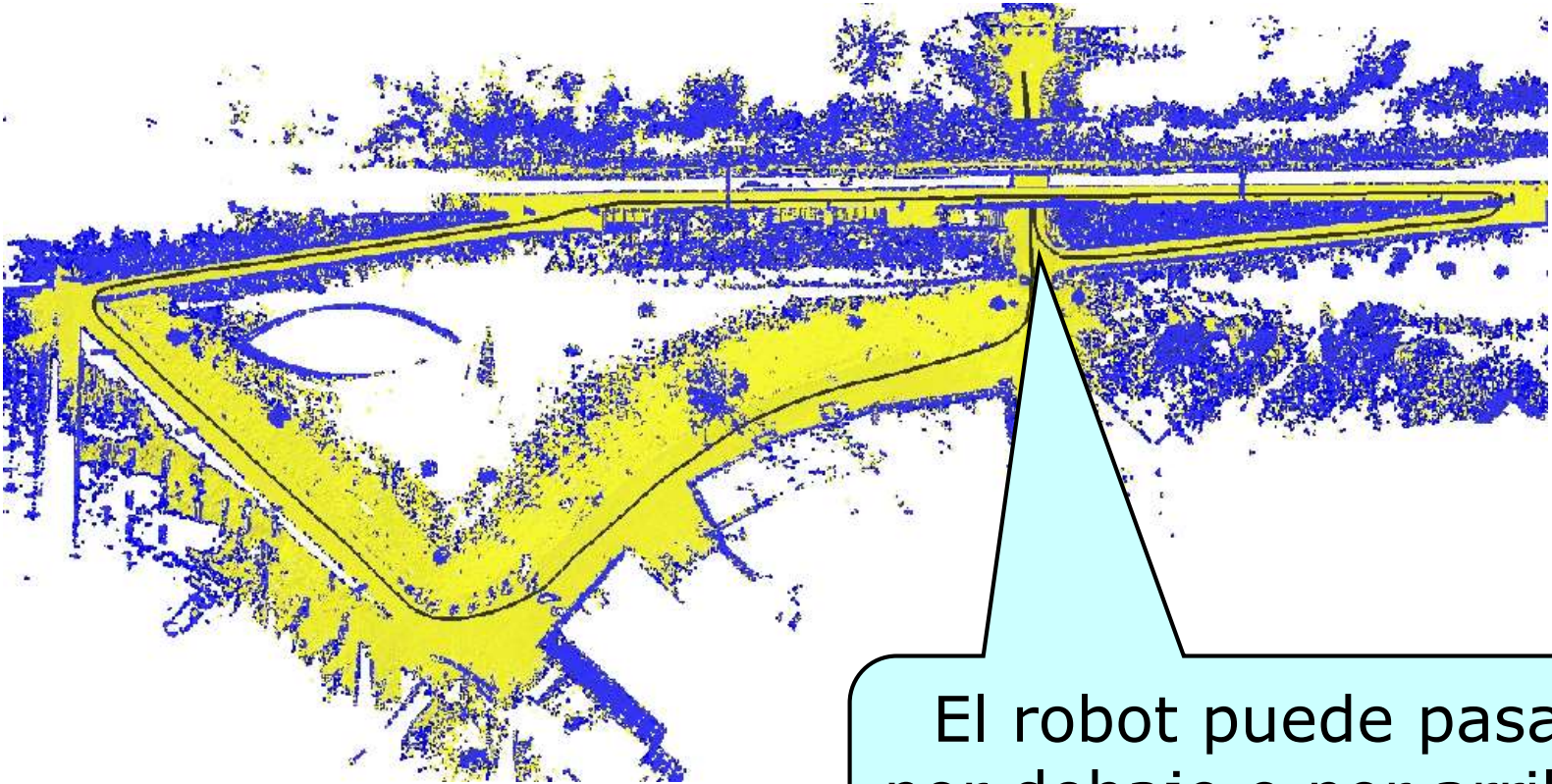
- Un panel puede no tener profundidad (objetos planos, ej: piso)
- Una celda puede tener muchos paneles (espacios verticales, ej: puentes)

De nube de puntos a mapas multi-nivel

- Determinar la celda para cada punto 3D
- Calcular intervalos verticales
- Clasificar en intervalos verticales ($>10\text{cm}$) u horizontales
- Aplicar Kalman para estimar la altura basada en todos los puntos para los intervalos horizontales
- Tomar la media y la varianza de las mediciones más altas de los intervalos verticales

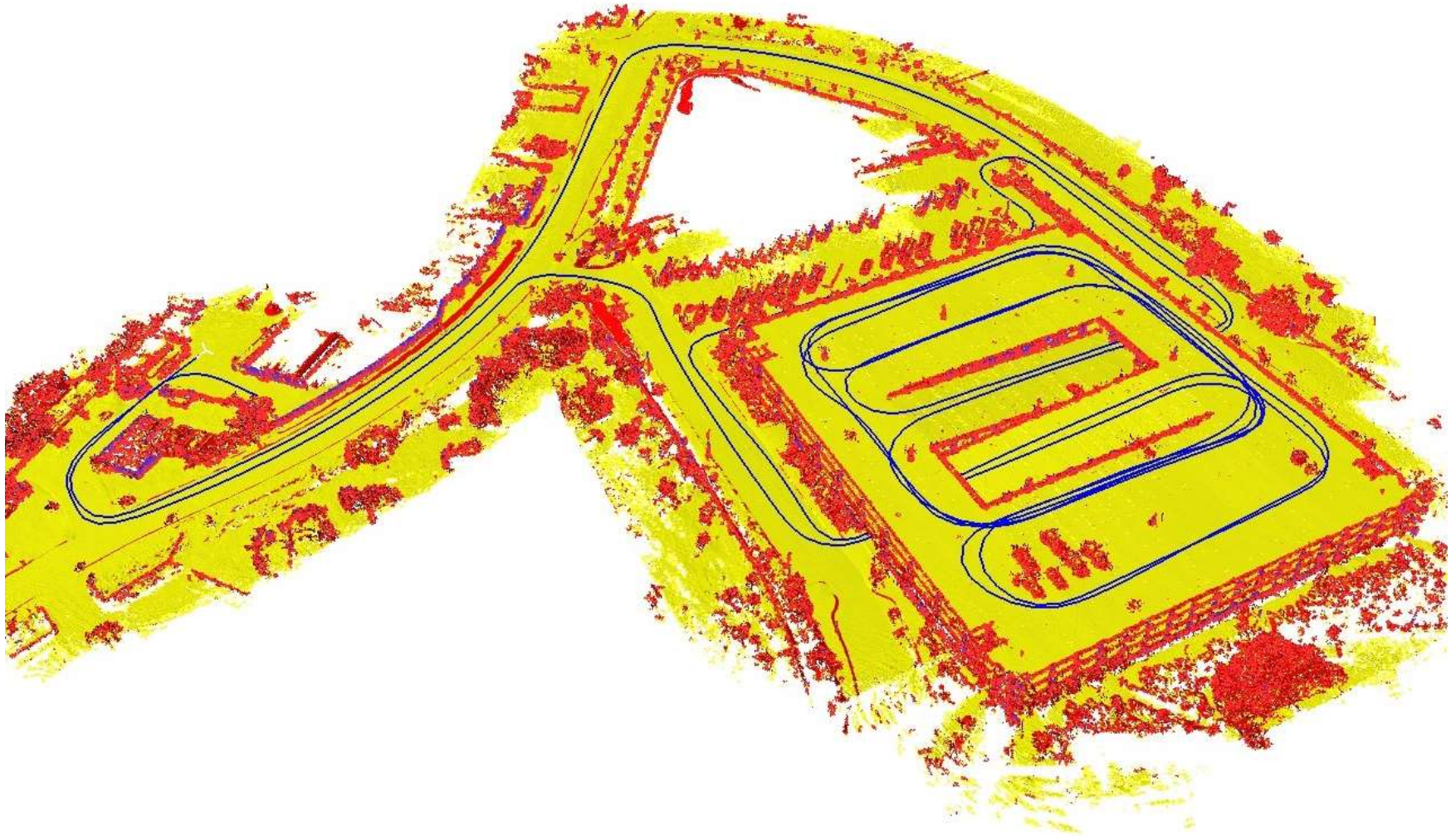


Resultados



- Tamaño del mapa: 299 x 1
- Resolución de la celda: 10 *cm*
- Número de puntos de datos: 45,000,000

Mapa MLS de estructura de estacionamiento



Mapas MLS

- Pro:

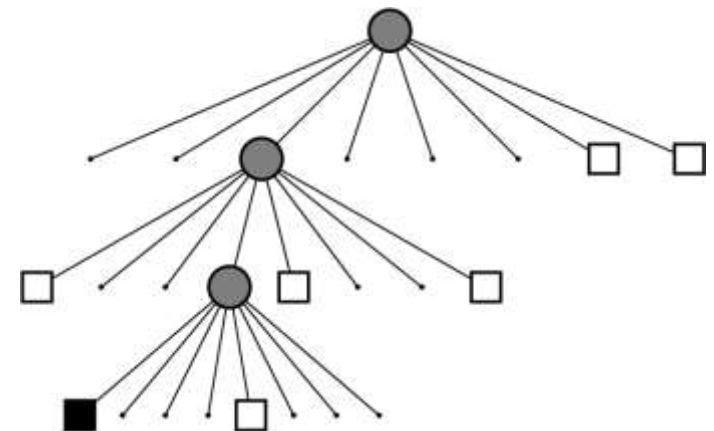
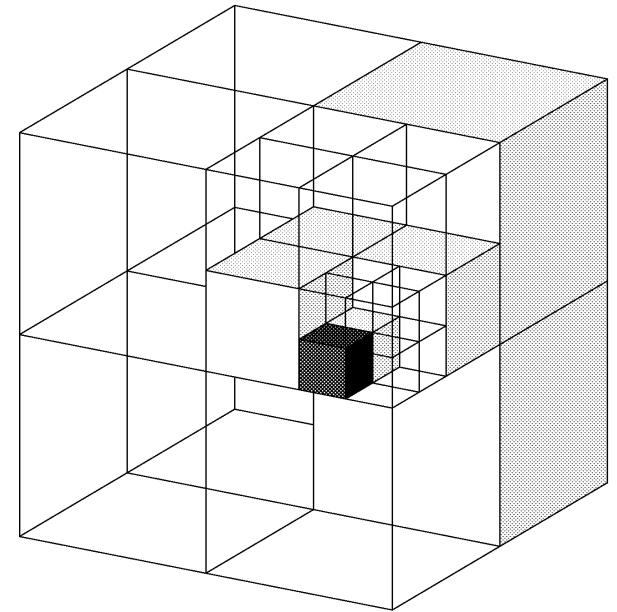
- Puede representar múltiples superficies por celda

- Contra:

- No representa áreas desconocidas
- No es una representación volumétrica sino una discretización en la dimensión vertical
- La localización en mapas MLS no es fácil de implementar

Representación basada en Octrees

- Estructura de datos basada en árboles
- Subdivisión recursiva del espacio en octantes
- Asignación de volumen a medida que se necesita
- “Grilla 3D inteligente”



Octrees

- **Pro:**

- Modelo completo 3D
- Probabilístico
- Resolución variable
- Eficiente en memoria



- **Contra:**

- La implementación es más compleja (memoria, actualizaciones, etc...)

OctoMaps

- Basado en **octrees**
- Representación **probabilística** y **volumétrica** de ocupación incluyendo espacios desconocidos
- Soporta acceso multi-resolución del mapa
- **Eficiente en memoria**
- **Archivo de mapa compacto**
- Implementaciones open-source: librerías C++ (<http://octomap.github.io/>)

Actualización probabilística del mapa

- Ocupación modelada como un **filtro de Bayes binario** recursivo [Moravec '85]

$$Bel(m_t^{[xyz]}) = \left[1 + \frac{1 - P(m_t^{[xyz]} | z_t, u_{t-1})}{P(m_t^{[xyz]} | z_t, u_{t-1})} \cdot \frac{P(m_t^{[xyz]})}{1 - P(m_{t-1}^{[xyz]})} \frac{1 - Bel(m_{t-1}^{[xyz]})}{Bel(m_t^{[xyz]})} \right]^{-1}$$

- Actualización eficiente usando notación **log-odds**

Actualización probabilística del mapa

- Recorte de valores (clamping) asegura actualización [Yguel '07]

$$Bel(m_t^{[xyz]}) \in [l_{\min}, l_{\max}]$$

- Acceso multi-resolución usando

$$Bel(n) = \max_{i=1\dots 8} Bel(n_i), n_i \in \text{children}(n)$$



0.08 m



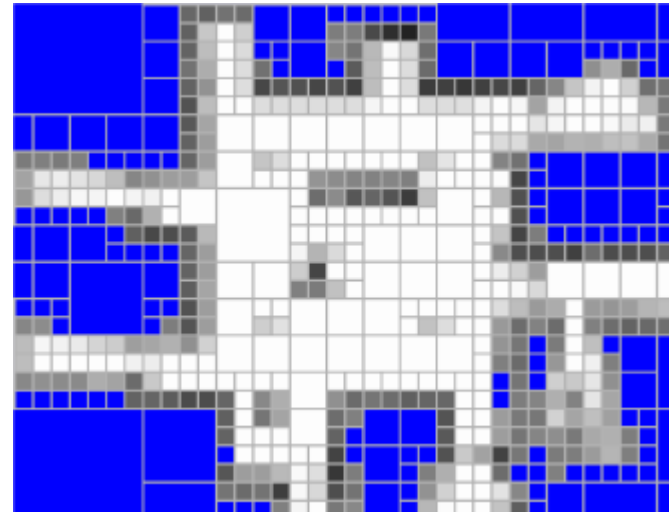
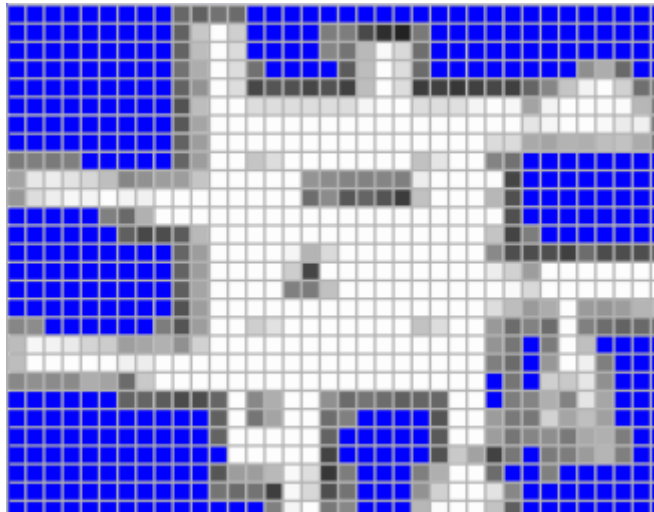
0.64 m



1.28 m

Compresión de mapa sin pérdidas

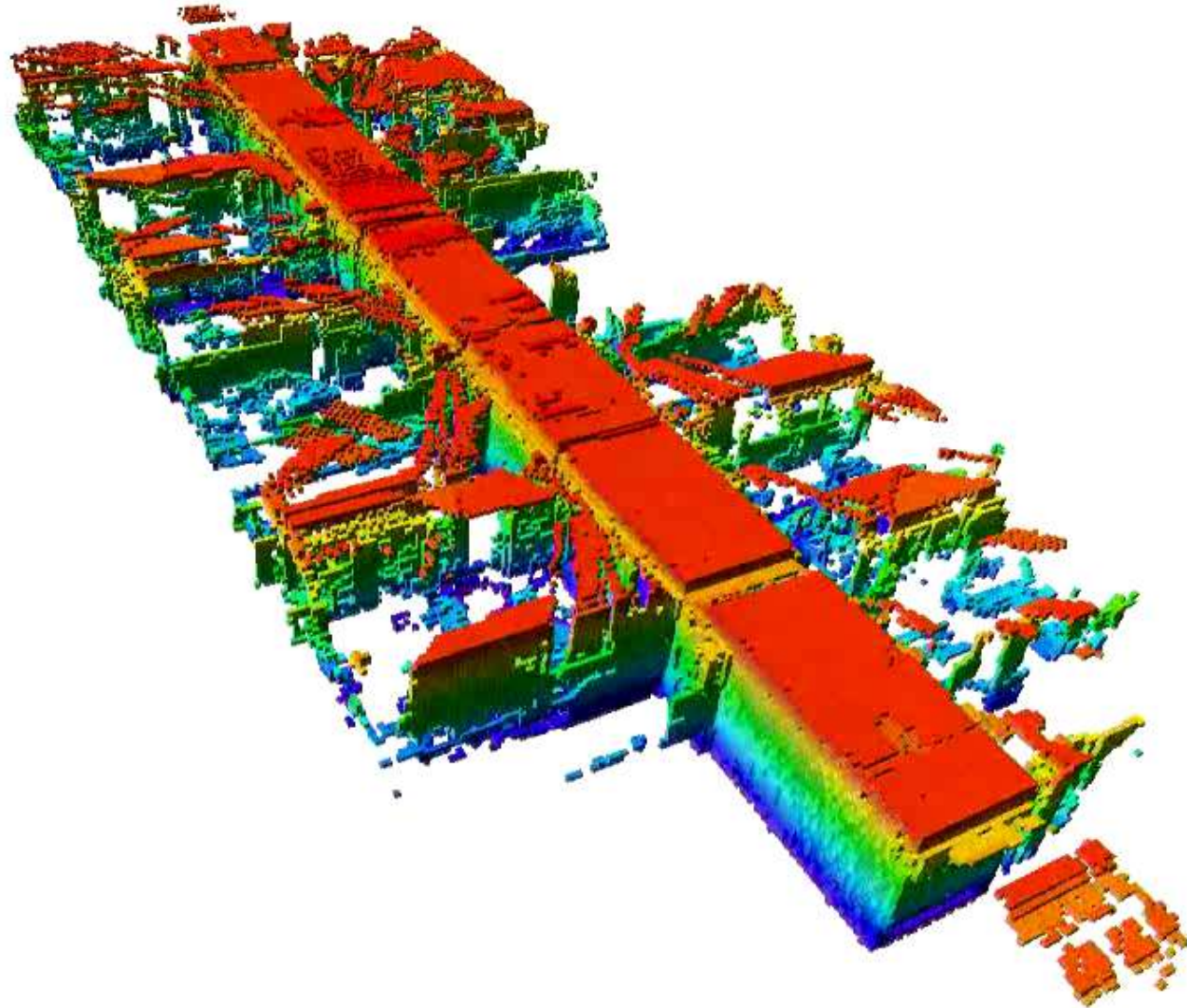
- Poda sin pérdida de nodos con hijos idénticos
- Permite altos niveles de compresión



[Kraetzschmar '04]

Video: Edificio de oficinas

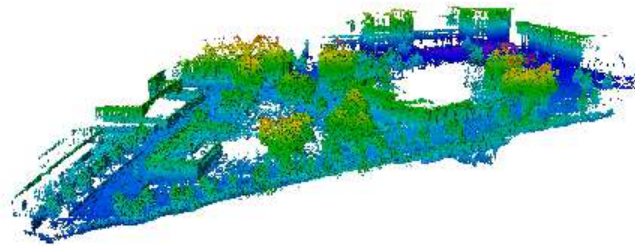
Univ. de Freiburg, Oficinas



Video: Grandes áreas

Campus Universitario (Freiburg)

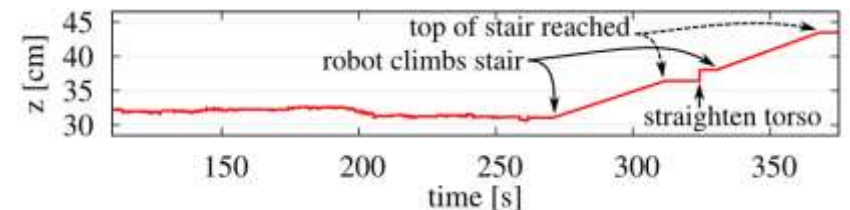
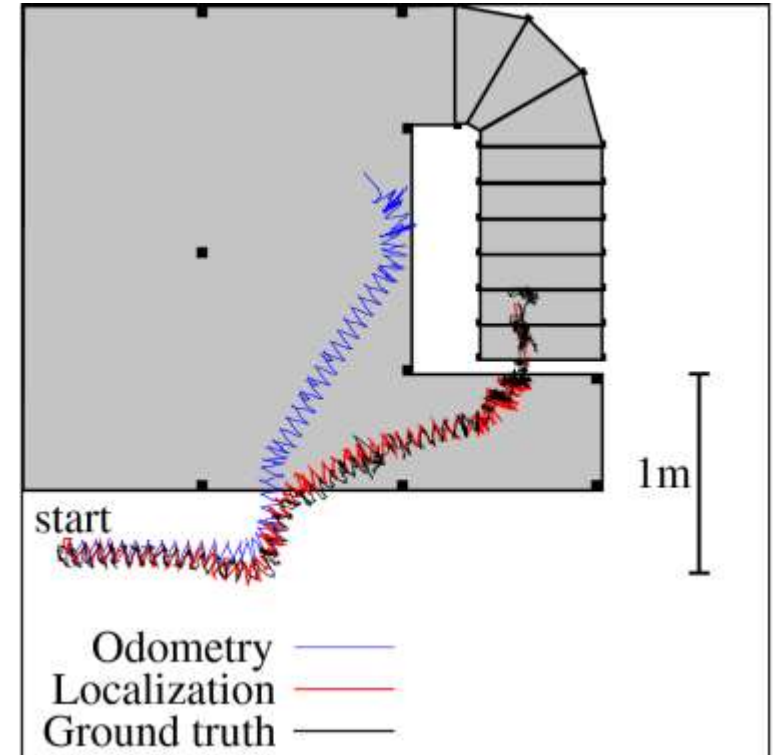
(292 x 167 x 28 m³, 20 cm de resolución)



Localización 6D con un humanoide



Objetivo: Estimación de pose precisa al caminar y subir escaleras

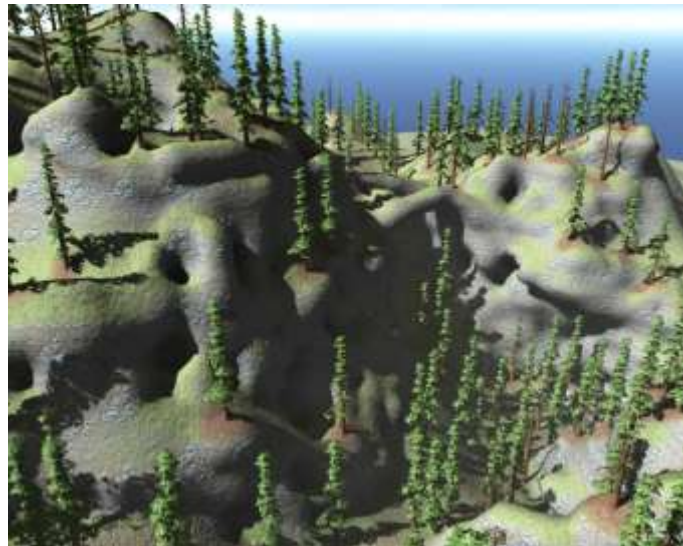


Video: localización con humanoides

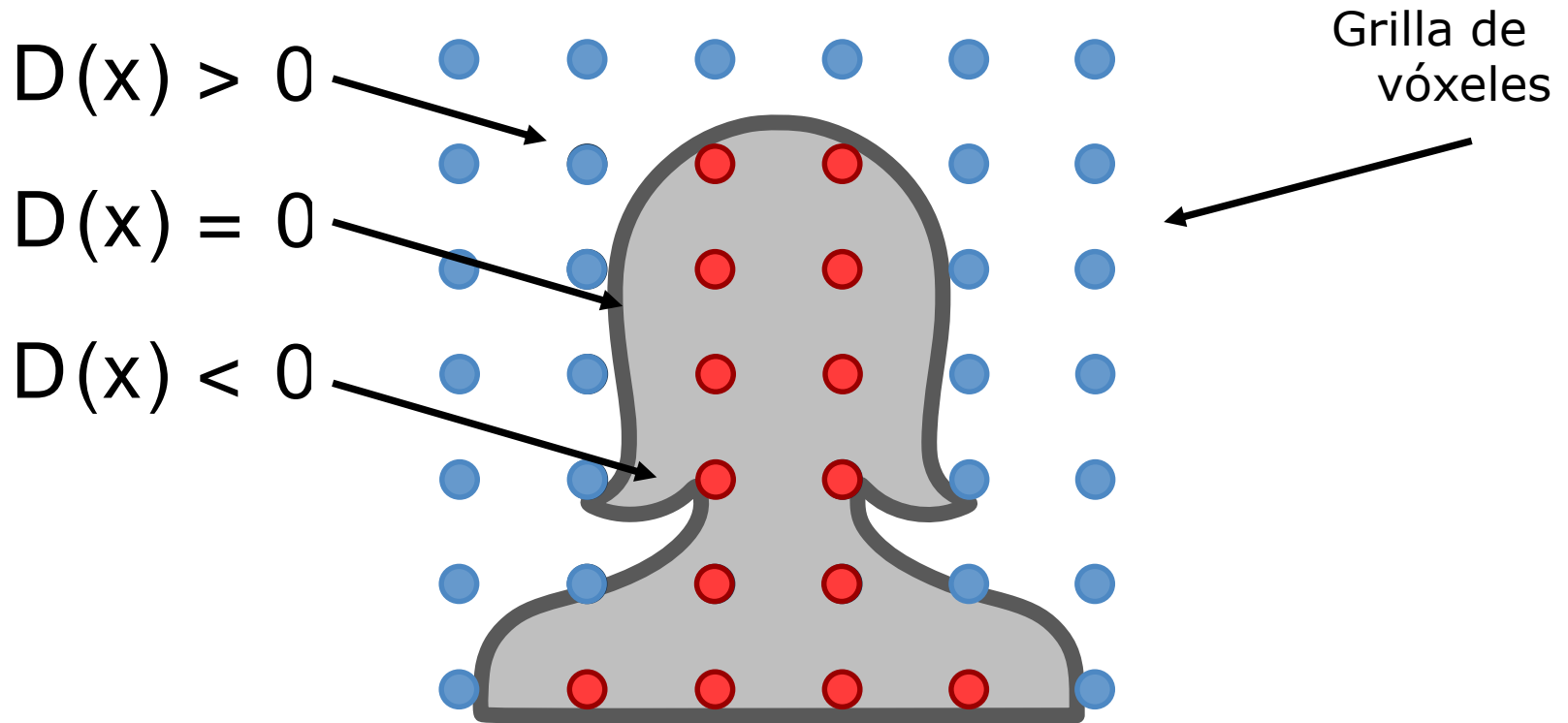


Representaciones de alta res.

- Cámaras RGBd:
 - Medición de distancia
- Alta resolución:
 - En general, mapeo de objetos, etc.



Función de distancia signada (SDF)

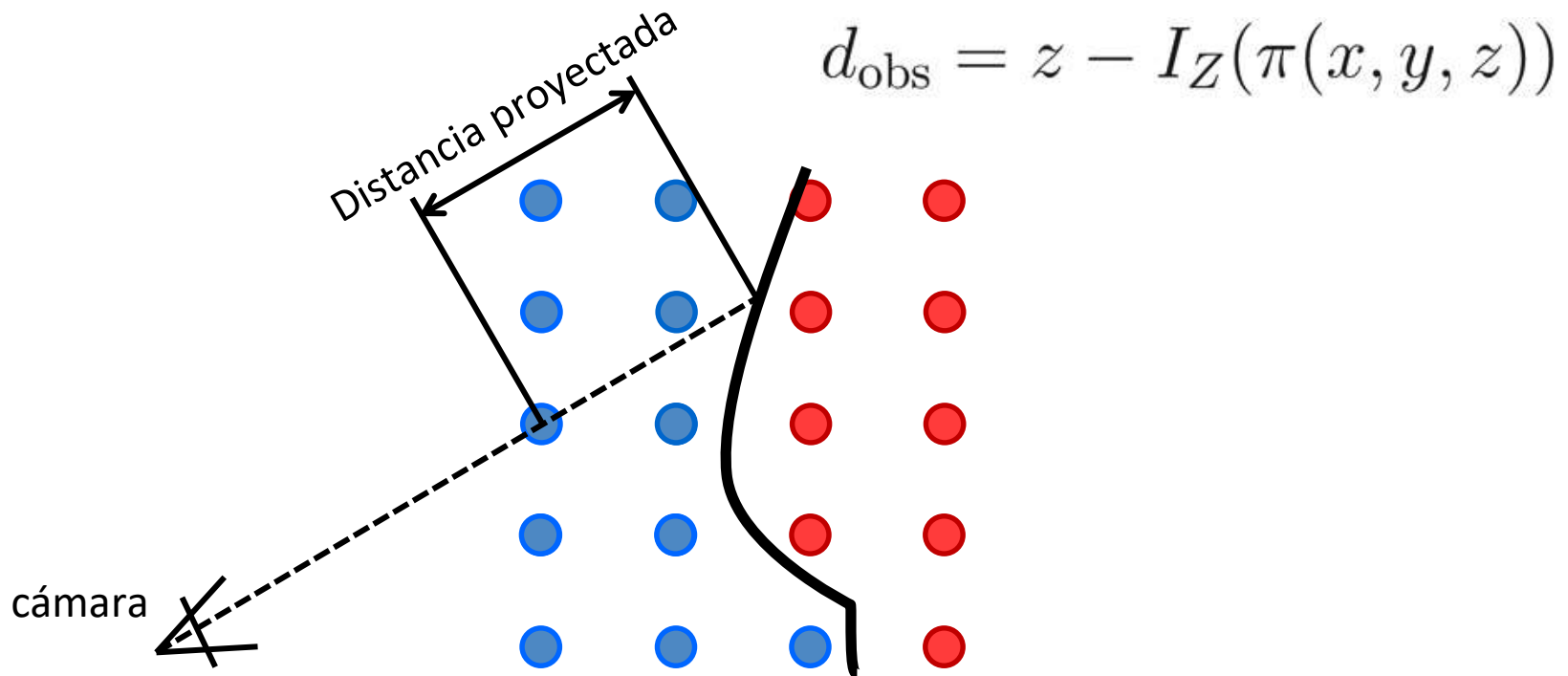


— Distancia de signo negativo (=adentro)

— Distancia de signo positivo (=afuera)

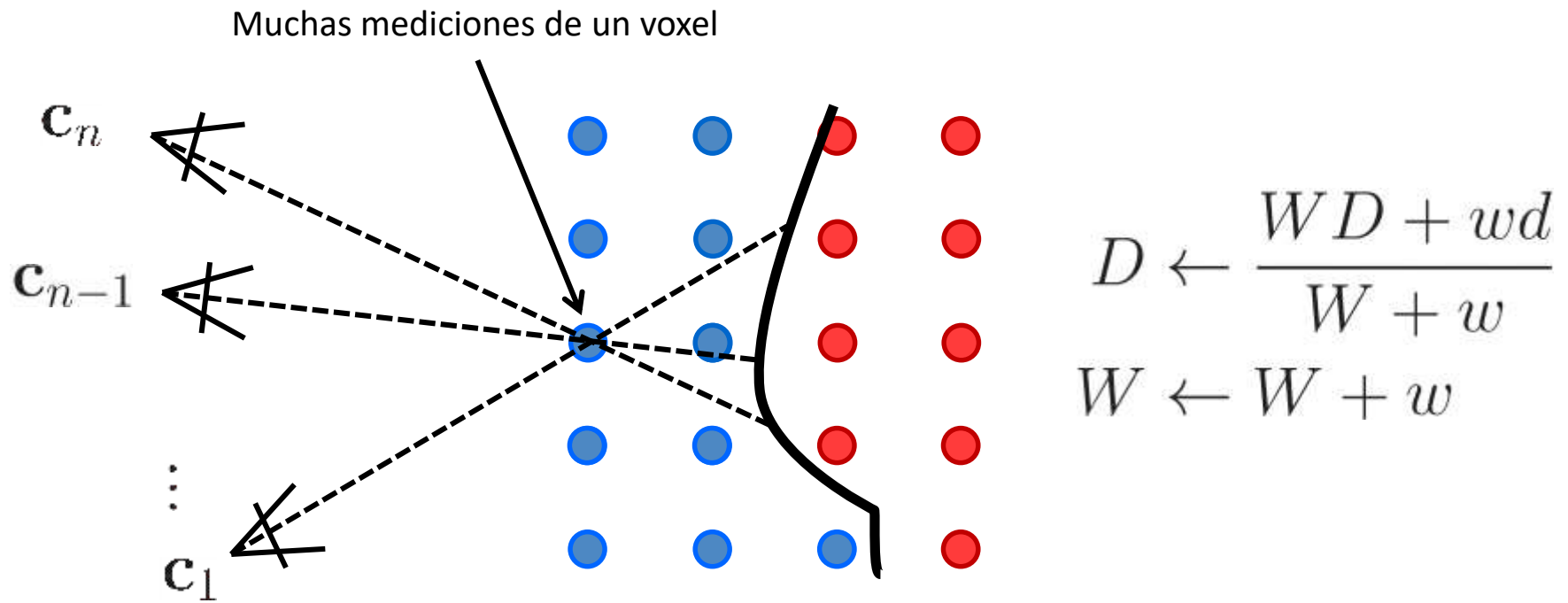
Función de distancia signada (SDF)

- Calcular SDF de una imagen de profundidad
- Calcular la distancia de cada voxel a la superficie observada
- Se puede hacer en paralelo para todos los vóxeles (\rightarrow GPU)
- Es muy eficiente si solo se considera un pequeño intervalo alrededor del punto de medición (truncamiento)



Función de distancia signada (SDF)

- Calcular el promedio pesado de todas las mediciones de cada voxel
- Suponiendo que se conoce la pose de la cámara



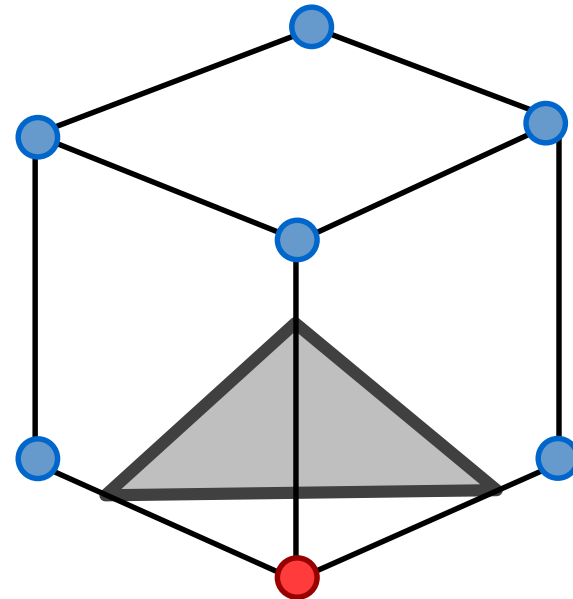
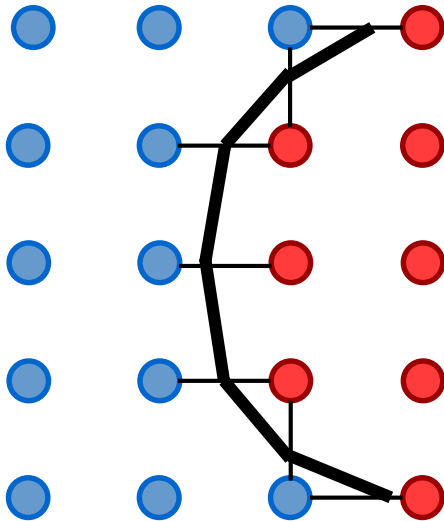
Visualización de campos de distancia signados

Métodos comunes de extracción de iso-superficies:

1. Ray casting (GPU, rápido)
Para cada pixel de la cámara, proyectar un rayo y buscar donde cruza el cero
2. Poligonización (CPU, lento)
por ejemplo, usando el algoritmo de "marching cubes"
ventaja: produce una malla de triángulos

Extracción de mallas usando “Marching Cubes”

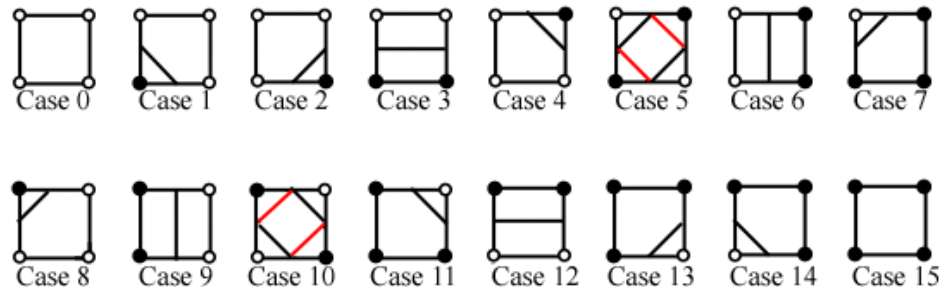
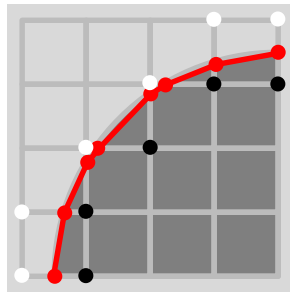
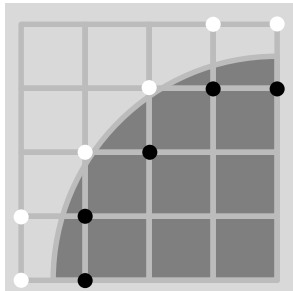
- Encontrar los cruces de cero en la función de distancia signada por interpolación



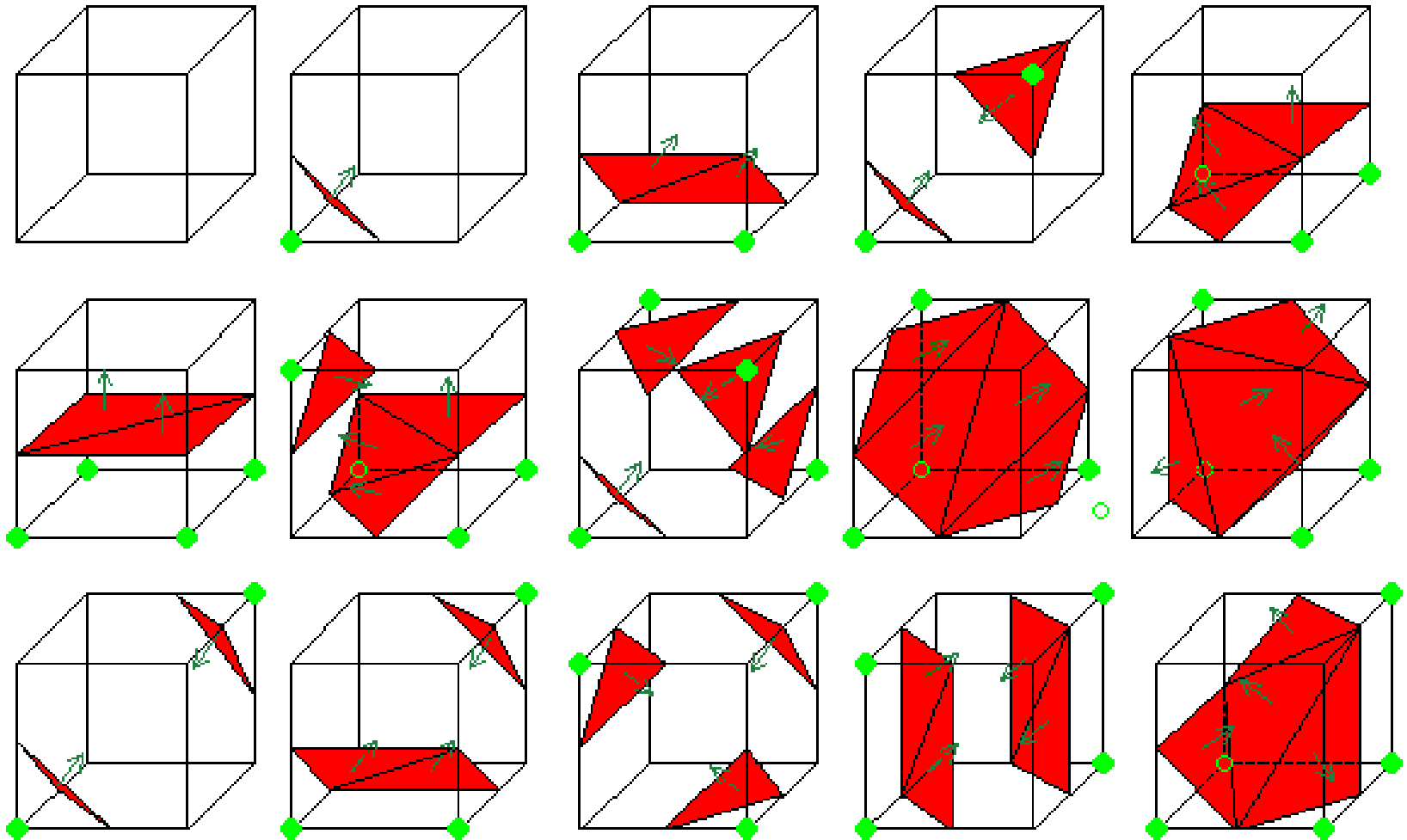
“Marching Cubes”

En el caso 2D: “**Marching squares**”

- Evaluar cada celda de forma separada
- Ver que aristas quedan adentro o afuera
- Generar triángulos según una tabla de 16 opciones
- Localizar vértices usando cuadrados mínimos



“Marching Cubes” (3D)



KinectFusion

- SLAM basado en ICP (más detalles luego) con métrica de punto-a-plano
- Función de distancia signada truncada (TSDF)
- Ray Casting

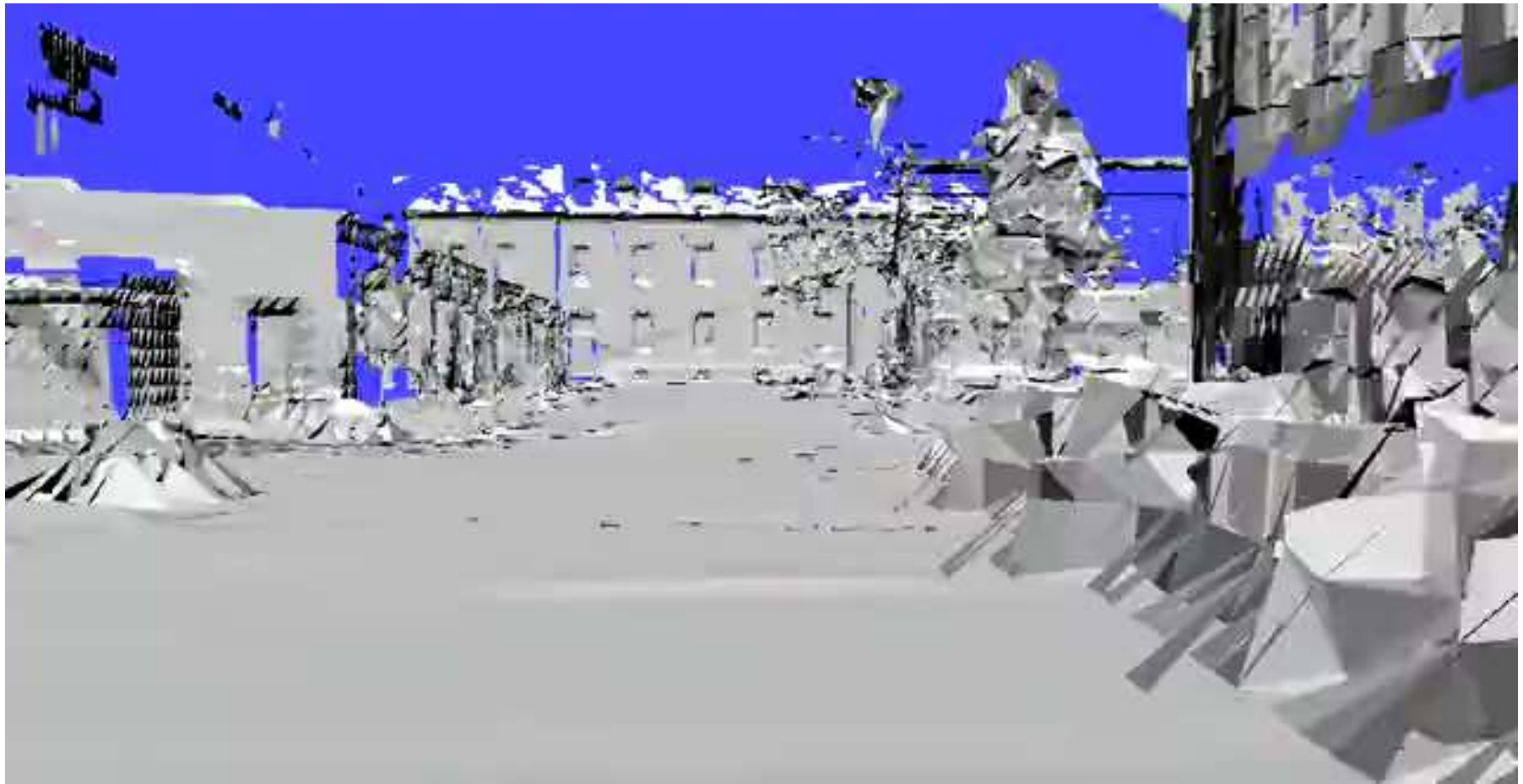


Ejemplo de aplicación



[Sturm, Bylow, Kahl, Cremers; GCPR 2013], cortesía de Jürgen Sturm]

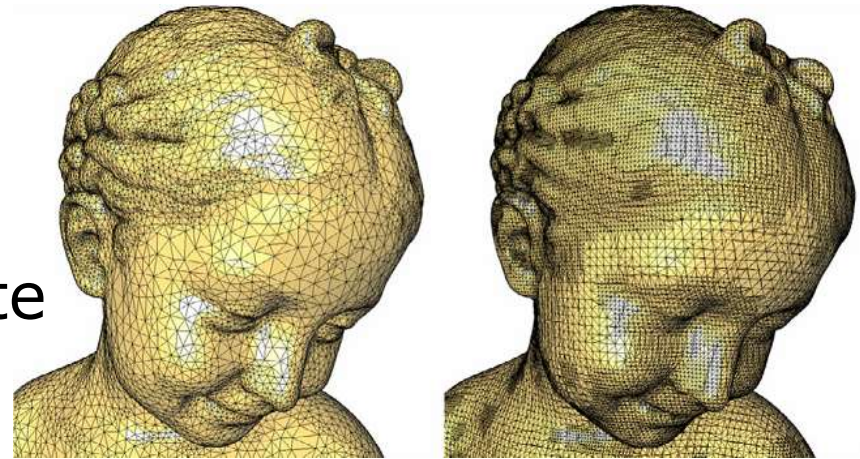
Ejemplo: Mapa 3D



Funciones de distancia signada

- **Pro:**

- Modelo 3D completo
- Resolución sub-pixel
- Implementación eficiente (tarjetas de video)



- **Contra:**

- Grilla de vóxeles consumen mucho espacio

Aplicación en robótica

Motion Primitives-based Path Planning for Fast and Agile Exploration Using Aerial Robots

Mihir Dharmadhikari, Tung Dang, Lukas Solanka, Johannes Loje,
Huan Nguyen, Nikhil Khedekar, Kostas Alexis



DARPA

The research effort for the results depicted is sponsored by the Defense Advanced Research Projects Agency. The presented content and ideas are solely those of the authors.

 **AUTONOMOUS
ROBOTS
LAB**

N



Resumen

- Diferentes **representaciones** de mapas 3D
- El modelo más apropiado depende de la aplicación
- Los modelos de **superficie** permiten análisis de **transitabilidad**
- Las representaciones de **vóxeles** permiten una representación **3D completa**
- **Octrees** son una representación probabilística. Son inherentemente multi-resolución.
- Funciones de **distancia signada** usan grillas 3D, pero permiten resoluciones sub-pixel de la superficie.