

RESUMEN TEORÍA VISIÓN ARTIFICIAL:

ÍNDICE DE CONTENIDOS:

TEMA 1: INTRODUCCIÓN

- 1.1 INTRODUCCIÓN**
- 1.2 CAPTURA DE UNA IMAGEN**
- 1.3 PROCESAMIENTO DE IMÁGENES**
- 1.4 ROBOT (COMPONENTES)**
- 1.5 SOFTWARE**

TEMA 2: FORMACIÓN DE LA IMAGEN

- 2.1 OJO HUMANO VS CÁMARA**
- 2.2 LENTES**
- 2.3 MODELOS**
- 2.4 FORMACIÓN DE LA IMAGEN**
- 2.5 IMAGEN DIGITAL**
- 2.6 ESPACIOS DE COLOR**

TEMA 3: TRANSFORMACIÓN DEL DOMINIO Y ESPACIAL

- 3.1 TRANSFORMACIÓN DEL DOMINIO**
- 3.2 TRANSFORMADA DE FOURIER**
- 3.3 TRANSFORMADA DEL COSENO**
- 3.4 TRANSFORMADA DE WAVELETS**
- 3.5 TRANSFORMACIÓN ESPACIAL Y PÍXEL A PÍXEL**
- 3.6 TRANSFORMACIONES DE VECINDAD**
- 3.7 TRANSFORMACIONES LÓGICAS**
- 3.8 TRANSFORMACIONES GEOMÉTRICAS**

TEMA 4: TRANSFORMACIÓN Y CORRECCIONES

- 4.1 TRANSFORMACIONES GEOMÉTRICAS**
- 4.2 TRANSFORMADAS ELEMENTALES Y SISTEMA DE COORDENADAS**
- 4.3 INTERPOLACIÓN**
- 4.4 INTERPOLACIÓN LINEAL**
- 4.5 INTERPOLACIÓN BICÚBICA**
- 4.6 SUPERMUESTREO**
- 4.7 REGISTRO Y TRANSFORMACIONES AFINES**
- 4.8 TRANSFORMACIONES RADIOMÉTRICAS**
- 4.9 SUAVIZADO**
- 4.10 HISTOGRAMA**
- 4.11 INTERPRETACIÓN**
- 4.12 EXPOSICIÓN**
- 4.13 CONTRASTE**
- 4.14 RANGO DINÁMICO**
- 4.15 PROPIEDADES**
- 4.16 ECUALIZACIÓN**
- 4.17 EXPANSIÓN**
- 4.18 CONTRACCIÓN, REALZADO Y DEFECTOS**
- 4.19 FILTRO HOMOMÓRFICO**
- 4.20 CORRESPONDENCIA Y COMPARACIÓN DE HISTOGRAMAS**

TEMA 5: BORDES, REGIONES Y PUNTOS DE INTERÉS

- 5.1 EXTRACCIÓN DE BORDES**
- 5.2 OPERADORES DE SOBEL Y OTROS EJEMPLOS**
- 5.3 CANNY (OBTENCIÓN DEL GRADIENTE)**
- 5.4 CANNY (SUPRESIÓN NO MÁXIMA)**
- 5.5 HISTÉRESIS DE UMBRAL**
- 5.6 LAPLACIANA Y DETECTORES**
- 5.7 CÓDIGOS DE CADENA**
- 5.8 MÍNIMOS CUADRADOS**
- 5.9 HOUGH LINEAL Y CIRCULAR**
- 5.10 RANSAC**
- 5.11 CONTORNOS DEFORMABLES**
- 5.12 BINARIZACIÓN MEDIANTE UMBRAL**
- 5.13 BINARIZACIÓN MEDIANTE OTSU**
- 5.14 BINARIZACIÓN MEDIANTE RIDLER-CALVARD**
- 5.15 COMPONENTES CONEXAS**
- 5.16 CRECIMIENTO Y DIVISIÓN**
- 5.17 DIVISIÓN Y FUSIÓN**
- 5.18 COLOR**
- 5.19 DESCRIPCIÓN DE REGIONES (PROPIEDADES)**
- 5.20 DESCRIPCIÓN DE REGIONES (TEXTURAS)**
- 5.21 MOMENTOS INVARIANTES DE HU**
- 5.22 HARRIS**
- 5.23 SIFT**
- 5.24 SIFT (DETECCIÓN DE EXTREMOS)**
- 5.25 SIFT (LOCALIZACIÓN DE PUNTOS DE INTERÉS)**
- 5.26 SIFT (FUNCIÓN EN EL EXTREMO Y HESSIANO DE 2X2)**
- 5.27 SIFT (ASIGNACIÓN DE ORIENTACIONES)**
- 5.28 SURF**

TEMA 6: OPERACIONES MORFOLÓGICAS

- 6.1 PRINCIPIOS BÁSICOS**
- 6.2 OPERACIONES BÁSICAS CON CONJUNTOS**
- 6.3 DILATACIÓN**
- 6.4 EROSIÓN**
- 6.5 APERTURA Y CIERRE**
- 6.6 CONTORNOS Y ESQUELETO**
- 6.7 RELLENADO DE HUECOS**
- 6.8 OPERACIONES EN IMÁGENES GRISES**
- 6.9 GRADIENTE MORFOLÓGICO**

TEMA 7: GEOMETRÍA Y CALIBRACIÓN

- 7.1 CALIBRACIÓN**
- 7.2 PROYECCIONES**
- 7.3 MODELO PIN-HOLE**
- 7.4 PARÁMETROS INTRÍNSECOS**
- 7.5 PARÁMETROS EXTRÍNSECOS**
- 7.6 MATRIZ DE PROYECCIÓN**
- 7.7 CALIBRACIÓN**

TEMA 8: VISIÓN 3D

8.1 INTRODUCCIÓN

8.2 LÁSER

8.3 RGB-D

8.4 VISIÓN ESTEREOSCÓPICA

8.5 GEOMETRÍA DE LOS SISTEMAS ESTÉREO

8.6 CORRELACIÓN Y CORRESPONDENCIA DE PUNTOS

8.7 MÉTODOS DE REGISTRO 3D

8.8 ICP

8.9 VARIANTES DE ICP

8.10 CÁMARA RGB-D KINECT

8.11 PCL (POINT CLOUD LIBRARY)

TEMA 9: FLUJO ÓPTICO

9.1 INTRODUCCIÓN

9.2 CAMPO DE MOVIMIENTO

9.3 FLUJO ÓPTICO Y CÁLCULO DEL FLUJO ÓPTICO

9.4 PROBLEMA DE LA APERTURA

9.5 MÉTODO DE LUCAS-KANADE

9.6 MÉTODO DE DIFERENCIAS

TEMA 10: RECONOCIMIENTO DE PATRONES

10.1 INTRODUCCIÓN

10.2 K-NN

10.3 SVM

10.4 RED NEURONAL MONOCAPA

10.5 RED NEURONAL MULTICAPA

10.6 ALGORITMO K-MEANS

10.7 MODELO SOM

10.8 ALGORITMO DE KOHONEN

10.9 MEDIDAS DE CALIDAD Y PRECISIÓN DEL MAPA

10.10 R-CNN

10.11 SSD Y YOLO

10.12 PROCESAMIENTO DE IMÁGENES EN YOLO

TEMAS 1: INTRODUCCIÓN

1.1 INTRODUCCIÓN

Visión: **Aristóteles** (saber qué hay y dónde mediante la vista), **Gibson** (recuperar de la información de los sentidos, como la vista, propiedades válidas del mundo exterior) y **Marr** (proceso que produce, a partir de las imágenes del mundo exterior, una descripción que es útil para el observador y que no tiene información irrelevante).

Aspectos importantes: Entiende la visión como un proceso computacional, la descripción que se obtiene depende del observador y es necesario reducir la información que no sea útil.

Actividades: Procesamiento de imágenes digitales (descripción y reconocimiento del contenido de una imagen digital) y visión computacional (pretende dotar a los ordenadores de la capacidad de poder simular la visión humana).

Dificultades: Cambios de iluminación y de escala, mimetización del entorno, oclusión, movimiento y pérdida de información (suelen percibir escenas tridimensionales pero las imágenes generadas son bidimensionales).

Nivel bajo: Trabaja directamente con los píxeles para extraer gradiente, profundidad, textura, color, etc.

Nivel intermedio: Agrupa píxeles para obtener bordes, líneas y regiones para segmentar la imagen.

Nivel alto: Interpreta los datos obtenidos en niveles inferiores y utiliza modelos / conocimiento a priori.

1.2 CAPTURA DE UNA IMAGEN

Dispositivo de captura: Dispositivo físico sensible a una determinada banda del espectro electromagnético que produce una señal eléctrica proporcional al nivel de energía detectado (CCD y CMOS).

Conversor A / D: Convierte la señal analógica obtenida por el dispositivo de captura en una señal digital.

Memoria de video: Memoria semiconductora (RAM) en la que se almacena la imagen digitalizada.

Procesador: Procesador de propósito general que permite operar sobre la imagen digital.

1.3 PROCESAMIENTO DE IMÁGENES

Objetivos: Mejorar la calidad visual de las imágenes para permitir la interpretación humana y extraer información de las imágenes en un formato entendible por el ordenador.

Digitalización: **Imagen analógica/continua** (imagen natural captada con una cámara, sensor o cualquier otro dispositivo), **imagen digital/discreta** (proyección de la imagen analógica para que pueda ser manipulada por ordenador), es la transformación de una imagen analógica a otra digital y es el primer paso para procesar imágenes digitales.

Problemas: Degradación de la imagen digitalizada por ruido o pérdida de definición de la imagen causado por calibración / enfoque de la cámara defectuosos, ruido producido por los sensores de captura, movimiento del dispositivo de captura o de la escena, transmisión defectuosa de la señal captada, perturbaciones aleatorias como la propagación de radiación en el medio de transmisión (aire), etc.

Preprocesamiento: Atenúa la degradación de la imagen para que las siguientes etapas tengan una probabilidad de éxito mayor, todo ello mediante operaciones como supresión de ruido o realce del contraste.

Segmentación: Extrae la información contenida en la imagen, cuya extracción se realiza mediante una descomposición de la imagen en unidades o partes que son homogéneas con respecto a una o más características y que tienen una fuerte relación con objetos o áreas del mundo real.

Problemas: Las partes u objetos que componen una imagen dependen de la aplicación, donde cada objeto de la imagen segmentada debe ser etiquetado para que pueda ser integrado dentro de una descripción de la imagen original.

Representación: Parametriza objetos o partes de ellos generados por la segmentación del paso anterior.

Problemas: Sistemas de coordenadas cartesianas y polares, códigos de cadena, etc.

Descripción: Extrae información (características o descriptores) de la representación elegida para permitir la posterior clasificación de los objetos elegidos (puntos más dominantes / significativos / relevantes del contorno, perímetro del contorno, área de una región, número de huecos, etc).

Reconocimiento: Clasifica los diferentes objetos de la imagen utilizando sus descriptores, donde aquellos que presenten descriptores semejantes se agrupan en una misma clase (personas, libros, tazas, plantas, etc).

Interpretación: Da un significado a los grupos de objetos reconocidos (localiza objetos estáticos / dinámicos en un mapa, esquivar obstáculos, sigue y detecta la posición del cuerpo).

} 1.4 ROBOT (COMPONENTES)

Robot: Sistema informático con sensores, actuadores y un computador cuya inteligencia reside en el software que hay que programar para conseguir los objetivos y lograr sensibilidad ante ciertas situaciones.

Sensores: Miden magnitudes físicas del entorno del robot (distancia, luz, etc), la percepción depende de los sensores del robot, el robot existe en el espacio de los sensores, los sensores son muy distintos de los biológicos, el diseñador tratará de “situarse” en el mundo del robot y el tipo de sensores dependerá de la tarea a realizar.

Actuadores: Un robot interacciona con el mundo a través de sus actuadores, le dotan de capacidad de movimiento o de hacer algo, los actuadores robóticos son muy distintos de los biológicos mediante locomoción (trasladarse de un lugar a otro) y manipulación (manejo de objetos) y dividen a grosso modo la robótica en dos campos (robots móviles y robots manipuladores / brazos).

Otros componentes: Controladores para todos los anteriores, computadores, comunicaciones con otros robots u ordenadores (redes) e interacción con humanos (interfaz de usuario, pantallas, botones y audio).

} 1.5 SOFTWARE

Software: Determina el comportamiento del robot, establece cómo se coordinan la percepción y la actuación y no hay una manera universalmente aceptada de programarlos.

Características: Lenguajes de bajo y alto nivel (ensamblador, C y C++), heterogeneidad (dispositivos hardware y encapsular funcionalidad), requisitos específicos, sistemas operativos, plataformas y simuladores.

TEMAS 2: FORMACIÓN DE LA IMAGEN

2.1 OJO HUMANO VS CÁMARA

Formación de la imagen a través del ojo humano: Capta el objeto real, la luz pasa a través del ojo, la imagen se proyecta invertida en la retina, el nervio óptico envía la imagen al cerebro y el cerebro gira la imagen.

Formación de la imagen a través de una cámara: Parte análoga (sujeto \Rightarrow luz \Rightarrow lente \Rightarrow filtro \Rightarrow sensor) y parte digital (proceso de conversión \Rightarrow almacenamiento de la imagen).

Similitudes: Ojo humano (iris, pupila, retina y cristalino) y cámara (diafragma, abertura, película y objetivo).

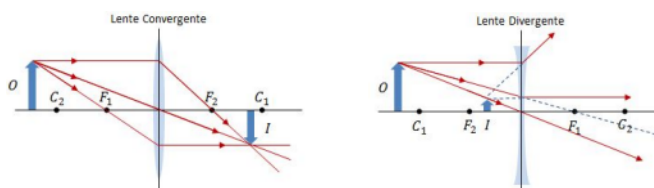
Nervio óptico: Punto ciego del ojo que el cerebro reemplaza con un relleno lo más parecido posible al área que rodea ese punto.

2.2 LENTES

Óptica: Rama de la física que estudia la luz, su propagación y su interacción con la materia, además de la reflexión, la refracción, la absorción en instrumentos ópticos como lentes, espejos y prismas, y los aspectos fisiológicos de la visión.

Convergentes / Convexas (lentes positivas): Los rayos de luz al atravesar una de estas lentes van a converger en un mismo punto (foco), son más gruesas en el centro que en los extremos, las imágenes son reales e invertidas, y establece que cuanto más cerca esté el objeto más grande será la imagen de éste, donde el foco está siempre detrás de la lente.

Divergentes / Cóncavas (lentes negativas): Los rayos de luz no convergen en un punto por la forma de la lente, sino que al atravesar la lente divergen en todas las direcciones, son más delgadas en la parte central que en los extremos, las imágenes de este tipo de lentes son virtuales (no se pueden recoger sobre ninguna pantalla), son siempre más reducidas que el objeto, el foco está delante de la lente y la imagen se encuentra entre el foco y la lente.



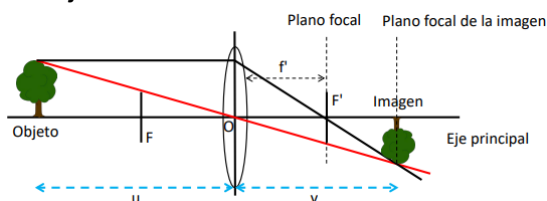
2.3 MODELOS

Modelo pinhole: Se pone una barrera con un pequeño orificio (apertura) entre el objeto y el sensor y se reduce el desenfoque por lo que se debe hacer que la apertura sea lo más pequeña posible (cámara estenopeica o pinhole), y cuando la luz de una imagen pasa a través de este agujero se forma una imagen invertida en el lado opuesto.

Problemas: Si se deja un agujero grande la imagen se verá borrosa, y cuanto más pequeño es el agujero por el que atraviesa la luz, más nítida y definida es la imagen, pero se hace más oscura.

Solución: Utilizar lentes.

Modelo de lente delgada: Los rayos que atraviesan el centro óptico no se desvían, concentra en un punto los infinitos rayos luminosos procedentes de un punto del espacio y todos los rayos paralelos al eje óptico convergen en un punto que es el foco y depende de la lente. A diferencia de la cámara estenopeica ideal, hay una distancia específica a la que los objetos están enfocados.



Fórmula de las lentes delgadas: Permite relacionar la posición del objeto y de la imagen con la distancia focal.

$$\frac{1}{F} = \frac{1}{v} - \frac{1}{u} \Rightarrow \frac{1}{\text{Distancia focal}} = \frac{1}{\text{Distancia de la imagen / imagen conjugada}} - \frac{1}{\text{Distancia del objeto / objeto conjugado}}$$

Problemas: Las lentes imperfectas pueden causar distorsión radial (desviaciones más notorias para los rayos que pasan por el borde de la lente), lo que se soluciona calibrando.

} 2.4 FORMACIÓN DE LA IMAGEN

Parámetros intrínsecos: Valores nominales proporcionados por los fabricantes (longitud focal, centro del plano de la imagen, tamaño del píxel en micras y la resolución horizontal-vertical del sensor en micras).

Parámetros extrínsecos: Aquellos que dependen de la posición del sistema de visión y que depende de cada escenario, así como su altura y su inclinación (ángulos de cabeceo \Rightarrow pitch, alabeo \Rightarrow roll y guiñada \Rightarrow yaw).

} 2.5 IMAGEN DIGITAL

Imagen digital: Está dividida en píxeles rectangulares capaces de captar la luz que les incide, donde el valor de cada píxel es proporcional a la cantidad de luz reflejada por la parte de la superficie del objeto que se proyecta sobre ese píxel, por lo que depende del material del objeto, la posición de las luces y el reflejo de otros objetos en la escena, lo que hace que la imagen real obtenida por el sensor sea en escala de grises.

Imagen a color: Necesita disponer de un sistema de filtros que cubran dicho sensor, cada píxel en el sensor será más o menos sensible a la radiación recibida en función de las longitudes de onda en el espectro visible y los filtros tendrán las longitudes de onda correspondientes al rojo, verde y azul.

Mosaico Bayer: Para cada píxel existe un filtro que limita la incidencia de la radiación a una única longitud de onda (roja, verde o azul), y para formar la imagen se recorre todo el mosaico en grupos de cuatro píxeles que aportan los datos de color a cada porción de imagen, cuya transformación da lugar a una imagen en el espacio de color RGB.

} 2.6 ESPACIOS DE COLOR

CMY: $[C \ M \ Y] = [255 \ 255 \ 255] - [R \ G \ B]$

Rojo: $[R] = [255 \ 0 \ 0]$

Verde: $[G] = [0 \ 255 \ 0]$

Azul: $[B] = [0 \ 0 \ 255]$

Blanco: $[W] = [255 \ 255 \ 255]$

Cyan: $[C] = [0 \ 255 \ 255]$

Magenta: $[M] = [255 \ 0 \ 255]$

Amarillo: $[Y] = [255 \ 255 \ 0]$

HSI: Modelo más utilizado por el ser humano para expresar el color en porcentajes.

H (Hue, matiz o tono, color / longitud de onda dominante):

$$H = \cos^{-1} \left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-B)^2 + (R-B)(G-B)]^{1/2}} \right]$$

S (saturación, cantidad de luz mezclada por el color dominante):

$$S = 1 - \frac{3}{R+G+B} [\min(R, G, B)]$$

I (Intensity, iluminación percibida):

$$I = \frac{1}{3} (R + G + B)$$

Normalización [0,1]: Cada valor RGB tiene que normalizarse dividiendo entre 255.

$$R = \frac{R}{255}, G = \frac{G}{255}, B = \frac{B}{255}$$

Si el color se encuentra en la mitad superior del triángulo de cromaticidad se debe restar el ángulo H a 360 para obtener el tono:

$$\text{si } (B > G), \text{ entonces } H = 360 - H$$

Paso del intervalo [0, 1] al intervalo [0, 255]:

$$H = \frac{H}{360} * 255; S = S * 255; I = I * 255$$

TEMA 3: TRANSFORMACIÓN DEL DOMINIO Y ESPACIAL

3.1 TRANSFORMACIÓN DEL DOMINIO

Transformación del dominio / frecuencia: Trabaja en el dominio espacial, donde los píxeles adyacentes representan partes adyacentes de la escena / componentes de frecuencia adyacentes.

Dominio no espacial: Permite identificar entidades que no se detectan tan fácilmente en el dominio espacial.

3.2 TRANSFORMADA DE FOURIER

Transformada de Fourier: Representa una imagen como una suma de exponenciales complejas de diferentes magnitudes, frecuencias y fases (mejora, análisis, restauración y compresión), muestra que una imagen está construida por la combinación de armónicos de frecuencias verticales y horizontales, por lo que a mayor frecuencia, más transiciones de luminancia en menos píxeles en la dirección determinada por la componente.

Características: Pasa del tiempo a la frecuencia, pinta en un gráfico de frecuencias la amplitud de las ondas simples, permite saber cuáles de las ondas son más grandes en amplitud e interpreta el contenido dinámico en forma de contenido en frecuencias.

Directa continua:
$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) * e^{-j2\pi(ux+vy)} dx dy$$

Inversa continua:
$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) * e^{j2\pi(ux+vy)} du dv$$

Directa discreta (creación del espectro):
$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) * e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

Inversa discreta:
$$f(x, y) = \frac{1}{M*N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) * e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$
, donde $e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$ es la función base y $F(u, v)$ son

los coeficientes de la transformada de Fourier (en todos los casos, $j = \sqrt{-1}$).

Términos: **Transformada de Fourier** ($F(u, v) = Re(u, v) + iIm(u, v)$),

amplitud ($|F(u, v)| = \sqrt{[Re(u, v)]^2 + [Im(u, v)]^2}$) y **fase** ($\phi(u, v) = \tan^{-1}[\frac{Im(u, v)}{Re(u, v)}]$).

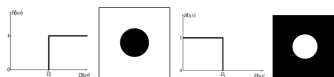
Separabilidad: Calcula la DFT de una función bidimensional como una combinación de dos DFTs, calculando primero una DFT sobre la variable de uno de los ejes y al resultado aplicarle de nuevo la DFT sobre la variable del otro eje.

Linealidad: La transformada de Fourier y su inversa son transformaciones lineales, es decir, poseen la propiedad distributiva respecto de la suma.

Traslación: Tanto la DFT como la transformada inversa, son periódicas de periodo N.

Simetría: Si la transformada es real, es simétrica conjugada ($|F(u, v)| = |F(-u, -v)|$), por lo que para calcular la magnitud de los puntos de un periodo completo, tan sólo necesitamos calcular los $(N/2) + 1$ primeros puntos, siempre y cuando el origen de la transformada esté centrado en el punto $(N/2, N/2)$. Para conseguir este movimiento del origen en la transformada, podemos aplicar la propiedad de traslación.

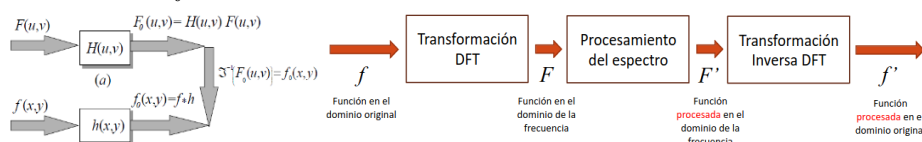
Rotación: Si rotamos la función $f(x, y)$ un ángulo determinado, la transformada de Fourier y su inversa también serán rotadas ese mismo ángulo.



Filtrado Paso Alto: $G(u, v) = H(u, v)$ (función de transferencia) * $F(u, v)$ (transformada de Fourier), donde $H(u, v)$ (si $D(u, v) \leq D_0 = 0$, si $D(u, v) > D_0 = 1$) y $D(u, v) = \sqrt{u^2 + v^2}$, conservando **altas frecuencias**.

Filtrado Paso Bajo: $G(u, v) = H(u, v)$ (función de transferencia) * $F(u, v)$ (transformada de Fourier), donde $H(u, v)$ (si $D(u, v) \leq D_0 = 1$, si $D(u, v) > D_0 = 0$) y $D(u, v) = \sqrt{u^2 + v^2}$, conservando **bajas frecuencias**.

Butterworth:
$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$



3.3 TRANSFORMADA DEL COSENO

Transformada del Coseno Discreta (DCT): Basada en la DFT, pero utilizando únicamente números reales.

Fórmula:
$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) * \cos\left[\frac{(2x+1)\pi u}{2N}\right] * \cos\left[\frac{(2y+1)\pi v}{2N}\right],$$

donde $u, v = 0, 1, 2, \dots, N-1$ y $\alpha(u)$ y $\alpha(v)$ ($\sqrt{1/N}$ para $u, v = 0$; $\sqrt{2/N}$ para $u, v = 1, 2, \dots, N-1$).

Compresión: Posee excelentes propiedades de compactación de energía lo que significa que concentra casi toda la energía en los primeros coeficientes de la transformada, la transformación es independiente de los datos, el algoritmo aplicado no varía con los datos que recibe, hay fórmulas para el cálculo rápido del algoritmo (FFT para la DFT), produce pocos errores en los límites de los bloques imagen y tiene una interpretación frecuencial de los componentes transformados que permite aprovechar al máximo la capacidad de compresión.

Pasos: Pasa la imagen del formato RGB al formato YIQ, que representa una división entre la luminosidad y la información sobre el color, donde el ojo humano es más sensible a la luminosidad que al color. Después, se realiza una transformación en la imagen mediante la DCT ya que permite comprimir la imagen según determinados patrones de frecuencia (método de compresión con pérdida de datos). Y por último, se codifica el conjunto de datos obtenidos al aplicar la DCT, usando un método que no produce pérdida (Huffman).

3.4 TRANSFORMADA DE WAVELETS

Transformada Directa (Paso Bajo): Haar $\left(\frac{1}{\sqrt{2}} [1, 1]\right)$, Daubechies $\left(\frac{1}{4\sqrt{2}} [1 - \sqrt{3}, 3 - \sqrt{3}, 3 + \sqrt{3}, 1 + \sqrt{3}]\right)$.

Transformada Directa (Paso Alto): Haar $\left(\frac{1}{\sqrt{2}} [-1, 1]\right)$, Daubechies $\left(\frac{1}{4\sqrt{2}} [-1 - \sqrt{3}, 3 + \sqrt{3}, -3 + \sqrt{3}, 1 - \sqrt{3}]\right)$.

Paso 1: Realizar la convolución de las filas con el filtro paso bajo y guardar los resultados.

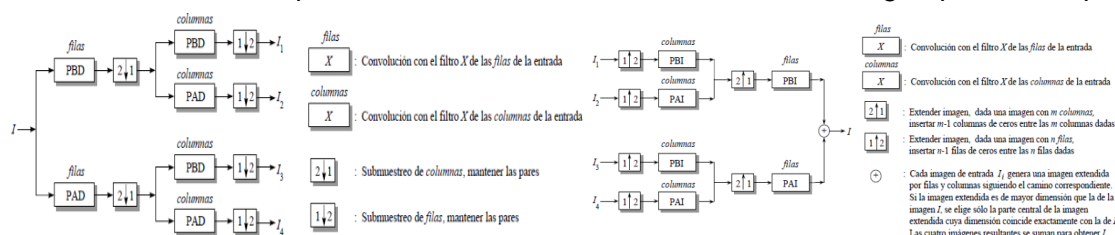
Paso 2: Realizar la convolución de las columnas con el filtro paso bajo, a partir de los resultados del paso 1. Obtener una imagen reducida tomando sólo un píxel de cada dos; esto nos genera una versión que denominamos paso bajo / paso bajo de la imagen.

Paso 3: Realizar la convolución del resultado del paso 1 con el filtro paso alto en las columnas. Obtener una imagen reducida tomando sólo un píxel de cada dos, obteniendo ahora una imagen paso bajo / paso alto.

Paso 4: Realizar la convolución de la imagen original con el filtro paso alto en las filas y guardar el resultado.

Paso 5: Realizar la convolución del resultado del paso 4 con el filtro paso bajo en las columnas. Obtener una imagen reducida tomando sólo un píxel de cada dos, obteniendo ahora una imagen paso alto / paso bajo.

Paso 6: Realizar la convolución de las columnas del resultado del paso 4 con el filtro paso alto. Obtener una imagen reducida tomando sólo un píxel de cada dos, obteniendo ahora una imagen paso alto / paso alto.



Transformada Inversa (Paso Bajo): Haar $\left(\frac{1}{\sqrt{2}} [1, 1]\right)$, Daubechies $\left(\frac{1}{4\sqrt{2}} [1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3}]\right)$.

Transformada Inversa (Paso Alto): Haar $\left(\frac{1}{\sqrt{2}} [1, -1]\right)$, Daubechies $\left(\frac{1}{4\sqrt{2}} [1 - \sqrt{3}, -3 + \sqrt{3}, 3 + \sqrt{3}, -1 - \sqrt{3}]\right)$.

3.5 TRANSFORMACIÓN ESPACIAL Y PÍXEL A PÍXEL

Transformación espacial: Consiste en modificar el contenido de la misma con un objetivo concreto, como puede ser el de prepararla para su posterior análisis.

Transformación píxel a píxel: Aplica una función sobre el valor de intensidad de cada píxel individualmente, donde la imagen obtenida tendrá el mismo tamaño que la original ($Out(i, j) = f(In(i, j))$).

Operador identidad: $Out(i, j) = In(i, j)$.

Operador inverso: $Out(i, j) = 255 - In(i, j)$.

Operador umbral: $Out(i, j) \Rightarrow 0$ si $In(i, j) \leq p$; 255 si $In(i, j) > p$.

Operador intervalo de umbral binario: $Out(i, j) \Rightarrow 0$ si $p_1 < In(i, j) \leq p_2$; 255 si $In(i, j) \leq p_1$ o $In(i, j) \geq p_2$.

Operador umbral de escala de grises: $Out(i, j) \Rightarrow In(i, j)$ si $p_1 < In(i, j) \leq p_2$; 255 si $In(i, j) \leq p_1$ o $In(i, j) \geq p_2$.

} 3.6 TRANSFORMACIONES DE VECINDAD

Transformación de vecindad: Los píxeles vecinos intervienen en la misma con distintas finalidades, tales como eliminar el ruido presente en la imagen para así suavizarla o con el fin de extraer bordes.

Características: El valor del píxel de salida se obtiene tras realizar sobre el píxel de origen una combinación de los valores de los píxeles vecinos, la transformación de la imagen se produce por la combinación de píxeles en lugar de realizar una transformación píxel a píxel, el vecindario de un píxel lo componen ocho valores correspondientes a las posiciones alrededor del píxel, el valor de intensidad del píxel de salida $Out(x, y)$ es la suma promediada de los valores de intensidad de los ocho vecinos alrededor del píxel de entrada $In(x, y)$, y la transformación de una imagen puede variar dependiendo de la influencia que cada uno de los vecinos ejerza, lo cual se consigue mediante una máscara que permite escoger de manera selectiva los vecinos que intervienen en la transformación y en qué medida contribuyen a la modificación del píxel central.

Convolución: $f_0 = f * h = \sum_m \sum_n f(m, n)h(x - m, y - n) = \sum_m \sum_n f(x - m, y - n)h(m, n)$

} 3.7 TRANSFORMACIONES LÓGICAS

Transformaciones lógicas: Considera que los valores numéricos de las imágenes pueden representarse a nivel de bits, cuyos valores se comprenden en el intervalo $[0, 255]$.

Operaciones habituales: **Lógicas** (and, or, xor y not), **transformaciones píxel a píxel para obtener imágenes binarias** $[0, 1]$ y **relacionales** ($<$, $>$, \leq y \geq).

} 3.8 TRANSFORMACIONES GEOMÉTRICAS

Transformaciones geométricas: Modifican las coordenadas espaciales de la imagen, ofreciendo aspectos de la misma bajo diferentes resoluciones, siendo en ocasiones necesario modificar también los valores de intensidad de los píxeles de la imagen original (escalado, rotación y traslación).

TEMA 4: TRANSFORMACIÓN Y CORRECCIONES

4.1 TRANSFORMACIONES GEOMÉTRICAS

Transformaciones geométricas: Modifican las coordenadas y la posición / rotación / escala / inclinación de una imagen. Este tipo de transformación no cambia el contenido de la imagen, donde la deformación se produce por un cambio en la posición de los píxeles que la componen.

Perspectiva: Visión de una figura desde distintos planos colocados entre la perspectiva y nuestra posición.

Espacio proyectivo: Forma en la que un ojo o una cámara proyecta una escena 3D sobre una imagen 2D.

Coordenadas homogéneas (August Ferdinand Möbius, 1837): Instrumento usado en geometría para describir un punto en el espacio proyectivo y su representación en un espacio dimensional se realiza a través de coordenadas de un espacio $(n + 1)$ - dimensional, que permite realizar transformaciones de manera sencilla al trabajar con matrices.

4.2 TRANSFORMADAS ELEMENTALES Y SISTEMA DE COORDENADAS

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & i_d \\ 0 & 1 & j_d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \theta & 0 \\ \theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

Escalado: $x = S_x i \mid y = S_y j$

Traslación: $x = i + i_d \mid y = j + j_d$

Rotación: $x = \cos(\theta)i - \sin(\theta)j \mid y = \sin(\theta)i + \cos(\theta)j$

Inclinación: $x = i + \theta * j \mid y = i * \theta + j$

Sistema de coordenadas: **Matrices** (filas i , columnas j) e **imágenes** (filas j , columnas i).

4.3 INTERPOLACIÓN

Interpolación: Mejora la calidad visual de una imagen al ajustar su tamaño sin perder demasiada información o calidad en el proceso.

Características: El valor de un píxel depende de otro píxel (o varios) cuya posición es calculada a través de un par de funciones f_1 (posición en i del resultado para el píxel original (i, j)) y f_2 (posición en j del resultado para el píxel original (i, j)), donde el tamaño de la imagen de salida puede ser distinto del tamaño de la imagen de entrada.

$$Out(i, j) = In(f_1(i, j), f_2(i, j))$$

Siendo In la imagen de entrada y $(0 \dots M_i, 0 \dots M_j)$ su tamaño, se tienen los siguientes **reflejos y rotaciones:**

Espejo horizontal: $Out(i, j) = In(M_i - i, j)$

Espejo vertical: $Out(i, j) = In(i, M_j - j)$

Rotación 90°: $Out(i, j) = In(j, M_j - i)$

Rotación 180°: $Out(i, j) = In(M_i - i, M_j - j)$

Rotación 270°: $Out(i, j) = In(M_i - j, i)$

Aumento x2: $Out(i, j) = In(i/2, j/2)$

Problema: Las imágenes son señales discretas, pero la transformación geométrica las trata como si fueran continuas (definidas en todo el plano), por lo que es necesario aplicar una interpolación.

Vecino más próximo: Cualquier punto del espacio toma el valor del píxel más cercano, la cual puede hacerse calculando la distancia euclídea con respecto a cada vecino, resultando en valores intermedios.

Características: **Ventajas** (muy sencilla y rápida de calcular) e **inconvenientes** (el efecto de cuadriculado es evidente, y da lugar a imágenes de poca calidad).

$$Out(i, j) = In(round(f_1(i, j)), round(f_2(i, j)))$$

4.4 INTERPOLACIÓN LINEAL

Lineal: En una dimensión, significa trazar una línea recta entre cada par de puntos consecutivos.

Si p es el punto a interpolar y se encuentra entre i y d ($i = \lfloor p \rfloor$ y $d = i + 1$), el valor interpolado en p será:

$$O(p) = (1 - a) * I(i) + a * I(d), \text{ siendo } a = p - i \text{ (media ponderada de } i \text{ y } d \text{ según la distancia } a)$$

Bilineal: Aplica dos interpolaciones lineales, una horizontal en las filas y otra vertical en todo el espacio.

Sea $p = (pi, pj)$ con $i = \lfloor pi \rfloor$, $d = i + 1$, $s = \lfloor pj \rfloor$, $r = s + 1$, $a = pi - i$ y $b = pj - s$, se tiene que:

$$O(pi, s) = (1 - a) * I(i, s) + a * I(d, s)$$

$$O(pi, r) = (1 - a) * I(i, r) + a * I(d, r)$$

$$O(pi, pj) = (1 - b) * O(pi, s) + b * O(pi, r)$$

$$O(pi, pj) = (1 - a) * (1 - b) * I(i, s) + a * (1 - b) * I(d, s) + (1 - a) * b * I(i, r) + a * b * I(d, r)$$

4.5 INTERPOLACIÓN BICÚBICA

Bicúbica (zoom): Basada en dos interpolaciones cúbicas (una horizontal en las filas existentes usando 4 puntos y otra vertical en todo el espacio usando los 4 puntos de la interpolación anterior), la cual consiste en trazar una cúbica entre los 4 puntos más próximos (2 a la izquierda y 2 a la derecha), donde el valor del punto se puede calcular como una media ponderada de los 4 x 4 píxeles circundantes.

$$I(x) = c_1 x^3 + c_2 x^2 + c_3 x + c_4$$

Sea p el punto a interpolar ($i = \lfloor p \rfloor$), se obtienen las 4 ecuaciones ($O(i - 1) = I(i - 1)$, $O(i) = I(i)$,

$O(i + 1) = I(i + 1)$ y $O(i + 2) = I(i + 2)$), se despejan c_1 , c_2 , c_3 y c_4 y se obtiene $O(p)$.

$$Out(pi, pj) = \sum_{n=-1..2} \sum_{m=-1..2} In(i + n, j + m) * P(n - a) * P(b - m)$$

Siendo $P(k) = 1/6(C(k + 2)^3 - 4 * C(k + 1)^3 + 6 * C(k)^3 - 4 * C(k - 1)^3)$ y $C(k) = \max(0, k)$

Reducción de k: $O(i, j) = In(k * i, k * j)$, donde $Out = \max(i)/k * \max(j)/k$, cuyo problema se debe a que los detalles son más pequeños que en la resolución de salida, donde los métodos de interpolación no pueden mejorar la situación ya que cada píxel es un muestreo ordenado de uno de entrada.

4.6 SUPERMUESTREO

Supermuestreo (reducción): Considera el píxel como un “volumen” con cierto área que aplica varias veces la transformación y toma la media, capaz de lograr un resultado de mucha más calidad evitando el problema del aliasing. Sin embargo, es mucho más costoso y requiere más cálculos, estableciendo que cuanto mayor es la reducción, mayor es el efecto del aliasing.

Alternativa: Aplicar primero un filtro de suavizado (por ejemplo, de media) y después un simple vecino más próximo, aunque esto sólo es aplicable en transformaciones que impliquen una reducción de resolución.

4.7 REGISTRO Y TRANSFORMACIONES AFINES

Registro: Consiste en encontrar los parámetros de la transformada (afín) para “descubrir” qué transformaciones ha sufrido una imagen con diferentes fines, y puede llevarse a cabo de múltiples maneras (transformaciones lineales, características y cambios de dominio / frecuencia).

Usos: Comparar dos imágenes, ver si una imagen está contenida en otra e integrar datos obtenidos de diferentes sensores (imágenes médicas).

Transformación afín: Mapean puntos en nuevos puntos aplicando una combinación lineal de operaciones de traslación, rotación y escalado.

Algoritmo: Paso 1 (convertir ambas imágenes a escala de grises), paso 2 (hacer coincidir las características de la imagen a alinear con la imagen de referencia y almacenar las coordenadas de los puntos clave correspondientes), paso 3 (los puntos clave son simplemente los pocos puntos seleccionados que se utilizan para calcular la transformación, y los descriptores son histogramas de los gradientes de la imagen para caracterizar la apariencia de un punto clave), paso 4 (hacer coincidir los puntos clave entre las dos imágenes), paso 5 (elegir las mejores coincidencias y eliminar las ruidosas), paso 6 (encontrar la transformada homomórfica) y paso 7 (aplicar esta transformación a la imagen no alineada original para obtener la imagen de salida).

4.8 TRANSFORMACIONES RADIOMÉTRICAS

Transformaciones radiométricas: Operaciones que tienden a eliminar cualquier anomalía en el valor radiométrico ND de la imagen ocurridas durante la adquisición de ésta. Estas anomalías son debidas al mal funcionamiento del sensor (líneas perdidas, píxeles aislados y bandeo por efecto de la atmósfera / dispersión).

Restauración de líneas / píxeles perdidos: Comprenden procedimientos estáticos simples, sustitución del píxel / línea y promedios de valores con banda auxiliar.

4.9 SUAVIZADO

Suavizado: Reduce las variaciones de intensidad entre píxeles vecinos, elimina pequeños detalles antes de la segmentación de un objeto de interés, rellena pequeños espacios y elimina el ruido modificando aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos.

Vecinos: Para cada píxel que se desea procesar, es necesario coger una ventana de vecindad / núcleo, sobre la que se aplicará una operación cuyo resultado asignará un valor al píxel de estudio (x,y).

$$Out(x, y) = h * In(x, y)$$

Parámetros: **Out** (imagen de salida procesada), **In** (imagen de entrada original) y **h** (operador que actúa sobre **In** y se define en un entorno de (x,y) también denominado núcleo).

$$Out(x, y) = \sum_{s=-1}^1 \sum_{t=-1}^1 h(s, t) * In(x + s, y + t)$$

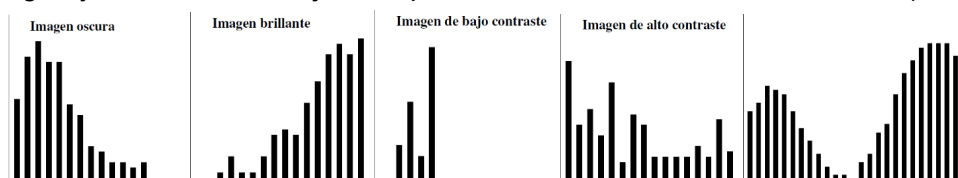
Límites de la imagen: Se puede aplicar la máscara añadiendo un marco de ceros de anchura o duplicando el borde de la imagen, lo que puede tener efectos no deseados (difuminación en los límites de la imagen), pero en general son poco significativos si la máscara es pequeña en relación con el tamaño de la imagen.

Filtro de media: Se reemplaza el valor de cada píxel por la media de los valores de los píxeles vecinos operando mediante convolución con una máscara determinada.

4.10 HISTOGRAMA

Histograma: Representación gráfica de las frecuencias relativas con las que aparecen los distintos colores en una determinada imagen.

Ventajas: Permite analizar la imagen utilizando la distribución de los niveles de gris, da información sobre el contraste de una imagen y sobre la exposición de la imagen al tomar una fotografía, permite su manipulación para mejorar la imagen y el número de objetos equivale al número de máximos locales (acumulaciones).



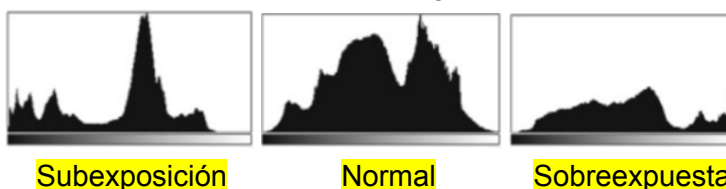
4.11 INTERPRETACIÓN

Ventajas: Muestra información sobre el contraste y el rango dinámico de la imagen, además de artefactos (áreas pequeñas de una imagen digital que tienen distorsión o un color diferente) resultantes de pasos de procesamiento de imágenes aplicados a la imagen, y examinando el tamaño y uniformidad de la distribución del histograma se puede determinar si la imagen está haciendo un uso efectivo de su rango de intensidades.

Desventajas: No ofrece información espacial, no codifica en donde está cada píxel, dado un histograma no es posible reconstruir la imagen y varias imágenes pueden tener el mismo histograma.

4.12 EXPOSICIÓN

Exposición: Cantidad de luz que llega el sensor de la cámara, además de ser lo que hace que una imagen salga clara u oscura, algo que se hace evidente en los histogramas.



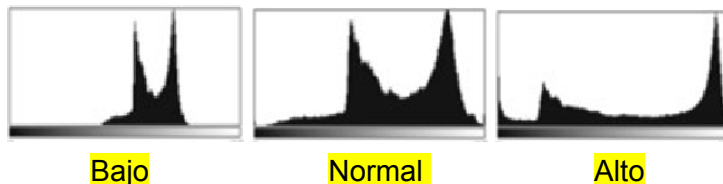
4.13 CONTRASTE

Contraste: Rango de valores de intensidad utilizados por una imagen, además de ser la diferencia que existe entre los valores máximo y mínimo de los píxeles de la imagen ($a = a_{max} - a_{min}$).

Imagen de contraste completo (full-contrast): Aquella que hace un uso efectivo de todo el rango de valores de intensidad disponibles siendo $a_{min} = 0$ el **color negro** y $a_{max} = K - 1$ el **color blanco**.

Contraste alto: Hay mucha diferencia entre el tono más negro y más blanco, con áreas extremadamente claras y oscuras.

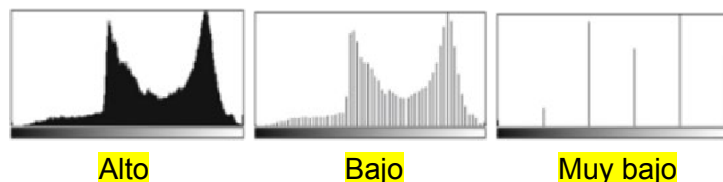
Contraste bajo: No presenta blancos o negros puros, sino una gama de tonos medios.



4.14 RANGO DINÁMICO

Rango dinámico: Número de valores de píxeles distintos en una imagen, que abarca todos los K valores de píxeles utilizables, en cuyo caso el rango de valores se utiliza por completo.

Rango dinámico máximo: Utiliza todos los valores de intensidad del rango ($a_{min} < a_{bajo} \dots a_{alto} < a_{max}$).



4.15 PROPIEDADES

Media:
$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_i \sum_j \frac{I(i,j)}{M}$$

Varianza:
$$\sigma^2 = \sum_{g=0}^{L-1} (g - \bar{g})^2 P(g)$$

Gris:
$$P(g) = \frac{N(g)}{M} = \frac{\text{Nivel de gris}}{\text{Máximo nivel de gris}}$$

Asimetría:
$$a = \sum_{g=0}^{L-1} (g - \bar{g})^3 P(g)$$

Entropía:
$$e = - \sum_{g=0}^{L-1} P(g) \log_2 [P(g)]$$

4.16 ECUALIZACIÓN

Ecualización: Mejora el contraste de la imagen y reparte de forma uniforme los valores del histograma.

Uniforme (1):
$$F(g) = g_{max} \sum_{g=0}^g p(g)$$

Uniforme (2):
$$F(g) = [g_{max} - g_{min}] P_g(g) + g_{min}$$

Exponencial:
$$F(g) = g_{min} - \frac{1}{\alpha} \ln[1 - P_g(g)]$$

Rayleigh:
$$F(g) = g_{min} + [2\alpha^2 \ln(\frac{1}{1 - P_g(g)})]^{1/2}$$

Hipercúbica:
$$F(g) = ([\sqrt[3]{g_{max}} - \sqrt[3]{g_{min}}] P_g(g) + \sqrt[3]{g_{min}})^3$$

Logaritmo hiperbólico:
$$F(g) = g_{min} [\frac{g_{max}}{g_{min}}] P_g(g)$$

4.17 EXPANSIÓN

Expansión: Distribuye las frecuencias en todo el ancho del histograma, donde se modifica el histograma de manera que se distribuyen las intensidades en la escala de valores disponibles para abarcar el máximo posible.

$$Out(i,j) = [\frac{\ln(I(i,j)) - \ln(I(i,j)_{MIN})}{\ln(I(i,j)_{MAX}) - \ln(I(i,j)_{MIN})}] [MAX - MIN] + MIN$$

4.18 CONTRACCIÓN, REALZADO Y DEFECTOS

Contracción: Disminuye el rango dinámico de la distribución de niveles de gris de la imagen.

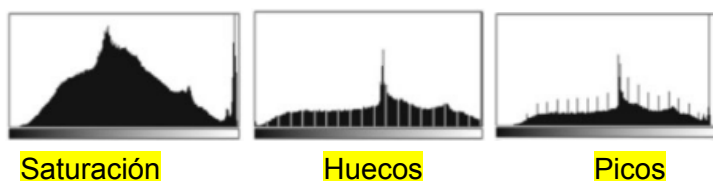
$$S_k = \frac{C_{max} - C_{min}}{r_{max} - r_{min}} [r_k - r_{min}] + C_{min}$$

Parámetros: C_{max} y C_{min} (valores deseados de la comprensión), r_{max} y r_{min} (máximo y mínimo nivel de gris de la imagen) y r_k (nivel del píxel evaluado).

Realzado: Imagen de entrada \Rightarrow (Filtrado Paso Bajo \Rightarrow Contracción de Histograma) \Rightarrow

Sustracción de imágenes \Rightarrow Expansión del Histograma \Rightarrow Imagen resultante.

Defectos: Pueden detectarse durante la adquisición o como resultado de un procesamiento posterior de imágenes, aunque normalmente las imágenes correctamente adquiridas y sin procesar tienen una distribución suave de intensidades. Los picos y huecos del histograma aparecen cuando la imagen ha sido procesada, es decir, cuando el rango de intensidades de la luz de la escena es mayor que el rango del sensor de la cámara aparecen picos en los extremos del histograma, y cuando se aumenta / reduce el contraste de la imagen, que se crean huecos / picos.



4.19 FILTRO HOMOMÓRFICO

Filtro homomórfico: Simultáneamente normaliza el brillo, comprimiendo el rango de intensidad e iluminación, y aumenta el contraste de una imagen.

$$Im(x, y) = i(x, y) * r(x, y)$$

Parámetros: Im (imagen), i (iluminación por bajas frecuencias) y r (reflectancia por altas frecuencias).

Filtro de paso alto: Transforma la ecuación anterior en el dominio de la frecuencia mediante log, por lo que se aplica la transformada de Fourier y un **filtro de paso alto H** para filtrar una **imagen N** . Posteriormente, se devuelve el dominio de la frecuencia al dominio espacial utilizando la transformada de Fourier inversa, y por último, se usa la función exponencial para eliminar el registro que se usa al principio al obtener la imagen:

$$\ln(Im(x, y)) = \ln(i(x, y)) * \ln(r(x, y)) \Rightarrow F(\ln(Im(x, y))) = F(\ln(i(x, y))) * F(\ln(r(x, y)))$$

$$N(u, v) = H(u, v) * F(v), \text{ donde } F(u, v) = I(u, v) + R(u, v)$$

$$n(x, y) = \text{inv}F(N(u, v)) \Rightarrow \text{newImage}(x, y) = \exp(n(x, y))$$

4.20 CORRESPONDENCIA Y COMPARACIÓN DE HISTOGRAMAS

Correspondencia de histogramas: Consiste en modificar la imagen B tomando como referencia la A a partir de los **histogramas h_A y h_B** , de los que se obtienen los valores de probabilidad acumulados, para así buscar para cada valor **$P(g_b)$** asociado con el nivel de gris **g_b** cuál es el valor más próximo **$P(g_a)$** a aquel que permita intercambiar el valor **g_b** por **g_a** en la imagen, obteniendo así una imagen **B_t** transformada:

$$P(g_a) = \sum_{i=0}^{g_a} P(g_i), P(g_b) = \sum_{i=0}^{g_b} P(g_i), \text{ donde } g_a \text{ y } g_b \text{ son los niveles de gris}$$

Correlación: $d(H_1, H_2) = \frac{\sum_{i=0}^N (H_1(i) - \overline{H_1})(H_2(i) - \overline{H_2})}{\sqrt{\sum_{i=0}^N (H_1(i) - \overline{H_1})^2 \sum_{i=0}^N (H_2(i) - \overline{H_2})^2}}$, donde $\overline{H_k} = \frac{1}{N} \sum_{j=0}^N H_k(j)$ y N son los bins del histograma

Intersección: $d(H_1, H_2) = \sum_{i=0}^N \min(H_1(i), H_2(i))$, cuanto mayor es el valor obtenido, mayor es la coincidencia.

Chi-Square: $d(H_1, H_2) = \sum_{i=0}^N \frac{(H_1(i) - H_2(i))^2}{H_1(i)}$, cuanto menor es el valor obtenido, mayor es el valor obtenido.

Distancia Bachattacharyya: $d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{H_1 * H_2 * N^2}} \sum_{i=0}^N \sqrt{H_1(i) * H_2(i)}}$

TEMA 5: BORDES, REGIONES Y PUNTOS DE INTERÉS

5.1 EXTRACCIÓN DE BORDES

Extracción de bordes / Bordes: Píxeles con alta variación en la escala de grises que se encuentran en la imagen a procesar generados por ruido, por lo que un buen algoritmo puede ser capaz de eliminarlo.

Primera derivada (gradiente): Calcula el gradiente que proporciona información sobre la variación de intensidad en la imagen, donde los bordes en una imagen suelen estar asociados con cambios bruscos en la intensidad y la detección de estos cambios se realiza buscando picos en la magnitud del gradiente.

Segunda derivada (laplaciano): Calcula el laplaciano que resalta las regiones donde hay cambios de dirección en el gradiente, lo que puede ayudar a identificar bordes más precisos y mejorar la localización de detalles finos.

Gradiente: Vector bidimensional perpendicular al borde donde el **vector G** apunta en la **dirección de variación máxima de f** en el **punto (x,y)** por unidad de distancia, además de la **magnitud G** , la **dirección Φ** y el **valor de umbral no negativo T** , que vienen dados por:

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x,y) \\ \frac{\partial}{\partial y} f(x,y) \end{bmatrix}$$

$$|G| = \sqrt{G_x^2 + G_y^2} \Rightarrow |G| \approx |G_x| + |G_y|, \text{ donde } G_x = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \text{ y } G_y = \frac{f(y+\Delta y) - f(y-\Delta y)}{2\Delta y}$$

$$g(x,y) = 1 \text{ si } G[f(x,y)] > T \text{ y } 0 \text{ si } G[f(x,y)] \leq T; \Phi(x,y) = \tan^{-1} \frac{G_y}{G_x}$$

5.2 OPERADORES DE SOBEL Y OTROS EJEMPLOS

Operadores de Sobel: Suavizan la imagen de tal manera que se elimina un poco de ruido si es que lo tiene y por consiguiente pueden desaparecer falsos bordes, para lo que se utilizan dos máscaras de 3x3, una para el gradiente horizontal G_x y una para el gradiente vertical G_y que se utilizan como operadores de Sobel.

$$G_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7); G_y = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Roberts

$$r_0 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Robinson

$$\begin{aligned} k_0 &= \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & k_1 &= \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & k_2 &= \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & k_3 &= \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\ 0^\circ & & 45^\circ & & 90^\circ & & 135^\circ \\ k_4 &= \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & k_5 &= \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & k_6 &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & k_7 &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\ 180^\circ & & 225^\circ & & 270^\circ & & 315^\circ \end{aligned}$$

Kirsch

$$\text{Prewitt} = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$

Prewitt (extensión)

$$\text{Sobel} = \begin{bmatrix} -1 & -1 & -1 & -2 & -1 & -1 & -1 \\ -1 & -1 & -1 & -2 & -1 & -1 & -1 \\ -1 & -1 & -1 & -2 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 \end{bmatrix}$$

Sobel (extensión)

5.3 CANNY (OBTENCIÓN DEL GRADIENTE)

Canny: Método para la detección de bordes basado en un criterio de **detección** (expresa el hecho de evitar la eliminación de bordes importantes y no suministrar falsos bordes), **localización** (establece que la distancia entre la posición real y la localizada del borde se debe minimizar) y **respuesta** (integra las respuestas múltiples correspondientes a un único borde).

Obtención del gradiente: Calcula la magnitud y orientación del vector gradiente en cada píxel.

Proceso: Aplica un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar ruido existente (hay que tener cuidado de no realizar un suavizado excesivo, pues se podrían perder detalles de la imagen), cuyo suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convolución de media cero y desviación estándar σ , y por último, se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes.

5.4 CANNY (SUPRESIÓN NO MÁXIMA)

Supresión no máxima: Logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr un píxel de ancho. Las dos imágenes generadas en el paso anterior sirven de entrada para generar una imagen con los bordes adelgazados.

Procedimiento: Se consideran cuatro direcciones (d_1, d_2, d_3, d_4) identificadas por las orientaciones de $0^\circ, 45^\circ, 90^\circ$ y 135° con respecto al eje horizontal, donde para cada píxel se encuentra la dirección que mejor se aproxime a la dirección de ángulo de gradiente. Si el valor de la magnitud de gradiente $E_m(i, j)$ es más pequeño que al menos uno de sus dos vecinos en la dirección d_k obtenida en el paso anterior se asigna el valor 0 a dicho píxel ($I_n(i, j) = 0$), en caso contrario se asigna el valor que tenga la magnitud del gradiente ($I_n(i, j) = E_m(i, j)$), obteniendo como salida una imagen con los bordes adelgazados (I_n).

5.5 HISTÉRESIS DE UMBRAL

Histéresis de umbral: Aplica una función de histéresis basada en dos umbrales, con lo que se pretende reducir la posibilidad de aparición de contornos falsos. A partir de la imagen con máximos locales por el ruido obtenida en el apartado anterior, se toma la orientación de los puntos del borde y dos umbrales (el primero más pequeño que el segundo) de dicha imagen. Por ello, se debe localizar el siguiente punto de borde no explorado $I_n(i, j)$ mayor al del segundo umbral (píxel de borde, $I_n(i, j) > t_2$), y a partir de esto, el punto debe seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal del borde siempre que sean mayores al primer umbral (no píxel de borde, $I_n > t_1$), marcando así todos los puntos explorados y almacenando la lista de dichos puntos en el contorno conectado, logrando así eliminar la unión en forma de Y de los segmentos que confluyen en un punto (G).

Gradiente de un píxel entre T2 y T1: Si el gradiente de un píxel está por encima de “T2” se declara un “píxel de borde”, si el gradiente de un píxel está por debajo de “T1” se declara como un “no-píxel de borde”, y si el gradiente de un píxel está entre “T2” y “T1” entonces hay que declararlo como un “píxel de borde” si y sólo si está conectado a un “píxel de borde” directamente o a través de píxeles entre “T1” y “T2”.

5.6 LAPLACIANA Y DETECTORES

Laplaciana (aproximación de Haralick y Shapiro): Ocurre tras el filtrado, donde un punto es considerado de borde si es menor / mayor que $-t / t$ y uno de sus ocho vecinos es mayor / menor que $t / -t$.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \text{ donde } \nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

Ventaja: Detecta mejor los bordes cuando las variaciones de intensidad de la imagen no son lo suficientemente abruptas.

Desventaja: Suelen ser muy sensibles al ruido, lo que unido a la incapacidad de detectar la dirección del borde, los hace muy limitados para el tratamiento digital de la imagen.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Puntos aislados

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

Líneas horizontales

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

Líneas verticales

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

Líneas diagonales

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Líneas diagonales

$$\frac{\partial}{\partial x} \equiv \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \frac{f}{\partial y} \equiv \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

5.7 CÓDIGOS DE CADENA

Propiedades de puntos de borde: Módulo y Dirección del Gradiente, Laplaciana y Varianza a Nivel de Gris.

Descripción de segmentos rectos: Dado el código de cadena de un contorno, se define el histograma del mismo de la forma $A = \{i \mid C(i)\}$ y $N = \text{cardinal}(A)$, donde $C(i)$ es la frecuencia de aparición del número determinado de código de cadena, i designa el correspondiente código de cadena y N es el cardinal de A (número de bandas del histograma).

Paso 1: Si el histograma tiene más de 4 barras ($N > 4$), la línea no es recta, ya que tiene al menos 4 orientaciones diferentes.

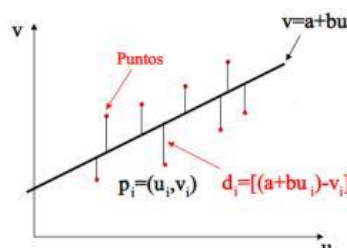
Paso 2: Si el histograma tiene una única barra ($N = 1$), la línea es puramente recta con alguna de las 8 direcciones cuantizadas.

Paso 3: Si el histograma tiene 2 barras ($N = 2$), se deben considerar 2 casos. Si las barras son adyacentes, hay que comprobar si la longitud máxima del código de menor frecuencia es menor que un umbral prefijado T , por lo que la línea se declara recta, y si la longitud máxima del código de menor frecuencia es mayor que un umbral prefijado T , la línea se declara no recta. Y por último, si las barras no son adyacentes, la línea se declara no recta, ya que contiene al menos 2 orientaciones diferentes, cuyos ángulos difieren al menos 90° .

Paso 4: Si el histograma tiene 3 barras ($N = 3$), se deben considerar dos casos. Si las 3 barras son adyacentes entre sí, la barra central es la más larga y la altura de la barra vecina más próxima es menor que un umbral T fijado por el usuario, por lo que la línea es declarada como recta. Por otro lado, si de las 3 barras que se tienen, 2 no son adyacentes, la línea se declara no recta, ya que posee dos orientaciones diferentes cuyos ángulos difieren al menos 90° .

5.8 MÍNIMOS CUADRADOS

Ajuste de líneas por mínimos cuadrados: Consiste en encontrar una recta tal que la suma de los cuadrados de las distancias verticales respecto a un conjunto de puntos $(u_{1..N}, v_{1..N})$ sea mínima, con el objetivo de obtener a y b tales que d^2 sea mínimo.

$$d^2 = \sum_{i=1}^N d_i^2 = \sum_{i=1}^N [(a + b * u_i) - v_i]^2$$


$$\begin{bmatrix} 1 & u_1 \\ 1 & u_2 \\ \vdots & \vdots \\ 1 & u_n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \rightarrow \begin{bmatrix} a \\ b \end{bmatrix} = U^* B; U^* = \text{pseudoinversa de } U$$

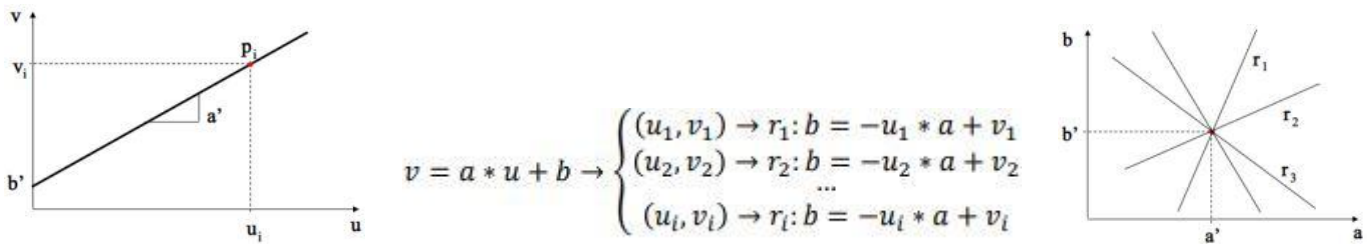
$$a = \frac{n \sum_{i=1}^n u_i v_i - (\sum_{i=1}^n u_i)(\sum_{i=1}^n v_i)}{\sum_{i=1}^n u_i^2 - (\sum_{i=1}^n u_i)^2} \quad y \quad b = \frac{(\sum_{i=1}^n u_i)(\sum_{i=1}^n v_i) - (\sum_{i=1}^n u_i)(\sum_{i=1}^n u_i v_i)}{\sum_{i=1}^n u_i^2 - (\sum_{i=1}^n u_i)^2}$$

El ajuste de mínimos cuadrados se puede extender para el ajuste de polinomios de cualquier grado:

$$d^2 = \sum_{i=1}^N d_i^2 = \sum_{i=1}^N [(a_1 + a_1 * u_i + a_2 * u_i^2 + \dots + a_m * u_i^m) - v_i]^2$$

5.9 HOUGH LINEAL Y CIRCULAR

Transformada de Hough: Detecta líneas y considera puntos (bordes) en el plano imagen, donde todas las líneas en el plano se pueden escribir como $v = a * u + b$. Además, si se considera un punto (u_i, v_i) sobre una determinada recta (a', b') $v_i = a' * u_i + b'$. Por otro lado, la ecuación de una recta cualquiera se escribe en el espacio de parámetros como $b = -u_i * a + v_i$, y por cada punto (u_i, v_i) situado sobre una recta en el plano imagen $(u - v)$ se obtiene una recta r_i en el espacio de parámetros.



Paso 1: Subdividir (cuantificar) a y b en intervalos apropiados, dentro de los rangos de variación (a_{min}, a_{max}) y (b_{min}, b_{max}) . Es importante decidir qué son intervalos apropiados.

Paso 2: Crea un array de acumulación $H(a, b)$, donde todas las celdas del array se inicializan a 0.

Paso 3: Por cada punto (u_i, v_i) del borde del plano imagen cuya magnitud supere un umbral, se obtienen los valores discretos (a, b) a partir de la ecuación:

$$b = -u_i * a + v_i \rightarrow \begin{cases} a = a_1 \rightarrow b_1 = -u_i * a_1 + v_i \\ a = a_2 \rightarrow b_2 = -u_i * a_2 + v_i \\ \dots \\ a = a_n \rightarrow b_n = -u_i * a_n + v_i \end{cases} \quad \forall a \in a_{min} < a < a_{max}$$

Paso 4: Incrementar todas las celdas $H(a, b)$ que resulten de la ecuación anterior.

Paso 5: Los máximos locales $H(a_i, b_i)$ se corresponden con puntos sobre rectas en el plano imagen, por lo que los parámetros de las rectas en el plano imagen serán a_i y b_i .

Problemas: Tanto la pendiente a como la ordenada en el origen b tienden a infinito conforme la recta se acerca a posiciones verticales. Así que, para evitar este problema, se usa la representación polar de la recta:

$$\rho = u * \cos \theta + v * \sin \theta, \text{ donde } 0 \leq \rho \leq \sqrt{(u_{max}^2 + v_{max}^2)} \text{ y } 0 \leq \theta \leq \pi$$

Nuevo algoritmo: Construye el acumulador de una forma similar al anterior, con la gran diferencia de que esta vez se obtendrán curvas sinusoidales en lugar de líneas rectas. Así, los N puntos colineales (u, v) pertenecientes a una recta en el plano imagen darán lugar a N curvas sinusoidales que se cortan en el espacio de parámetros en el punto (ρ_k, θ_k) .

$$\rho_k = u * \cos \theta + v * \sin \theta_k \text{ para los } N \text{ puntos } (u, v)$$

Características: Se puede simplificar notablemente el número de operaciones si se considera la dirección del vector gradiente en cada píxel de borde. Además, dado que la dirección del vector gradiente de un píxel de borde es perpendicular a éste, la recta que se busca debe tener una dirección próxima a la perpendicular del vector gradiente, restringiendo así el rango de orientaciones al evaluar cada punto de borde.

Detección de círculos: Similar al de las rectas, donde los círculos están definidos por un centro (a, b) y un radio r $((u - a)^2 + (v - b)^2 = r^2)$. Por lo que ahora, el espacio de parámetros es tridimensional con celdas cúbicas y acumuladores de la forma $H(a, b, r)$.

Procedimiento: Consiste en incrementar a y b y hallar r para que satisfaga la ecuación del círculo y pueda actualizar el valor del acumulador correspondiente a la célula asociada con el vector (a, b, r) .

5.10 RANSAC

RANSAC (RANdom SAMple and Consensus): Algoritmo de estimación robusta que permite hallar un modelo matemático a partir de datos contaminados con numerosos valores espurios que no se ajustan al modelo, teniendo en cuenta datos erróneos o outliers (pueden ser superiores al 50%). Supone que existe un porcentaje insuficiente de datos incorrectos (inliers) cuyo uso permite estimar los parámetros del modelo con la precisión requerida. Sin embargo, no es un algoritmo determinista, ya que el modelo proporcionado es válido solo con una cierta probabilidad, la cual aumenta a medida que se van realizando más iteraciones.

Idea: Selección aleatoria de muestras mediante un procedimiento probabilístico evaluando su ajuste al modelo, a la vez que busca el mejor modelo teniendo en cuenta todos los píxeles de contorno, incluso aquellos que no se ajustan al modelo buscado.

Proceso de selección de muestras: Siendo X los puntos necesarios para establecer los parámetros del modelo (2 para una línea, 3 para una circunferencia y 5 para una elipse), se calculan y evalúan los píxeles de la imagen del contorno original próximos al modelo calculado dentro de una tolerancia T, almacenándose de nuevo si el nuevo resultado es mejor.

Parámetros: **N** (número de iteraciones que tiene que realizar el algoritmo, que además está determinado por la probabilidad P de encontrar al menos una muestra no contaminada), **T** (tolerancia que determina cuando un píxel se ajusta al modelo, a partir de la cual se seleccionan 2 puntos y se calcula el conjunto del consenso formado por los puntos situados a una distancia inferior de la tolerancia) y **D** (número de inliers que garantiza que un modelo es válido).

$$N = \frac{\log(1-P)}{\log(1-\epsilon^x)}$$

Parámetros: ϵ^x (probabilidad de seleccionar una muestra con todos los inliers) y x (probabilidad deseada, la cual normalmente oscila entre 0,95 y 0,99).

Algoritmo: Seleccionar aleatoriamente una muestra de X píxeles, calcular los parámetros de la recta que pasa por los píxeles, determinar los píxeles de la imagen que se ajustan al modelo hallado con una tolerancia T, y guardar el modelo actual si el número de inliers es el más alto encontrado. Si el número de puntos cercanos al modelo es superior a un umbral D, el ajuste es válido, además de que conviene recalcular por mínimos cuadrados el modelo a partir de todos los inliers del resultado obtenido.

5.11 CONTORNOS DEFORMABLES

Minimizar: $E = \sum_{i=1}^N (\alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{imag}), \text{ donde } \alpha_i, \beta_i, \gamma_i \geq 0$

Continuidad del contorno: $E_{cont} = ||p_i - p_{i-1}||^2 \text{ con } p_i = (x_i, y_i)$

Suavidad del contorno: $E_{curv} = ||p_{i-1} - 2p_i + p_{i+1}||^2$

Atracción del borde: $E_{img} = - ||\nabla I||^2$

Paso 1: Definir el contorno inicial con un número de puntos N e inicializar α_i, β_i y γ_i .

Paso 2: Mientras el número de puntos que se mueven a una nueva localización sea mayor que un determinado umbral T_1 , se debe realizar el paso 3.

Paso 3: Desde $i=0$ hasta N se debe fijar E_{min} a un valor elevado, obtener $E_j = \alpha_i E_{cont,j} + \beta_i E_{curv,j} + \gamma_i E_{imag,j}$ para todos los puntos j en un entorno de vecindad del punto p_i de dimensión mxm, si $E_j < E_{min}$ entonces $E_{min} = E_j$ y $j_{min} = j$. Y por último, se debe mover el punto p_i a la localización j_{min} y contabilizar ese movimiento de posición.

Paso 4: Determinar las curvaturas desde $i=0$ hasta N-1 para obtener c_i . Si la curvatura es mayor que los vecinos ($c_i > c_{i-1}$ y $c_i > c_{i+1}$), mayor que un determinado umbral ($c_i > T_2$) y la magnitud del gradiente $p_i > T_3$ junto a la fuerza del borde supera un umbral, entonces se fija β_i a 0 para eliminar la contribución de la curvatura en el cómputo de la energía para la siguiente iteración.

$$c_i = ||\frac{u_i}{|u_i|} - \frac{u_{i+1}}{|u_{i+1}|}||, \text{ donde } u_i = (x_i - x_{i-1}, y_i - y_{i-1}) \text{ y } u_{i+1} = (x_{i+1} - x_i, y_{i+1} - y_i)$$

5.12 BINARIZACIÓN MEDIANTE UMBRAL

Binarización mediante umbral: Formada por dos objetos (fondo y otros) con probabilidades.

$$g(x, y) = 0 \text{ si } f(x, y) > T; 1 \text{ si } f(x, y) \leq T$$

$$P_1 p_1(z) = P_2 p_2(z), \text{ donde } p_1(z) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right], p_2(z) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right]$$

Haciendo $z=T$ (AT^2+BT+C): $A = \sigma_1^2 - \sigma_2^2$; $B = 2(\mu_1^2\sigma_2^2 - \mu_2^2\sigma_1^2)$; $C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln \frac{\sigma_2 P_1}{\sigma_1 P_2}$

Parámetros: P_x (probabilidad de la clase), p_x (distribución de la clase normal), m_x (media del lóbulo x) y σ_x (desviación estándar).

5.13 BINARIZACIÓN MEDIANTE OTSU

Binarización mediante Otsu: Pretende encontrar el umbral que maximiza la varianza inter-clase (dos clases distintas) y minimiza la varianza intra-clase (dos elementos de la misma clase).

Parámetros: $[0, L - 1]$ (imagen con niveles en el rango), C_0 (píxeles con los niveles entre $[0, \dots, k]$), C_1 (píxeles con los niveles entre $[k+1, \dots, L-1]$), k (umbral), L (nivel máximo de intensidad), $h(i)$ (número de píxeles en el nivel i), N (número total de píxeles de la imagen), p_i (probabilidad de pertenencia a cada nivel de gris), ω (probabilidad de pertenencia) y μ (nivel medio de la clase en el nivel k).

$$p_i = \frac{h(i)}{N}$$

$$\omega_0(k) = Pr(C_0) = \sum_{i=0}^k p_i$$

$$\omega_1(k) = Pr(C_1) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega_0(k)$$

$$\mu_0(k) = \sum_{i=0}^k iPr(i|C_0) = \sum_{i=0}^k ip_i/\omega_0(k)$$

$$\mu_1(k) = \sum_{i=k+1}^{L-1} iPr(i|C_1) = \sum_{i=k+1}^{L-1} ip_i/\omega_1(k)$$

Varianza inter-clase: $\sigma_B^2(k) = \omega_0(k) * \omega_1(k) * (\mu_1(k) - \mu_0(k))^2$

Varianza inter-clase del umbral óptimo k^* : $\sigma_B^2(k^*) = \max_{0 \leq k < L} [\sigma_B^2(k)]$, donde $k^* = [0, 255]$

Binarización píxel a píxel del umbral óptimo: $Out(i, j) = 0 \text{ si } In(i, j) \leq k^* ; 1 \text{ si } In(i, j) > k^*$.

5.14 BINARIZACIÓN MEDIANTE RIDLER-CALVARD

Binarización mediante Ridler-Calvard ($k = 0$): Calcula el valor medio de la imagen $T(k) = \mu$, cuyo valor determina dos clases x_1 y x_2 formadas por píxeles cuya intensidad es menor y mayor, respectivamente.

Binarización mediante Ridler-Calvard ($k = 1$): Determina los valores medios de cada clase μ_1 y μ_2 , obteniendo como resultado $T(k) = (\mu_1 + \mu_2)/2$. El nuevo umbral determina la existencia de otras dos nuevas clases para cada iteración, las cuales se repiten siempre que $|T(k + 1) - T(k)| \geq \varepsilon$.

5.15 COMPONENTES CONEXAS

Componentes conexos: Asigna una etiqueta con un valor numérico (4 u 8) a todos los píxeles de un mismo objeto o región, es decir, a todos aquellos que están unidos entre sí.

Algoritmo recursivo: Por cada punto del vecino que pertenezca a un objeto y no tenga etiqueta, se le asigna una etiqueta y se rellena un punto con dicha etiqueta. Mientras que para cada punto de la imagen que no tenga una etiqueta, se le asigna una nueva y se rellena un punto con dicha etiqueta.

Algoritmo iterativo: Se inicializa cada punto con una etiqueta distinta y se repite hasta que no haya cambios. Después, por cada línea y sus respectivos puntos, se encuentra la etiqueta más pequeña entre la del punto y sus vecinos, y si la etiqueta no es menor, se cambia.

Algoritmo clásico: Basado en una tabla de equivalencias que sólo da dos pasadas a través de la imagen.

5.16 CRECIMIENTO Y DIVISIÓN

Crecimiento y división: Procedimiento que agrupa los píxeles o subregiones de la imagen en regiones mayores basándose en un criterio prefijado. Normalmente se empieza con “puntos semilla” (valor de 255) para una determinada región, añadiendo aquellos píxeles vecinos que cumplan una propiedad específica, teniendo en cuenta la información sobre conectividad o adyacencia de la imagen en el crecimiento de regiones. Y por último, es importante mencionar la condición de parada.

Criterios de aumento: Diferencia en valor absoluto con un píxel semilla menor que 65, y 8-adyacencia con algún píxel de la región.

5.17 DIVISIÓN Y FUSIÓN

División y Fusión: Consiste en subdividir la imagen inicial en un conjunto de regiones disjuntas dentro de las cuales se realiza una nueva subdivisión o fusión entre ellas, dependiendo de si se verifican las condiciones prefijadas.

Árbol cuaternario (formación): Define un test de homogeneidad, subdivide la imagen en cuatro cuadrantes disjuntos, calcula la medida de homogeneidad para cada cliente, fusiona dos regiones si la condición de homogeneidad se verifica para la unión de las mismas, y por último, si una región no verifica la condición, se vuelve a subdividir en cuatro cuadrantes y se repite el proceso hasta que todas las regiones pasen el test de homogeneidad.

Características: Condición **P** (la diferencia en valor absoluto entre un píxel y la media de la región debe ser inferior a 2), división en cuatro cuadrantes disjuntos las regiones que no cumplen la condición, es decir,

$P(R_i) = \text{Falso}$, fusionar las regiones adyacentes para las cuales su unión cumple la condición

$P(R_i \cup R_j) = \text{Verdadero}$, y parar cuando no sea posible realizar más fusiones ni divisiones.

5.18 COLOR

Extracción de regiones mediante color: Descompone la imagen original en sus tres canales RGB, creando tres nuevas imágenes (**RegR**, **RegG** y **RegB**). Por lo que, dado un **umbral T** para cada **píxel (i,j)**:

$\text{Si } R(i, j) > G(i, j) \ \& \ R(i, j) > B(i, j) \ \& \ R(i, j) > T \Rightarrow \text{RegR}(i, j) = 255$

$\text{Si } G(i, j) > R(i, j) \ \& \ G(i, j) > B(i, j) \ \& \ G(i, j) > T \Rightarrow \text{RegG}(i, j) = 255$

$\text{Si } B(i, j) > R(i, j) \ \& \ B(i, j) > G(i, j) \ \& \ B(i, j) > T \Rightarrow \text{RegB}(i, j) = 255$

5.19 DESCRIPCIÓN DE REGIONES (PROPIEDADES)

Agujeros: El número de agujeros de una imagen 2D coincide con el número de componentes conexos del fondo de la imagen menos uno, suponiendo que nuestra imagen está enmarcada por un cuadrado de píxeles blancos.

Número de Euler: Imagen binaria 2D definida como el número de componentes conexas (negras) menos el número de agujeros/huecos.

Área: Número de píxeles de una región A.

Perímetro: Número de píxeles de borde $P = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$

Compatibilidad: P^2/A

Centroide: Centro de gravedad definido por $\bar{x} = 1/A \sum_i x_i$ y $\bar{y} = 1/A \sum_i y_i$

Elongación o razón de aspecto: Longitud/ancho, cuadrado = 1.

Ejes mayores y menores: Poseen una elipse envolvente con una determinada orientación.

Redondez: Razón entre el área y el eje mayor al cuadrado.

Compacidad: Cociente entre la raíz cuadrada del área y el eje mayor.

$$S_x = \sum x_i; S_y = \sum y_i; S_{xx} = \sum x_i^2; S_{yy} = \sum y_i^2; S_{xy} = \sum x_i y_i$$

$$\phi = \tan^{-1} \left\{ \frac{M_{xx} - M_{yy} + \sqrt{(M_{xx} - M_{yy})^2 + 4M_{xy}^2}}{2M_{xy}} \right\}, \text{ donde } M_{xx} = S_{xx} - \frac{S_x^2}{A}, M_{yy} = S_{yy} - \frac{S_y^2}{A} \text{ y } M_{xy} = S_{xy} - \frac{S_x S_y}{A}$$

5.20 DESCRIPCIÓN DE REGIONES (TEXTURAS)

Media: $\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_i \sum_j \frac{I(i,j)}{M}$

Varianza: $\sigma^2 = \sum_{g=0}^{L-1} (g - \bar{g})^2 P(g)$

Asimetría: $a = \sum_{g=0}^{L-1} (g - \bar{g})^3 P(g)$

Matriz de concurrencia: Dada una imagen, se define un operador de posición de píxel dentro de la imagen $P \equiv (d \cos \theta, d \sin \theta)$ y una matriz A de dimensiones K x K cuyo elemento a_{ij} es el número de veces que los píxeles cuya intensidad z_i aparecen en la posición especificada por P en relación a puntos de intensidad z_j con $1 \leq i, j \leq k$.

a_{ij} : Número de veces que un píxel de intensidad $z_{i-1} = i - 1$ aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad $z_{j-1} = j - 1$.

Matriz de concurrencia de nivel de gris (c): Se define dividiendo cada elemento de la matriz A por n (número total de pares de puntos de la imagen que satisfacen P), donde cada elemento de la matriz (c_{ij}) es una estimación de la probabilidad compuesta de que un par de puntos que satisfagan P tengan valores (z_i, z_j). Y por último, ya que C depende de P, es posible detectar la presencia de unos patrones de textura dados eligiendo adecuadamente el operador de posición P.

Probabilidad máxima: $\max(i, j)(c_{ij})$

Energía: $\sum_i \sum_j |c_{ij}|^2$

Correlación: $\frac{1}{\sigma_x \sigma_y} \sum_i \sum_j (i - \mu_x)(j - \mu_y) c_{ij}$

Entropía: $-\sum_i \sum_j c_{ij} \log(c_{ij})$

Uniformidad: $\sum_i \sum_j c_{ij}^2$

Inercia: $\sum_i \sum_j (i - j)^2 c_{ij}$

Homogeneidad local: $\sum_i \sum_j \frac{1}{1+(i-j)^2} c_{ij}$

Momento de distinción de elementos de orden k: $\sum_i \sum_j (i - j)^k c_{ij}$

Momento inverso de distinción de elementos de orden k: $\sum_i \sum_j \frac{1}{(i-j)^k} c_{ij}$, donde $i \neq j$

Parámetros: $\mu_x = \sum_i i \sum_j c_{ij}$; $\mu_y = \sum_j j \sum_i c_{ij}$; $\sigma_x^2 = \sum_i (i - \mu_x)^2 \sum_j c_{ij}$; $\sigma_y^2 = \sum_j (j - \mu_y)^2 \sum_i c_{ij}$

5.21 MOMENTOS INVARIANTES DE HU

Momentos invariantes de Hu (1962): Dada una imagen digital binaria (1 objeto y 0 fondo) se define el momento de orden p + q como $m_{pq} = \sum u \sum v u^p v^q$ donde el sumatorio se toma sobre todas las coordenadas (u, v) de puntos de la región objeto ($m_{00} = \sum u \sum v 1 = \text{superficie en píxeles de un objeto}$), donde si $f(u, v) = 1, \forall (u, v) \in R$ y $f(u, v) = 0, \forall (u, v) \notin R$.

Momentos invariantes de Hu: Momentos centrales invariantes a la posición/traslación, donde \bar{u} y \bar{v} son las coordenadas del centroide o centro de masas del objeto.

$$\mu_{pq} = \sum u \sum v (u - \bar{u})^p (v - \bar{v})^q, \text{ donde } \bar{u} = \frac{m_{10}}{m_{00}} = \frac{\sum \sum u}{\sum \sum 1} \text{ y } \bar{v} = \frac{m_{01}}{m_{00}} = \frac{\sum \sum v}{\sum \sum 1}$$

Ángulo que forma el eje de mínima inercia con el eje u: $\theta = \frac{1}{2} \tan^{-1} \frac{2(m_{00}m_{11} - m_{10}m_{01})}{(m_{00}m_{20} - m_{10}^2) - (m_{00}m_{02} - m_{01}^2)}$

Momentos invariantes ante escalado: $\eta_{pq} = \frac{\mu_{pq}}{(\eta_{00})^\gamma}$, donde $\gamma = \frac{p+q}{2} + 1$, para $p + q = 2, 3, \dots$

Momentos invariantes a traslaciones / rotaciones / cambios de escala: $\phi_n = \text{abs}(\ln(\text{abs}(\eta_n)))$.

5.22 HARRIS

Harris: Basados en la búsqueda de esquinas muy poco susceptibles a cambios de rotación y escala, donde cada esquina se caracteriza por ser una región de la imagen con cambios de intensidad en diferentes direcciones, y pueden ser **flat** (sin cambios en todas las direcciones), **edge** (sin cambios en las direcciones del borde) y **corner** (cambios significativos en todas las direcciones).

$$\frac{Tr(H)}{det(H)} < \frac{(r+1)^2}{r}, \text{ donde } r \text{ es la relación entre curvaturas principales } (r = 10)$$

Matriz de estructura Hessiana (H): Contiene las derivadas parciales segundas de la función de intensidad de la imagen, donde I_x e I_y son las derivadas parciales primeras de la imagen con respecto a x e y.

$$H = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Matriz de autovalores de H (λ_1, λ_2): Proporciona información sobre la estructura local de la imagen en un punto específico, por lo que se utiliza la función de respuesta en el contexto del operador de Harris.

$$R = det(H) - k * (tr(H))^2$$

Parámetros: $det(H)$ (determinante de $H(\lambda_1, \lambda_2)$), $tr(H)$ (traza de H, suma de los términos de su diagonal $(\lambda_1 + \lambda_2)$), k (constante de ajuste, normalmente 0,04) y $R > umbral$ (regiones consistentes con esquinas).

5.23 SIFT

SIFT (Scale-Invariant Feature Transform): Detecta y describe características locales en imágenes que presentan diversas propiedades que las hacen adecuadas para hallar la correspondencia entre distintos puntos de vista de un objeto o escena, además de detectar puntos característicos en una imagen para luego describirlos mediante un histograma orientado de gradientes.

Características: Invariantes a escalado y rotación, y son parcialmente invariantes a cambios de iluminación y de puntos de vista de una cámara 3D. Cada punto está definido por su vector de características de 128 elementos, del cual se obtiene información de la posición en coordenadas de la imagen, escala y orientación dominante de la región alrededor de dicho punto.

5.24 SIFT (DETECCIÓN DE EXTREMOS)

Detección de extremos en la escala-espacio: Establecen los candidatos iniciales para los puntos de interés.

Proceso: Cada punto de la muestra se compara con sus ocho vecinos de la imagen en la que se encuentra y también con los nueve vecinos de la imagen que se encuentra a una escala superior e inferior, donde el punto se selecciona como un posible extremo sólo si el valor de D es mayor o menor que sus 26 vecinos totales y cuyo coste es bajo, ya que la mayoría de los candidatos son descartados después de unas pocas comparaciones.

5.25 SIFT (LOCALIZACIÓN DE PUNTOS DE INTERÉS)

Localización de puntos de interés: Obtienen los puntos de interés definitivos a partir de los candidatos de la etapa anterior, lo que resulta en una escala σ para cada punto de interés, que utiliza un filtro Gaussiano sobre la imagen de partida para después crear la imagen de trabajo L. Otra forma de localizar estos puntos es mediante el ajuste de una función cuadrática 3D a los puntos de una muestra local de forma que se pueda determinar la localización interpolada del máximo, mejorando la búsqueda de correspondencias y la estabilidad, eliminando puntos que tienen bajo contraste, sean sensibles al ruido o que no estén bien localizados al encontrarse a lo largo de un borde.

$$L = G(x, y, \sigma) * I(x, y); G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Parámetros: x, y (coordenadas del píxel), $*$ (operación de convolución en dichas coordenadas) y k (factor multiplicativo constante para detectar valores extremos del espacio de forma eficiente y estable).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

5.26 SIFT (FUNCIÓN EN EL EXTREMO Y HESSIANO DE 2X2)

Función en el extremo: Útil para rechazar extremos inestables con bajo contraste en la siguiente función, aunque posteriormente Lowe propuso descartar todos los extremos de $|D(\hat{x})|$ con un valor menor de 0,03 asumiendo que todos los niveles de gris de la imagen están en el rango $[0,1]$.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D}{\partial x} \hat{x}$$

Hessiano de 2x2: Permite obtener la curvatura de un punto a una escala determinada a partir de las derivadas parciales segundas con respecto a x e y estimadas a partir de las diferencias de los vecinos del punto de muestreo.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

5.27 SIFT (ASIGNACIÓN DE ORIENTACIONES)

Asignación de orientaciones: Consigue invarianza ante la rotación de imágenes, asignando a cada descriptor una orientación en función de las propiedades locales de la imagen, por lo que se calcula la magnitud m y la fase θ del gradiente para cada punto de interés.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$
$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right), \text{ donde } \sigma_2 = 1,5\sigma \text{ (ventana circular gaussiana)}$$

Descriptor de puntos de interés: Obtiene las características SIFT para cada uno de los puntos de interés localizados, para los que se genera un vecindario con una región cuadrada que se subdivide en 16 subregiones del mismo tamaño a lo largo del punto característico, y que a su vez contiene 8 direcciones para los gradientes obtenidos en la etapa anterior con un bin de 45° y una ventana gaussiana $\sigma_3 = 0,5\sigma$, los cuales se concatenan, obteniendo así descriptores de 128 elementos y el vector de características SIFT.

Octava: Espacio que reduce sucesivamente el tamaño de la imagen original y que se obtiene eliminando una de cada dos filas y columnas sobre la imagen de la octava anterior, reduciendo sus dimensiones a la mitad, cuya cantidad está condicionada por el tamaño y el detalle de la imagen original (normalmente se aplican 4 octavas y 5 escalas).

5.28 SURF

SURF (Speeded Up Robust Features): Detecta puntos de interés en una imagen y obtiene descriptores que ofrecen una mayor repetibilidad, diferenciación y robustez, además de usar imágenes integrales para operar con imágenes para que el método sea considerablemente más rápido. Indica cómo se distribuye la intensidad de los píxeles dentro de un vecindario dependiente de la posición y escala de cada punto de interés fijado.

Complejidad: Asigna una orientación sobre el punto en que trabaja en base a una región circular alrededor del propio punto y se construye una región cuadrada alineada con dicha orientación para extraer el descriptor.

Proceso de extracción: Construir una región cuadrada alrededor del punto de interés de tamaño 20σ , subdividir la región cuadrada en 16 subregiones del mismo tamaño manteniendo la información espacial, y calcular los wavelets de Haar de tamaño 2 para 25 puntos distribuidos de forma uniforme, obteniendo así un vector de 4 valores para cada subregión, donde dx y dy son las respuestas wavelet de x e y , para concatenarlos y formar así un descriptor SURF formado por 64 características.

$$V_{\text{subregión}} = [\Sigma dx, \Sigma dy, \Sigma |dx|, \Sigma |dy|]$$

TEMA 6: OPERACIONES MORFOLÓGICAS

6.1 PRINCIPIOS BÁSICOS

Procesamiento morfológico: Técnica de la visión por computador que analiza las imágenes basándose en propiedades de la forma, empleando álgebra de conjuntos. Estos métodos pueden ser aplicados tanto a imágenes binarias, como a imágenes en niveles de gris y de color.

Utilidad: Preprocesamiento de la imagen, operaciones para la supresión de ruido o el realce de objetos y análisis de imágenes (extraer características tales como el esqueleto o definición del entorno).

6.2 OPERACIONES BÁSICAS CON CONJUNTOS

Inclusión: Un conjunto X está incluido en otro Y , si todo elemento de X también pertenece a Y .

$$X \subset Y: \{\forall x \in X \text{ entonces } x \in Y\}$$

Características: A cada transformación $\psi(X)$, se le puede asociar una transformación dual $\psi^*(X)$, de modo que, si dos operaciones son duales, por ejemplo, la intersección y la unión, se llegue al mismo resultado sustituyendo una por la otra y los conjuntos por sus complementarios. Esto también es aplicable al álgebra de Boole con las operaciones AND y OR reemplazando los 0 por 1.

$$\text{Complemento: } X^C = \{x \mid x \notin X\} \quad \text{Dualidad: } \psi^*(X) = (\psi(X^C))^C$$

Unión de conjuntos: Un conjunto es unión de los conjuntos X e Y si está constituido por todos los elementos de los conjuntos X e Y , es decir, sus elementos pertenecen a X , a Y , o a ambos. La unión se programa fácilmente entre imágenes binarias con la operación booleana OR.

$$X \cup Y = \{x \mid x \in X \vee x \in Y\}$$

Intersección de conjuntos: Conjunto formado por los elementos comunes de X e Y . La intersección es fácil de realizar entre imágenes binarias con la operación booleana AND.

$$X \cap Y = \{x \mid x \in X \wedge x \in Y\} \text{ de forma dual } X \cap Y = (X^C \cup Y^C)^C$$

Conmutatividad: $X \cap Y = Y \cap X$; $X \cup Y = Y \cup X$

Asociatividad: $(X \cap Y) \cap Z = X \cap (Y \cap Z)$; $(X \cup Y) \cup Z = X \cup (Y \cup Z)$

Idempotencia: Esta propiedad, que se cumple con algunas operaciones, implica que no se logra nada al repetir la operación sobre un conjunto si también se repite el mismo operador.

$$(X \cap Y) \cap Y = (X \cap Y); (X \cup Y) \cup Y = (X \cup Y)$$

Diferencia entre conjuntos: Conjunto donde sus elementos son pertenecientes a un conjunto, pero no al otro.

$$X - Y = \{x \mid x \in X \wedge x \notin Y\} = X \cap Y^C$$

Traslación por un vector: Un conjunto X es trasladado por un vector v de coordenadas (v_1, v_2) cuando cada uno de los elementos de x de coordenadas (a, b) sufre esa traslación.

$$X_v = \{y \mid y = x + v \forall x \in X\}$$

Transformaciones morfológicas básicas: Consisten en el empleo de las operaciones de conjuntos ante vistas. Se emplea, por una parte, un conjunto X constituido por un elemento de la imagen identificado como objeto, y otro conjunto B , habitualmente de pequeño tamaño, llamado elemento estructural. Según la forma y tamaño del elemento estructural, así como del tipo de funciones que se combinen, podrán realizarse multitud de transformaciones de la imagen con diversos ámbitos de aplicación.

6.3 DILATACIÓN

Dilatación: Operación que da como resultado un conjunto de puntos, formado por todas las posibles sumas de pares de puntos, en las que uno pertenece al conjunto X y otro al elemento estructural desplazado.

$$X \oplus B = \{c = x + b, \forall x \in X \wedge \forall b \in B\}$$

Características: Al dilatar la imagen aumenta su tamaño, pero pueden perderse detalles de la forma, se rellenan intersticios de la imagen y también, como efecto muchas veces pernicioso, se pueden fusionar objetos diferentes que se encuentren próximos entre sí, cuando el elemento estructural es grande el efecto es más acusado.

Conmutatividad

$$\text{Invarianza a la traslación: } X_v \oplus B = (X \oplus B)_v$$

Asociatividad

$$\text{Creciente: Si } X \subset Y \rightarrow X \oplus B \subset Y \oplus B$$

Distributiva respecto a la unión: $B \oplus X \cup Y \rightarrow B \oplus X \cup B \oplus Y$

} 6.4 EROSIÓN

Erosión: Operación con el efecto contrario a la dilatación, por lo que reduce el tamaño de los objetos. La mecánica es semejante, desplazando el elemento estructural a lo largo de la imagen, pero ahora el resultado será 1 para el punto en estudio si todos los puntos del EE están contenidos dentro del conjunto del objeto. La erosión reducirá el tamaño de los objetos dependiendo de la dimensión y forma del elemento estructural. Aunque el efecto es contrario al de la dilatación esto no debe llevar al error de considerar que erosionando una imagen dilatada seremos capaces de recuperar la imagen original, esto nunca es así.

$$X \ominus B = \{x \mid B_x \subseteq X\} = \{x \mid B_x \cap X^c \neq \emptyset\}$$

Propiedades: No existe conmutatividad, invariante a la traslación, creciente, y además se cumple:

$$(X \cup Y) \ominus B \supseteq (X \ominus B) \cup (Y \ominus B); B \ominus (X \cup Y) = (B \ominus X) \cap (B \ominus Y); \text{Si } D \subseteq B \text{ entonces } X \ominus B \subseteq X \ominus D$$

} 6.5 APERTURA Y CIERRE

Apertura: Esta operación se realiza mediante una erosión seguida de una dilatación empleando el mismo elemento estructural. Al erosionar se pierden detalles pequeños, desaparece el ruido y mengua la imagen. Haciendo seguidamente la dilatación la imagen crece, y los detalles que no llegaron a perderse se resaltan.

$$X \circ B = (X \ominus B) \oplus B$$

Cierre: Esta operación se realiza mediante una dilatación seguida de una erosión empleando el mismo elemento estructural. Al dilatar se rellenan intersticios, la imagen crece. Cuando se erosiona la imagen vuelve a un tamaño semejante al original.

$$X \cdot B = (X \oplus B) \ominus B$$

} 6.6 CONTORNOS Y ESQUELETO

Contorno exterior: Toma la imagen dilatada y se le resta la original ($\text{Contorno}_{\text{exterior}} = (X \oplus B) - X$).

Contorno interior: Resta de la imagen original la imagen erosionada ($\text{Contorno}_{\text{interior}} = X - (X \ominus B)$).

Esqueleto: Procedimiento para obtener un descriptor de una región mediante un grafo. La idea es que dos objetos diferentes tendrán distinto grafo, que ocupan mucho menos espacio de almacenamiento que la región y que son también más fáciles de procesar y comparar. El esqueleto de una región sería el grafo de los puntos interiores cuya distancia mínima a algún punto del borde se repite dos o más veces.

Características: A base de sucesivas iteraciones se iría adelgazando la imagen, hasta obtener el esqueleto cuando no haya diferencia entre dos iteraciones. En cada iteración deben respetarse 2 normas (no suprimir puntos extremos ni debe romper la conectividad). Este método es también computacionalmente costoso, en cada iteración se emplean varias máscaras y debe iterarse hasta reducir la figura a un grafo.

Método Zhang-Suen: Se definen 8 vecinos de un píxel y, de forma iterativa, se aplican dos conjuntos de condiciones. Aquellos píxeles que cumplan estas condiciones en una etapa todas las condiciones de un conjunto se eliminan, y no debe romperse la conectividad.

Conjunto 1: Para preservar el mantenimiento de los puntos finales se examina si el número de vecinos distintos de 0 está entre 2 y 6 ($2 \leq N(P) \leq 6$), que solamente se pasa de 0 a 1 una vez si se recorre el borde del entorno de P1 a P8 garantiza que el píxel no elimine partes conectadas importantes, que alguno de los vecinos P1, P3 y P5 es cero evita eliminar puntos críticos de la estructura, y que alguno de los vecinos P3, P5 o P7 es cero refuerza la condición de preservación de la estructura.

Conjunto 2: Igual a la del conjunto 1, que alguno de los P1, P3 y P7 es cero, y que alguno de los P1, P5 o P7 es cero.

} 6.7 RELLENADO DE HUECOS

Proceso: Se irá construyendo un conjunto de puntos X_k , partiendo de una imagen en la que solo contiene un punto del interior, hasta que se encuentre que X_k y X_{k-1} son iguales. En cada iteración, el conjunto X_k es el resultado de la dilatación de X_{k-1} con el Elemento Estructural y la posterior intersección con el complementario del conjunto de la imagen original.

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

6.8 OPERACIONES EN IMÁGENES GRISES

Valor de gris: Representativo del valor de la función en cada punto ($E \rightarrow T$), donde E es el espacio de puntos y T el de niveles de grises, que será siempre un subconjunto de los números reales.

Dilatación: $X \oplus B = \max\{f(x - i, y - j) + b(i, j) \mid (x - i, y - j) \in Df; (i, j) \in Db\}$

Erosión: $X \ominus B = \min\{f(x - i, y - j) - b(i, j) \mid (x - i, y - j) \in Df; (i, j) \in Db\}$

Siendo Df el dominio de la imagen y Db el dominio del EE; (x, y) definen las coordenadas en el dominio de la imagen e (i, j) lo hacen en el del EE. Como los valores (i, j) de las coordenadas del EE aparecen restando a las (x, y) de la imagen, a la hora de operar puede emplearse el reflejo del EE. Para cada evaluación de un entorno el resultado será el valor máximo o mínimo del resultado de sumar en una vecindad los valores de los puntos de la imagen y los del EE reflejado. Es frecuente utilizar EE planos, con todos sus valores a cero.

6.9 GRADIENTE MORFOLÓGICO

Gradiente morfológico: Consiste en resaltar los elementos de alta frecuencia de la imagen, y de forma particular los contornos de la misma. Este efecto se puede conseguir sustrayendo de la dilatación el resultado de la erosión.

$$X \oplus B - X \ominus B$$

TEMA 7: GEOMETRÍA Y CALIBRACIÓN

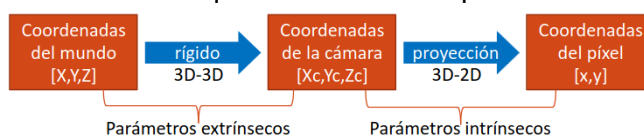
7.1 CALIBRACIÓN

Calibración: Corrección de lo que percibe el sensor para que coincida con la realidad, ya que todos los sensores tienen ruido y captan una versión distorsionada de la realidad (parámetros intrínsecos). Esto involucra saber la transformación que hay entre sus dos sistemas de coordenadas, la cual viene definida por los parámetros extrínsecos (aquellos que dependen de la posición del sistema de visión y de cada escenario).

Parámetros intrínsecos: Definen la geometría interna y la óptica de la cámara (proyección 3D en el plano 2D). Son fijos siempre y cuando no se modifiquen las características de la cámara, aunque éstos se podrían cambiar modificando el grupo óptico o la apertura de la cámara.

Parámetros extrínsecos: Relacionan los sistemas de referencia del mundo real y la cámara, describiendo la posición y orientación de la cámara en el sistema de coordenadas del mundo real. A diferencia de los parámetros intrínsecos, éstos no son fijos, ya que dependen de la cámara con respecto del mundo, aunque son capaces de alinear el sistema de coordenadas de los puntos de la escena 3D con respecto al sistema de coordenadas de la cámara.

Usos: Correspondencia de sensores, eliminación de la distorsión y profundidad del campo definida. Sin embargo, puede aparecer la distorsión radial que no está contemplada en la matriz K.

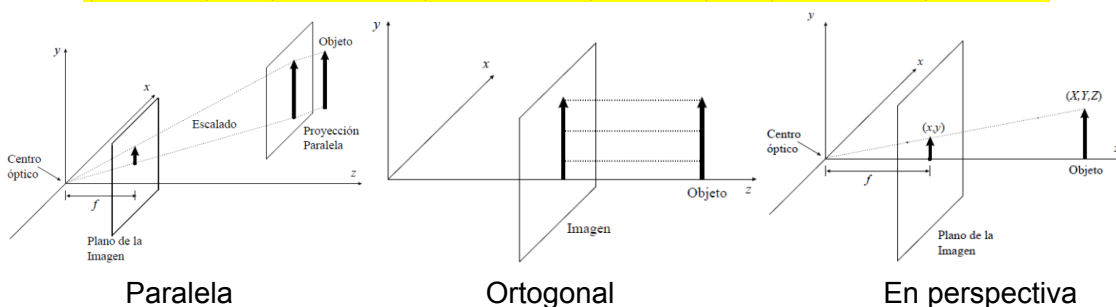


7.2 PROYECCIONES

Proyección ortogonal: Define un área de visión de tipo paralelepípedo (cuadrado en todas sus caras).

Proyección en perspectiva: Define un área de visión en forma de sección piramidal de forma que los objetos más alejados del observador se reducen en tamaño.

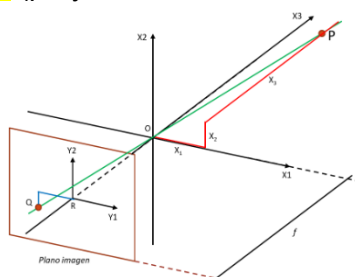
$$\frac{x}{f} = -\frac{X}{Z-f} = \frac{X}{f-Z} \Rightarrow x = \frac{fX}{f-Z} = \frac{fX}{Z} ; \quad \frac{y}{f} = -\frac{Y}{Z-f} = \frac{Y}{f-Z} \Rightarrow y = \frac{fY}{f-Z} = \frac{fY}{Z}$$



7.3 MODELO PIN-HOLE

Modelo Pin-Hole: Describe una cámara sin lente y con una pequeña apertura, donde los rayos de luz atraviesan dicha apertura proyectando una versión invertida de lo que se capta en el fondo de la cámara o en el plano de la imagen, cuya distancia entre la apertura y el plano de la imagen se denomina distancia focal.

Parámetros: **P** (punto en el espacio 3D que se va a proyectar en la imagen), **X1X2X3** (define el sistema de coordenadas de la cámara con origen en la apertura de la cámara **O**, donde **X3** coincide con la dirección de la cámara), **plano de la imagen** (paralelo a **X1X2** desplazado una distancia **f** en la dirección negativa de **X3**), **R** (punto que intersecta **-X3** y el plano de la imagen, además de ser el centro óptico de la imagen), **Q** (intersección de la recta que parte de **P** y pasa por **O** en el plano de la imagen, es decir, la proyección del punto 3D en el plano de la imagen), **Y1Y2** (proyección 2D de todos los puntos 3D).



7.4 PARÁMETROS INTRÍNSECOS

Distancia focal (F): $f_x = F * k$, $f_y = F * l$, donde k y l son el ancho y el alto de un píxel, respectivamente.

Centro de la imagen: c_x (coordenada x del centro de la imagen), c_y (coordenada y del centro de la imagen).

Skew (s): Ángulo que toma el eje Y del sistema de coordenadas del plano de la imagen cuando los ejes no son perpendiculares ($s = -f_x * \cot(\alpha)$).

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Relación puntos 3D-2D: $x = f_x \frac{X}{Z} - f_x \cot(\alpha) \frac{Y}{Z} + c_x$; $y = \frac{f_y}{\sin(\alpha)} \frac{Y}{Z} + c_y$; $[x \ y \ 1]^T = K[X \ Y \ Z]^T$.

Relación puntos 2D-3D: $X = \frac{(x - c_x) * d}{f_x}$; $Y = \frac{(y - c_y) * d}{f_y}$; $Z = d$, donde d es el valor de profundidad.

Conversión 3D-2D y 2D-3D: $(x, y) = (f_x \frac{X}{Z}, f_y \frac{Y}{Z})$; $(X, Y, Z) = (\frac{(x - c_x) * d}{f_x}, \frac{(y - c_y) * d}{f_y}, d)$.

7.5 PARÁMETROS EXTRÍNSECOS

Matrices de Rotación (R): Rotación de los 3 ejes entre los sistemas de coordenadas, y está compuesta por la acumulación de la rotación independiente de cada eje para hacer cuadrar ambos sistemas de coordenadas.

$$R_x(\theta) = \begin{matrix} \text{roll} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \end{matrix} \quad R_y(\theta) = \begin{matrix} \text{pitch} \\ \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \end{matrix} \quad R_z(\theta) = \begin{matrix} \text{yaw} \\ \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$[R_0] = \begin{bmatrix} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi) \\ \sin(\psi)\cos(\theta) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}$$

Matrices de Traslación (t): Desplazamiento en x,y,z entre los orígenes de los sistemas de coordenadas, la cual está compuesta por un vector que indica cuánto habría que mover el origen de un sistema de coordenadas para hacerlo coincidir con el otro ($t = [t_x \ t_y \ t_z]^T$).

7.6 MATRIZ DE PROYECCIÓN

Matriz de proyección de un punto 3D en el mundo a un punto 2D en el plano de la imagen: $x = K(RX + t)$.

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Parámetros intrínsecos Parámetros extrínsecos

Parámetros: X, Y, Z (coordenadas de un punto 3D en el mundo real), u, v (coordenadas del punto proyectado en píxeles), f_x, f_y (longitudes focales en píxeles), c_x, c_y (puntos pertenecientes al centro de la imagen), r, t (matrices de rotación y traslación entre la cámara y la zona proyectada), s (ángulo que tiene un píxel respecto al eje y, normalmente 0) y w (factor de escala que sirve para homogeneizar u y v).

7.7 CALIBRACIÓN

Paso 1 (captura de imágenes mediante el elemento de calibración): Se necesita tener puntos sobre el plano, el cual debe sufrir transformaciones de perspectiva para dotar al sistema de diferentes valores de profundidad, donde se utiliza un patrón estilo tablero de ajedrez ya que se pueden obtener las intersecciones de las casillas mediante un algoritmo de detección de esquinas.

Paso 2 (cálculo de puntos 2D para minimizar el error de reproyección): Calcula la transformación del patrón y establece el origen de coordenadas en los puntos de vista de la cámara, donde el origen del mundo se sitúa en el patrón y se calcula la transformación para la pose de la cámara en cada imagen (ambas aproximaciones son equivalentes y obtienen la misma información, salvo que una es la inversa de la otra).

Paso 3 (minimización del error de reproyección): Proyecta los puntos 3D en 2D usando una K , r y t iniciales, minimiza el error entre la proyección anterior y los puntos 2D observados, lo que modifica K , r y t para hacer coincidir la proyección con la posición observada. Y por último, todo esto termina devolviendo K , r y t ya optimizadas.

TEMAS 8: VISIÓN 3D

8.1 INTRODUCCIÓN

Sensores 3D: Permiten obtener la información de las superficies que componen una escena escaneando puntos sobre éstas (pueden incluir información de color o no), cuyo resultado es una nube de puntos compuestos por sus coordenadas XYZ y, opcionalmente, color RGB.

Características: Alcance, resolución, precisión y tecnologías (estéreo, RGB-D, infrarrojos y láser ToF).

8.2 LÁSER

Láser: Emiten una señal y esperan hasta que choque (refleje) con un objeto y vuelva al origen, el Tiempo de Vuelo (ToF) es el tiempo que tarda la señal en reflejarse, cuanto mayor longitud de onda más precisión y alcance, el láser es dirigido con espejos en 1 o 2 dimensiones, tiene la posibilidad de montar un láser 2D en un sistema pan-tilt y el sistema genera una nube de puntos.

Nube de puntos: Representación 3D de puntos en el espacio, donde cada punto corresponde a una medición de distancia obtenida, normalmente en el sistema de referencia del propio sensor.

Características: Largo alcance (500 m, Luminar Hydra), precisión (< 1 mm independiente a distancia), resolución alta, reflectividad o color RGB, precio muy alto (5000 - 10000 €), peso y consumo muy elevado.

8.3 RGB-D

Imagen de profundidad: Representa la distancia desde el centro óptico del sensor (la cámara) hasta un punto específico de la escena en el espacio tridimensional.

Cámara RGB-D activa: Contienen sensores compuestos por un emisor y un receptor de luz infrarroja (IR) y una cámara 2D, el emisor IR proyecta un patrón conocido sobre la escena, el receptor capta el patrón proyectado y analiza las divergencias con el patrón original para obtener la profundidad de los puntos, y la cámara 2D se utiliza para asignar color RGB a los puntos 3D.

Características (infrarrojo): Bajo coste, detección en zonas sin textura, permiten trabajar en la oscuridad, alcance limitado (40 - 10 cm), baja resolución (640 x 480), bajo framerate (30 FPS), error de medida (2 - 5 %), falla en zonas que reflejan mal la luz IR, no funciona en exterior y suele fallar en los bordes.

Cámara RGB-D pasiva: Componen lo que se denomina un sistema estereoscópico (sirve para obtener la imagen de disparidad) y estos sistemas proporcionan dos imágenes (en color o escala de grises) cada una correspondiente a un sensor de imagen.

Características (estereoscopia): Coste variable (hardware adicional), cálculo de la disparidad por software, no funciona en escenas no iluminadas, problemas con zonas sin texturas, nubes de puntos 3D (escala de gris o RGB), error de medida (2 - 4 %), alcance limitado / baseline (5 - 15 m), alta resolución (1280 x 1024), alto framerate (120 FPS) y funciona en exterior.

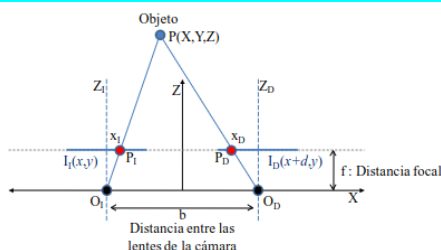
Cálculo de la profundidad: Se proyecta un patrón de puntos con un láser infrarrojo. A partir de la luz infrarroja que llega rebotada a la cámara de infrarrojos se obtiene la profundidad a la que se encuentra cada píxel. El patrón de puntos está memorizado para una profundidad conocida. Al poner objetos delante, el patrón aparece distorsionado y así se puede calcular la profundidad. Como hay menos puntos en el patrón de IR que el mapa de profundidad, algunas partes de este mapa son interpoladas.

8.4 VISIÓN ESTEREOSCÓPICA

Visión estereoscópica: Permiten inferir información geométrica 3D sobre la escena, basándose en aprovechar la información procedente de dos imágenes distintas para suplir la pérdida de información que supone el proceso de proyección de la escena en la imagen 2D, donde los objetos aparecerán en distinta posición en una imagen (izquierda) y en la otra (derecha), ya que se utilizan dos cámaras. Para poder realizar esta reconstrucción se necesita identificar las proyecciones de un mismo punto en la escena de cada imagen para después utilizar la información geométrica sobre la disposición relativa de las imágenes y así poder obtener la posición de dichos puntos en el espacio.

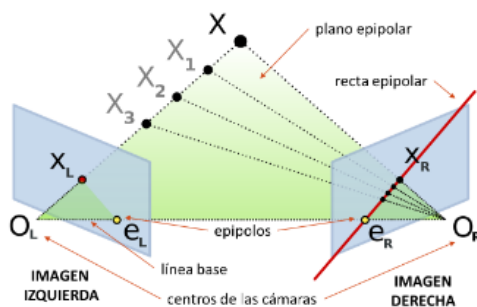
Restricción epipolar: Liga la geometría de ambas imágenes y resulta clave para facilitar la búsqueda de correspondencias o para inferir la posición y orientación de una cámara respecto a la otra cuando el par estéreo no ha podido ser calibrado, estableciendo que cuanto más cerca estén los objetos de la cámara, mayor será su cambio relativo, cuyo desplazamiento relativo de los objetos en las dos imágenes es lo que se conoce como disparidad.

8.5 GEOMETRÍA DE LOS SISTEMAS ESTÉREO



$$\left. \begin{aligned} O_I : \frac{b+X}{Z} &= \frac{x_I}{f} \\ O_D : \frac{b-X}{Z} &= \frac{x_D}{f} \end{aligned} \right\} \Rightarrow \left. \begin{aligned} x_I &= \frac{f}{Z} \left(X + \frac{b}{2} \right) \\ x_D &= \frac{f}{Z} \left(X - \frac{b}{2} \right) \end{aligned} \right\} \Rightarrow d = x_I - x_D = \frac{fb}{Z} \Rightarrow Z = \frac{fb}{d}$$

Parámetros: Z_I, Z_D (ejes ópticos mutuamente paralelos y separados por una distancia horizontal b denominada línea base), I_I, I_D (planos imagen con centros de proyección O_I, O_D respectivamente) y f (longitud focal efectiva de cada cámara, donde la disparidad d se calcula para cada par de puntos emparejados $P_I(x_I, y_I)$ y $P_D(x_D, y_D)$ como $d = (x_I - x_D)$).



Línea base: Segmento 3D que une los centros de proyección de ambas cámaras.

Plano epipolar: Definido por un punto de la escena y los centros de proyección de ambas cámaras, capaz de provocar que los planos epipolares formen un haz de planos cuyo eje sea precisamente la línea base.

Epipolo: Punto en el que la línea base corta al plano imagen asociado a dicha vista (proyección de una cámara y/o el centro de proyección de la otra cámara que forma el par estéreo), que puede caer fuera del trozo del plano imagen que devuelve la cámara cuando la línea base es paralela al plano imagen.

Recta epipolar: Recta que intersecta el plano epipolar con el plano imagen.

8.6 CORRELACIÓN Y CORRESPONDENCIA DE PUNTOS

Correlación: Usa una zona alrededor de un punto para compararla con puntos de la otra imagen (estéreo denso), y consiste en sumar el resultado de multiplicar píxel a píxel una cierta vecindad en las dos imágenes, siendo capaz de identificar características para calcular su profundidad (estéreo disperso).

Algoritmo: Define una cierta ventana de tamaño W , calcula el desplazamiento (d_x, d_y) producido en la imagen derecha por cada punto (x, y) , encontrando así el desplazamiento que maximiza la función anterior. Este método se puede realizar utilizando la línea epipolar en vez de buscar por toda la imagen.

$$c(x, y, d_x, d_y) = \sum_{k=-W}^W \sum_{l=-W}^W I_I(x+k, y+l) * I_D(x+k+d_x, y+l+d_y); (d_x, d_y) = \max\{c(x, y, d_x, d_y)\}$$

8.7 MÉTODOS DE REGISTRO 3D

Matching basado en características: Extrae características distinguibles que permitan ser identificadas en ambos scans y calcula la transformación rígida que las alinea a partir del emparejamiento correspondiente entre características de ambos scans (RANSAC).

Matching como un problema de optimización: Usa una función de coste que mide la calidad del alineamiento entre puntos de ambos scans, donde normalmente suele determinarse la transformación rígida que minimiza dicha función de coste en un proceso iterativo.

Cálculo de la transformación rígida (solución cerrada): Aplicable cuando se conoce la correspondencia entre los puntos, y puede resolverse de 4 formas diferentes (Descomposición en Valores Singulares de una Matriz, matrices ortonormales, cuaterniones simples y cuaterniones duales).

Cálculo de la transformación rígida (solución iterativa): Aplicable cuando las correspondencias entre los puntos son desconocidas mediante el algoritmo ICP (Iterative Closest Points).

8.8 ICP

ICP (Iterative Closest Points): Método iterativo que permite alinear dos superficies asumiendo un solapamiento parcial/total y un conocimiento inicial de un alineamiento grueso, permitiendo encontrar la transformación 3D que mejor alinea los datos de entrada. Las superficies a alinear vienen caracterizadas por conjuntos de puntos 3D utilizando una medida de distancia entre los elementos de un conjunto y los del otro, es decir, la distancia euclídea.

Entradas del algoritmo: Modelo/Conjunto de puntos de una/ambas superficies, la transformación inicial (odometría, acción, etc) y el criterio de parada.

Primer algoritmo: Aplicar la transformación lineal al conjunto escena, emparejar cada punto de la escena transformada con el más cercano del conjunto modelo, y estimar la transformación (rotación/traslación) a partir de los emparejamientos de manera que minimice la función de coste utilizando uno de los métodos en forma cerrada.

Segundo algoritmo: Sean dos conjuntos de puntos 3D, **model set** ($|M| = N_m$) y **data set** ($|D| = N_d$), se trata de encontrar la **transformación rígida** (R, t) que minimiza la siguiente **función de coste**:

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{ij} ||m_i - (Rd_j + t)||^2$$

Parámetros: w_{ij} (1 si m_i empareja con d_j , y 0 en otro caso). Generalmente se calcula primero la rotación R y después la traslación t , cuya solución es en la que las 2 nubes de puntos tienen el mismo centroide. Para ello, se modifican las 2 nubes de puntos restándoles sus respectivos centroides de forma que el problema se reduzca al minimizar la función de error:

$$D' = \{d'_i = d_i - d_m\}_{1, \dots, N} ; M' = \{m'_i = m_i - c_m\}_{1, \dots, N} ; E(R, t) \propto \sum_{i=1}^N ||m'_i - Rd'_j||^2$$

Descomposición en Valores Singulares de una Matriz (SVD): Descompone una matriz como producto de matrices ortogonales (direcciones y rotaciones) y diagonales (escalado), donde la matriz H puede expresarse como:

$$H = U_{n \times n} S_{n \times m} V_{m \times m}^T$$

Parámetros: $U_{n \times r}$ y $V_{m \times r}^T$ (matrices ortogonales, donde $U^T U = I_{n \times n}$ y $V^T V = I_{m \times m}$), S (diagonal con r elementos positivos en la diagonal siendo r el rango de la matriz y el resto cero, cuyos elementos se denominan valores singulares de H) y U y V (direcciones principales de salida y entrada, respectivamente).

Análisis: Método iterativo (costoso y posee un cuello de botella en la búsqueda de emparejamientos, por lo que se utilizan árboles de búsqueda KD), minimiza el error de alineamiento, es muy sensible a outliers (puntos que están solamente en uno de los dos conjuntos). Por otro lado, puede extenderse a conjuntos de diferente tipo (punto-punto, punto-plano, plano-plano, punto-línea, punto-superficie, etc), por lo que se necesita definir la distancia entre ellos.

ICP en robótica: Presenta el inconveniente de la gran cantidad de outliers que suelen presentar los datos, existen métodos para minimizar el impacto de los outliers, se tienen graves problemas con el cambio de escalas (frecuencia de muestreo de objetos a diferentes distancias), si no se dispone de una transformación inicial aplicada al aplicar ICP sin disponer de odometría, conviene utilizar información extra de las superficies que encontremos (superficies planas), y por último, se puede utilizar ICP con parches planares en vez de con puntos.

8.9 VARIANTES DE ICP

Selección de puntos en uno o ambos conjuntos: Coge todos los puntos disponibles y los muestrea uniformemente o aleatoriamente con diferentes muestras en cada iteración, utilizando un alto gradiente de color/intensidad. También se puede realizar un muestreo con la mayor distribución de normales uniformemente en cada subespacio angular de normales, facilitando la convergencia en escenas con características pequeñas y dispersas.

Emparejamiento de puntos entre los dos conjuntos: Consiste en encontrar el punto más cercano en el otro conjunto (Closest Point), utilizando un Kd-tree para acelerar la búsqueda. Además, trata de encontrar la intersección del rayo en la dirección de la normal con el conjunto modelo (Normal Shooting), proyectando así el punto sobre el conjunto modelo desde el punto de vista de la cámara del conjunto modelo (Reverse Calibration).

Ponderación de correspondencias entre puntos: Poseen pesos constantes y asignan pesos en función de la distancia entre los puntos. Además, pueden ponderar en función de la compatibilidad de normales, en función del color, o basándose en la incertidumbre de la medición/ruido del sensor.

$$Weight = 1 - \frac{Dist(p_1, p_2)}{Dist_{max}} ; Weight = n_1 * n_2$$

Eliminación de correspondencias incorrectas (outliers): Consiste en eliminar correspondencias con distancia entre sus puntos superior a un determinado umbral, un determinado porcentaje (10%) de los peores emparejamientos siguiendo ciertas métricas (distancia entre puntos), aquellos emparejamientos cuya distancia entre sus puntos sea superior a un determinado múltiplo (2.5) de la desviación típica de las distancias, emparejamientos que no sean consistentes con otros emparejamientos vecinos, y por último, correspondencias que contienen puntos en los límites del conjunto destino, siendo aplicable cuando el solapamiento entre los conjuntos no sea completo.

Si $|Dist(p_1, p_2) - Dist(q_1, q_2)| < Umbral$, los dos emparejamientos (p_1, p_2) y (q_1, q_2) son inconsistentes

Medición del error de emparejamiento: Compuesto por métricas punto-a-punto que consideran la suma de distancias al cuadrado entre puntos correspondientes y la diferencia de color. Además, es la distancia punto-al-plano que considera la suma de distancias al cuadrado entre cada punto con el plano más cercano perpendicular a la normal del punto destino.

Minimización del error: Genera repetidamente emparejamientos usando la transformación actual y encontrando una nueva transformación que minimice la métrica del error, donde la minimización iterativa puede realizarse combinada con la extrapolación en el espacio de transformaciones para acelerar la convergencia, con distintas perturbaciones en las condiciones iniciales para seleccionar el mejor resultado, y por último, usando distintos subconjuntos de puntos seleccionados aleatoriamente eligiendo un resultado óptimo usando una métrica robusta (least median of squares).

8.10 CÁMARA RGB-D KINECT

Orígenes: Comercializada desde Noviembre de 2010 y desarrollada por Primesense y utilizada por Microsoft para la consola Xbox 360. Es capaz de reconocer gestos, comandos de voz, objetos e imágenes, y su uso se ha extendido a la comunidad científica.

APIs: OpenNI (Primesense), SDK (Microsoft) y OpenKinect (comunidad abierta).

Sensores: Cámara RGB (32 bits de color, 640x480 píxeles, 57°x47° de campo visual y 30 Hz), sensor de profundidad (proyector y cámara de infrarrojos de 16 bits de profundidad, 320x240, 30 Hz y 1,2-3,5 metros), motor de inclinación (tilt hacia arriba o hacia abajo de hasta 27°) y micrófonos multiarray (4 canales a 16 bits de audio y 16 KHz).

Cálculo de la profundidad: Proyecta un patrón de puntos con un láser infrarrojo, a partir de la luz infrarroja que llega rebotada a la cámara de infrarrojos se obtiene la profundidad a la que se encuentra cada píxel. El patrón de puntos está memorizado para una profundidad conocida, por lo que al poner objetos delante, el patrón aparece distorsionado pudiendo así calcular la profundidad. Como hay menos puntos en el patrón de IR que en el mapa de profundidad, algunas partes de este mapa son interpoladas.

Ventajas: Bajo coste (100 €), proporciona 30 fps con resolución 640x480, capaz de detectar profundidad en zonas sin textura y puede funcionar a oscuras.

Inconvenientes: Alcance limitado (50cm - 4m), falla en zonas que reflejan la luz infrarroja, no funciona en entornos de exterior y suele fallar en los bordes de los objetos.

8.11 PCL (POINT CLOUD LIBRARY)

PCL (Point Cloud Library): Librería de procesamiento de nubes de puntos 3D con un software de libre distribución y licencia BSD desarrollado por la empresa Willow Garage, la cual incluye diversos algoritmos (filtrado, extracción de características, reconstrucción de superficies, alineamiento, ajustes de modelo, segmentación y visualización), implementada en C++ y utilizada en plataformas de desarrollo Linux, MacOS, Windows y Android, además de poder soportar de forma nativa las interfaces 3D de OpenNI.

PointT: Añade tipos definidos por el usuario, además de los tipos de puntos predefinidos ya existentes (PointXYZ, PointXYZRGB, PointNormal, etc).

PointCloud (int width, int height, std::vector<PointT> points): Permite almacenar una nube de puntos, organizada (height = número de filas) o desorganizada (height = 1).

libpcl_filters (filtros de datos): Reducción de la resolución (downsampling), eliminación de outliers, etc.

libpcl_features (extracción de características 3D): Normales de superficies, puntos frontera, descriptores SIFT, NARF, etc.

libpcl_io (entrada y salida): Escribir y leer ficheros en formato PCD (Point Cloud Data).

libpcl_segmentation (segmentación): Extracción de clusters, ajustes de modelo mediante métodos de consenso, etc.

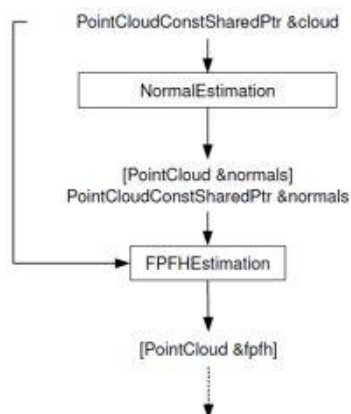
libpcl_surface: Reconstrucción de superficies.

libpcl_registration: ICP.

libpcl_keypoints: Extracción de puntos clave que puede utilizarse para decidir dónde extraer descriptores de características.

libpcl_range_image: Dar soporte a mapas de profundidad creados a partir de nubes de puntos.

Procesamiento pipeline en PCL: Crear el objeto de procesamiento (filtrado, estimador de características, segmentación, etc), utilizar setInputCloud para pasar la nube de puntos de entrada al módulo de procesamiento, actualizar algunos parámetros, invocar al correspondiente módulo de procesamiento (compute, filter, segment, etc) para obtener el resultado.



Proyecto KinectFusion: Fusiona el mundo real y el virtual, generando modelos 3D en tiempo real (objetos, personas y habitaciones enteras), todo ello mediante un motor de físicas realista que permite escanear objetos para ser manipulados con realismo.

TEMA 9: FLUJO ÓPTICO

9.1 INTRODUCCIÓN

Detección de movimiento: Tema fundamental en visión artificial y la robótica, posee múltiples aplicaciones (seguimiento de objetos, cálculo de la estructura a partir del movimiento, reconstrucción 3D, y representación y compresión de vídeo).

Objetivos: Calcular el movimiento en la escena 3D donde pueden darse varias posibilidades, de la forma 3D-3D (registro de dos conjuntos 3D, ICP), 2D-3D (características 2D sobre datos 3D, RANSAC) o 2D-2D (inferir movimiento 3D en base a movimientos detectados en imágenes 2D).

Emparejamiento de datos en dos imágenes consecutivas: **Métodos dispersos** (cuando utilizan alguna característica extraída de las imágenes) y **métodos densos** (cuando tienden a emparejar todos los puntos de las imágenes).

9.2 CAMPO DE MOVIMIENTO

Campo de movimiento: En una escena 3D, un punto P se mueve siguiendo un vector V_p , la proyección de P en la imagen es un punto m que tiene asociado un vector v_m , donde la posición de m en función del tiempo depende tanto del desplazamiento de P como de su profundidad Z relativa a la cámara. Sabiendo que

$P = [X, Y, Z]^T$ y $m = [x, y]^T$, se tiene que:

$$\frac{dP}{dt} = \frac{dZ}{dt} [x, y, 1]^T + Z \left[\frac{dx}{dt}, \frac{dy}{dt}, 0 \right]^T; V_p = (V_p^T k) \hat{m} + Z v_m; v_m = \frac{1}{Z} (V_p - (V_p^T k) \hat{m})$$

donde el **primer término** representa el efecto del cambio de profundidad Z en el movimiento de la imagen, y el **segundo término** representa la velocidad del punto en la imagen (escala por la profundidad).

Parámetros: k (vector unitario en la dirección de profundidad), V_p (velocidad del punto en 2D), $(V_p^T k) \hat{m}$ (proyección de V_p en la dirección k) y $Z v_m$ (relaciona el campo de movimiento en la imagen con la profundidad).

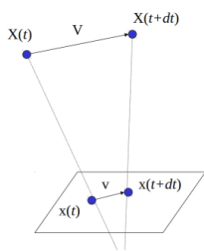
Valor de V_p : Si Z es grande o el objeto está lejos v_m es pequeño (el movimiento aparente en la imagen es menor), y si Z es pequeño o el objeto está cerca v_m es grande (el objeto se desplaza más en la imagen para la misma velocidad).

9.3 FLUJO ÓPTICO Y CÁLCULO DEL FLUJO ÓPTICO

Flujo óptico: Patrón del movimiento aparente de los objetos, superficies y bordes en una escena causado por el movimiento relativo entre un observador (un ojo o una cámara) y la escena. Aunque, en general, el flujo óptico no es igual al campo de movimiento, ya que es lo único que se tiene para estimarlo, por lo que se asume que el campo de movimiento no estará muy lejos del flujo óptico detectado en una sucesión de imágenes.

Cálculo del flujo óptico: Dada una distribución espacio-temporal de intensidad procedente de la **secuencia de imágenes** $I(x, y, t)$, tiene como objetivo obtener un **vector de velocidad** $v = (u, v)$.

Velocidad aparente del píxel en la imagen: $m = [x, y]^T$, $v_m = [v_x, v_y]^T = \left[\frac{dx}{dt}, \frac{dy}{dt} \right]^T$



Parámetros: $X(t)$ (punto 3D en movimiento) y $x(t) = (x(t), y(t))$ (proyección de X en la imagen $I(x, y, t)$).

Cambio de intensidad nulo en el tiempo: $I(x + v_x dt, y + v_y dt, t + dt) = I(x, y, t)$.

Cambio de intensidad leve (Series de Taylor): $I(x, y, t) + \frac{dl}{dx} v_x dt + \frac{dl}{dy} v_y dt + \frac{dl}{dt} dt + O(dt^2) = I(x, y, t)$,

$$\frac{dl}{dx} v_x dt + \frac{dl}{dy} v_y dt + \frac{dl}{dt} dt = 0 \text{ y } \nabla I * v_m + \frac{dl}{dt} = 0.$$

9.4 PROBLEMA DE LA APERTURA

Problema de la apertura: El campo de movimiento y el flujo óptico pueden no coincidir, ya que se tiene una ecuación con dos incógnitas, por lo que no se puede determinar unívocamente el flujo óptico y sólo se podrá calcular el flujo óptico en la dirección normal del gradiente. Cada neurona del sistema visual es sensible a la entrada en una pequeña parte del campo de visión, ya que la está mirando a través de una pequeña ventana o apertura, aunque si aparece una variedad de contornos con diferentes orientaciones moviéndose a diferentes velocidades podrían causar respuestas visuales idénticas.

$$\left(\frac{dI}{dx}, \frac{dI}{dy}\right) * (V_x, V_y) = - \frac{dI}{dt}$$

Parámetros: I_x, I_y (gradientes de intensidad en las direcciones x e y), I_t (cambio temporal de intensidad) y V_x, V_y (flujo óptico buscado).

9.5 MÉTODO DE LUCAS-KANADE

Consistencia luminosa: Asume que los keypoints cambian su posición, pero no su intensidad. Cualquier cambio en la intensidad de la imagen se debe al movimiento, la iluminación de la escena permanece constante y las superficies de los objetos son opacas (superficies Lambertianas).

Coherencia espacial: Puntos vecinos pertenecen a la misma superficie y comparten misma velocidad.

Persistencia temporal: Los movimientos de un objeto en la escena son muy lentos en cuanto al tiempo Δt . Los incrementos temporales son más rápidos que los movimientos de los objetos.

Método local: Analiza la vecindad de cada punto y se asume que nosotros vemos la escena a través de una zona limitada (ventana de vecindad $N \times N$), donde la intensidad a dentro de esa ventana es variable y del frame 1 al frame 2 se ha movido a b . De este modo, se asume que a se ha desplazado arriba y a la izquierda a una nueva intensidad b , visible dentro de la ventana.

Movimiento de un píxel (x, y) (movimiento de u / v píxeles en la dirección x / y): $I_x(x, y) * u + I_y(x, y) * v$

Diferencia local en intensidad: $I_x(x, y) * u + I_y(x, y) * v = - I_t(x, y)$, donde el signo negativo es necesario para garantizar que el flujo óptico estimado apunta en la dirección correcta en relación con el cambio de intensidad entre las dos imágenes.

Ecuación lineal de vecindad: $I_x(x + \Delta x, y + \Delta y) * u + I_y(x + \Delta x, y + \Delta y) * v = - I_t(x + \Delta x, y + \Delta y)$

Ecuación lineal de vecindad resumida: $S(u \ v)^T = \bar{t}$, donde S es la matriz $(m \times n) \times 2$ con derivadas parciales de intensidad (I_x, I_y) , \bar{t} es un vector que contiene los $m \times n$ términos $- I_t$ y $(u \ v)^T$ es el vector de flujo óptico que buscamos.

Solución por mínimos cuadrados: $S^T S(u \ v)^T = S^T \bar{t}$, e invirtiendo $S^T S ((u \ v)^T = (S^T S)^{-1} S^T \bar{t})$.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}^{-1} \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix}$$

Idea: Obtener unos puntos para rastrear y se obtienen los vectores de flujo óptico de esos puntos.

Problemas: No funciona bien en zonas homogéneas (problemas de apertura y no trabaja bien con movimientos grandes).

Solución: Utilizar una pirámide, cuando se sube en la pirámide se eliminan los movimientos pequeños y los movimientos grandes se convierten en pequeños, obteniendo así el flujo óptico con la escala.

9.6 MÉTODO DE DIFERENCIAS

Método de diferencias: Sean $f_1(i, j, t_1)$ y $f_2(i, j, t_2)$ dos imágenes consecutivas separadas por un intervalo de tiempo, donde un elemento $d(i, j)$ de la diferencia de imágenes entre f_1 y f_2 puede tener un valor 1, debido a alguna de las siguientes razones ($f_1(i, j)$ es un píxel de un objeto en movimiento y $f_2(i, j)$ es un píxel estático del fondo o viceversa, $f_1(i, j)$ es un píxel de un objeto en movimiento y $f_2(i, j)$ es un píxel de otro objeto en movimiento, $f_1(i, j)$ es un píxel de un objeto en movimiento y $f_2(i, j)$ es un píxel de una parte diferente del mismo objeto, y ruido o imprecisiones en el posicionamiento de la cámara estacionaria).

Método de diferencias acumuladas: $d_{acum}(i, j) = \sum_{k=1}^n a_k |f_1(i, j) - f_k(i, j)|$, donde a_k proporciona la importancia de las imágenes en la secuencia de las n imágenes ($a_k = \frac{k-1}{n-1}$).

TEMA 10: RECONOCIMIENTO DE PATRONES

10.1 INTRODUCCIÓN

Machine Learning / Aprendizaje Automático o Automatizado: Tiene como objetivo desarrollar técnicas que permitan que los ordenadores, o máquinas con capacidad de procesamiento aprendan. Están programadas para adaptar un algoritmo conforme reciben datos (cuantos más datos reciben, mejores serán los algoritmos que crean, siempre y cuando los datos introducidos sean datos fiables y de calidad, ya que si se mete algún dato erróneo puede resultar en un mal aprendizaje). Esto es posible en parte gracias a las redes neuronales.

Algoritmo DL: Aprenden sin intervención humana previa, sacando ellos mismos las conclusiones de todos los datos. Además, son capaces de dotar a los objetos de un significado y manejar conceptos abstractos.

Aprendizaje supervisado: Ajustan el modelo o función de clasificación mediante un conjunto de datos cuya clase es conocida como **conjunto de entrenamiento**. Éste conjunto se representa como $\{(x_1, y_1), \dots, (x_N, y_N)\}$, donde x_i es el i-ésimo vector del conjunto de entrenamiento mientras que y_i es la clase a la que pertenece dicho vector. Una vez ajustado el modelo, la función de clasificación es capaz de asignar la clase y_k a la que pertenecen nuevos elementos x_k , donde el conjunto de elementos de clase desconocida se llama conjunto de test.

Aprendizaje no supervisado: Ajustan un modelo de clasificación a partir de un **conjunto de datos no etiquetados** $\{x_1, \dots, x_N\}$. Para ello, el algoritmo debe encontrar de forma autónoma las características, regularidades y correlaciones de los datos, de forma que pueda separar las categorías similares entre sí.

10.2 K-NN

Algoritmo k-NN (k Nearest Neighbour): Algoritmo de clasificación supervisado basado en criterios de vecindad, cuya idea consiste en que los nuevos ejemplos serán clasificados con la misma clase que tengan la mayor cantidad de vecinos más parecidos a ellos del conjunto de entrenamiento.

Algoritmo k-NN (versión simple): Explora todo el conocimiento almacenado en el conjunto de entrenamiento para determinar cuál será la clase a la que pertenece una muestra nueva, teniendo en cuenta únicamente el vecino más próximo a ella.

Algoritmo k-NN (variante): Pondera la distribución de cada vecino de acuerdo con la distancia entre él y la muestra a ser clasificada, dando mayor peso a los vecinos más cercanos frente a los que puedan estar alejados. Esta variante es muy efectiva ante muchos problemas prácticos y es robusta ante el ruido de los datos y es suficientemente efectiva en conjuntos grandes de datos.

$$w_i = \frac{1}{d(x, x_i)^2} ; \operatorname{argmax}_{v \in V} \sum_{i=1}^k w_i$$

Parámetros: x (ejemplo que se quiere clasificar), V (posibles clases de clasificación), $\{x_1, \dots, x_k\}$ (conjunto de los k ejemplos de entrenamiento más cercanos), w_i (peso de x_i respecto a x), donde la **clase asignada a x** es aquella que verifique que la suma de los pesos de sus representantes sea máxima.

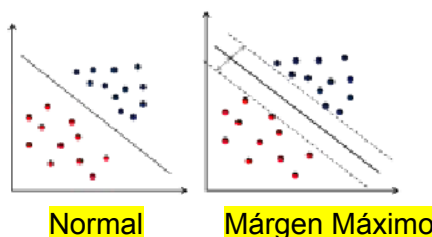
Problema: Requiere mucha memoria y tiempo de ejecución porque hay que almacenar continuamente todos los datos que definen el espacio de ejemplos inicial, ya que muchas de las muestras iniciales no son necesarias para clasificar a las demás, ya que su información es redundante con las otras existentes.

k-NN condensado: Dado un orden en los datos de entrada, cada ejemplo del conjunto se clasifica por medio de K-NN haciendo uso únicamente de los datos si la clasificación obtenida coincide con la real, por lo que ese ejemplo se elimina de los datos, si no, permanece. Dependiendo del orden dado a los datos, puede tener el problema de conservar los datos que introducen ruido al sistema.

k-NN reducido: Comienza con el conjunto completo de datos, y se eliminan aquellos que no afectan a la clasificación del resto de datos de entrada. Al contrario de la condensación, este método es capaz de eliminar las muestras que producen ruido, y guarda aquellas que son críticas para la clasificación.

10.3 SVM

SVM (Support Vector Machines): Consiste en un clasificador biclase (la mayoría de sus implementaciones soportan clasificación multi-clase) cuyo objetivo es inferir una frontera de decisión (lineal o no) en el espacio de características, de modo que las observaciones posteriores se clasifican automáticamente en uno de los dos grupos definidos por dicha frontera (hiperplano).

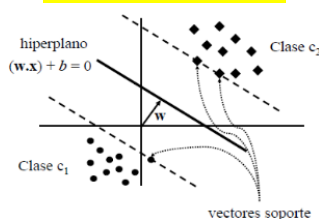


Función de decisión (parámetros): $f(x)$ (función de decisión que clasifica las muestras de entrada), x (vector de características de entrada), w (vector de pesos), b (término de sesgo/intercepción), $*$ (producto escalar entre w y x) y signo (función que devuelve el signo del resultado).

$$f(x) = \text{signo}(w * x + b)$$

Problema de optimización (parámetros): Minimiza la norma del vector de pesos sujeto a ciertas restricciones, donde M (margen), (x_i, y_i) (compuesto por un vector de entrada y etiquetas de clase con el valor ± 1), w (vector de pesos), b (término de sesgo) e i (punto de datos de entrenamiento).

$$y_i(w * x_i + b) \geq M$$



10.4 RED NEURONAL MONOCAPA

Red neuronal monocapa: Compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos.

Perceptrón simple: Lee los valores de entrada y las suma de acuerdo a unos pesos, cuyo resultado se introduce en una función de activación que genera el resultado final. Su entrenamiento consiste en la determinación de los pesos sinápticos y del umbral que mejor hagan que la entrada se ajuste a la salida siguiendo un proceso adaptativo (modifica valores aleatorios iniciales según la diferencia entre los valores deseados y los calculados por la red). Por último, mencionar que el perceptrón aprende de forma iterativa y solo es capaz de representar funciones lineales ya que no dispone de capas ocultas (perceptrón multicapa).

Paso 1 (inicialización de los pesos y del umbral): Se asignan valores aleatorios a cada uno de los pesos w_i ($i = 1, 2, \dots, n, n + 1$ y al bias θ).

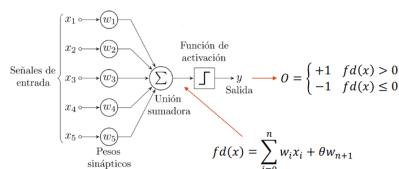
Paso 2 (presentación de un nuevo par): Presenta un nuevo par de entrada $x_p = \{x_1, x_2, \dots, x_n\}$ junto con la salida esperada $fd_i(k)$.

Paso 3 (cálculo de la salida actual): $y(k) = f[w^t x_i - w_{n+1}]$, siendo en este caso f la función de transferencia escalón.

Paso 4 (adaptación de los pesos): $w_i(k + 1) = w_i(k) + \alpha(k)[fd_i(k) - y(k)]x_i$, donde la salida esperada $fd_i(k)$ es 1 si el patrón pertenece a la clase A y -1 si es de la clase B, cuya tasa de aprendizaje viene representada por $\alpha(k)$.

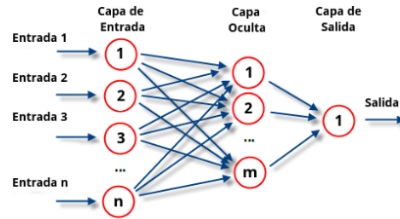
Paso 5 (criterio de parada): Si no hay convergencia (los pesos cambian), se vuelve al paso 2.

Decisión: $x \in c_i$ si $fd_i(x) > fd_j(x)$, donde $\forall j = 1, 2$ para $j \neq i$.



10.5 RED NEURONAL MULTICAPA

Red neuronal multicapa: Generalización de la red neuronal monocapa que dispone de un conjunto de capas intermedias/ocultas entre la capa de entrada y la de salida, la cual puede estar total o parcialmente conectada dependiendo del número de conexiones que presente la red.



Perceptrón multicapa: Evoluciona del perceptrón simple e incorpora capas de neuronas ocultas, consiguiendo así representar funciones no lineales. Está compuesto por una capa de entrada, una o varias capas de salida y n capas ocultas intermedias, caracterizado por tener salidas disjuntas pero relacionadas entre sí, de tal forma que la salida de una neurona es la entrada de la siguiente.

Fase de propagación: Calcula el resultado de la salida de la red desde los valores de entrada hacia delante.

Fase de aprendizaje: Los errores obtenidos a la salida del perceptrón se van propagando hacia atrás con el objetivo de modificar los pesos de las conexiones para que el valor estimado de la red se asemeje cada vez más al real, los cuales se aproximan mediante la función gradiente de error.

10.6 ALGORITMO K-MEANS

Algoritmo K-means: Aplicable cuando se tenga una representación de los datos como elementos del espacio métrico, el cual intenta encontrar una partición de las muestras en K agrupaciones, de forma que cada ejemplo pertenezca a cada una de ellas, concretamente a aquella cuyo centroide esté más cerca (el valor de K no se conoce a priori y depende completamente de los datos con los que se trabaje).

Varianza total del sistema (parámetros): Se intentará minimizar si c_i es el centroide de la agrupación i-ésima y $\{x_j^i\}$ el conjunto de elementos clasificados en esa agrupación.

$$\sum_i \sum_j d(x_j^i, c_i)^2$$

Algoritmo: **Paso 1** (seleccionar al azar K puntos del conjunto de datos como centros iniciales de los grupos), **paso 2** (asignar el resto de ejemplos al centro más cercano teniendo K agrupaciones iniciales), **paso 3** (calcular el centroide de los grupos obtenidos), **paso 4** (reasignar los centros a estos centroides), **paso 5** (repetir desde el paso 2 hasta que no haya reasignación de centros, o que los últimos desplazamientos estén por debajo de un umbral y no haya cambios en las agrupaciones obtenidas).

10.7 MODELO SOM

Modelo SOM (Self Organizing Maps): Compuesto por una **capa de entrada** (formada por N neuronas, una por cada variable de entrada, encargada de recibir y transmitir a la capa de salida la información procedente del exterior) y una **capa de salida** (formada por M neuronas, encargada de procesar la información y formar el mapa de rasgos).

Características: Las conexiones entre las dos capas que forman la red son siempre hacia delante, es decir, desde la entrada hasta la salida; cada neurona de entrada i está conectada con cada una de las neuronas de salida j mediante un peso w_{ij} ; las neuronas de salida tienen asociado un vector de pesos W_j llamado vector de referencia, que constituye el vector prototipo/promedio de la categoría representada por la neurona de salida j , donde las neuronas adyacentes pertenecen a una vecindad N_j de la neurona j , además de que éste se acerca a las áreas donde la densidad de datos es alta; la topología y el número de neuronas permanece fijo desde el principio; el número de neuronas determina la suavidad de la proyección, influyendo así en el ajuste y la capacidad de generalización del SOM; y por último, durante la fase de entrenamiento del SOM, éste forma una red elástica que se pliega dentro de la nube de datos originales, cuyo algoritmo controla la red de modo que tiende a aproximar la densidad de los datos.

} 10.8 ALGORITMO DE KOHONEN

Paso 1: Un vector x es seleccionado al azar del conjunto de datos y se calcula su distancia/similitud a los vectores de referencia, usando por ejemplo, la distancia euclídea.

$$||x - m_c|| = \min_j \{ ||x - m_j|| \}$$

Paso 2: Una vez que se ha encontrado el BMU (Best Matching Unit) el resto de valores se actualizan, donde el BMU y sus valores se mueven cerca del vector x en el espacio de datos, cuya magnitud está regida por la tasa de aprendizaje α . Mientras se produce el proceso de actualización y se asignan nuevos vectores al mapa, la tasa de aprendizaje crece gradualmente hacia 0, además del radio de vecindad. **La regla de actualización para el vector de referencia dado i es la siguiente:**

$$m_j(t+1) = \begin{cases} m_j(t) + \alpha(t) (x(t) - m_j(t)) & j \in N_c(t) \\ m_j(t) & j \notin N_c(t) \end{cases}$$

} 10.9 MEDIDAS DE CALIDAD Y PRECISIÓN DEL MAPA

Medida de preservación de la proyección: Describe cómo se adaptan o responden las neuronas a los datos, ya que habitualmente, el número de datos es mayor que el número de neuronas y el error de precisión siempre es distinto de 0. Para calcular la precisión de la proyección se usa el error medio de cuantificación sobre el conjunto completo de datos:

$$\varepsilon_q = \frac{1}{N} \sum_{i=1}^N ||x - m_c||$$

Medida de preservación de la topología: Describe la manera en la que el SOM preserva la topología del conjunto de datos. Esta medida considera la estructura del mapa. En un mapa que esté retorcido de manera extraña, el error topográfico es grande incluso si el error de precisión es pequeño.

$$\varepsilon_q = \frac{1}{N} \sum_{i=1}^N u(x_k)$$

Parámetros: $u(x_k)$ es igual a 1 si el primer y el segundo BMU de x_k no están próximos el uno al otro. De otro modo, $u(x_k)$ es igual a 0.

} 10.10 R-CNN

R-CNN original: Detector de objetos que requería un algoritmo como la búsqueda selectiva/equivalente para proponer Bounding Boxes candidatos que pudieran contener objetos. Luego, estas regiones se pasaron a una CNN para su clasificación, lo que finalmente condujo a uno de los primeros detectores de objetos basados en aprendizaje profundo. Sin embargo, este método estándar era tremendamente lento y no era un detector de objetos completo de un extremo a otro.

Fast R-CNN: Aumentó la precisión y redujo el tiempo necesario para realizar un pase hacia adelante. Sin embargo, el modelo aún se basaba en un algoritmo de propuesta de región externa.

Faster R-CNN: Detector de objetos de aprendizaje profundo de extremo a extremo por eliminar el requisito de búsqueda selectiva, y en su lugar, confiar en una red de propuesta de región (RPN) que es totalmente convolucional y puede predecir los cuadros delimitadores de objetos y las puntuaciones de objetividad (probabilidad de que una región de una imagen pueda contener una imagen).

} 10.11 SSD Y YOLO

SSD y YOLO: Aumentaron la velocidad de los detectores de objetos basándose en el aprendizaje profundo, pasando de 5 FPS a 45 FPS. Estos algoritmos tratan la detección de objetos como un problema de regresión, tomando una imagen de entrada dada y aprendiendo simultáneamente las coordenadas del cuadro delimitador y las probabilidades de etiqueta de clase correspondientes.

10.12 PROCESAMIENTO DE IMÁGENES EN YOLO

División de la imagen en cuadrículas: YOLO divide la imagen en una cuadrícula de celdas, donde cada celda de la cuadrícula se encarga de predecir un conjunto fijo de cuadros delimitadores (Bounding Boxes) y las probabilidades asociadas con las clases de objetos.

Predicción de cuadros delimitadores y clases: Cada celda predice sobre los cuadros delimitadores que contienen los objetos, donde se asocia un puntaje de confianza a cada cuadro, que indica la certeza de que el cuadro delimitador contiene un objeto y una probabilidad para cada clase de objeto posible.

Optimización del modelo: Durante el entrenamiento, YOLO optimiza los pesos de la red para mejorar la precisión de las detecciones utilizando una función de pérdida que tiene en cuenta tanto la precisión de la detección de los cuadros delimitadores como la clasificación de los objetos.

Supresión de No Máximos: Después de las predicciones, Yolo aplica la supresión de No Máximos para eliminar detecciones redundantes y mantener sólo las más confiables. Se comparan los puntajes de confianza de los cuadros delimitadores y se eliminan aquellos que tienen una alta superposición con cuadros delimitadores más confiables.