Universidad Autónoma Chapingo

Departamento de Mecánica Agrícola

Ingeniería Mecatrónica Agrícola

Informe 4: Búsqueda y detección de características

Asignatura:

Visión por computadora

Nombre del profesor:

Luis Arturo Soriano Avendaño

Alumno:

Cocotle Lara Jym Emmanuel [1710451-3]

GRADO:	**GRUPO:**
6°	**7**

Fecha de entrega: 06/07/2021

Chapingo, Texcoco Edo. México

Contenido

Introducción

Uno de los principales objetivos de la visión por computadora es poder diferenciar los objetos presentes en una imagen (en este caso, en una imagen digital), de tal manera, que esto logre que la identificación de estos (un proceso posterior) sea una tarea más fácil de realizar. Dentro de la visión computacional, podemos encontrar el procesamiento de imágenes, y dentro de este, una parte muy importante se encarga del análisis de estas. Esto es, dada una imagen, lo que deseamos obtener es una descripción de dicha imagen.

Es decir, dada una imagen, el análisis se encarga de entregar información de ella. Por lo que, en todos estos ejemplos, el análisis depende primeramente de detectar determinadas partes de la imagen (regiones u objetos). Para generar tal descripción es necesario segmentar (o separar) adecuadamente e identificar la región deseada.

El separar la imagen en unidades significativas es un paso importante en visión computacional para llegar al reconocimiento de objetos. Este proceso se conoce como segmentación. Una forma de segmentar la imagen es mediante la determinación de los bordes. El dual de este problema es determinar las regiones; es decir, las partes o segmentos que se puedan considerar como unidades significativas. Esto ayuda a obtener una versión más compacta de la información de bajo nivel, ya que, en vez de miles o millones de pixeles, se puede llegar a decenas de regiones, y de ahí reconocer los objetos. Las características más comunes para delimitar o segmentar regiones son: intensidad de los pixeles, textura, color y gradiente [1].

En este informe se pretende dar a conocer los métodos de detección de imágenes y la forma en que se lleva a cabo dicha detección, con el fin de comprender de manera más amplia los conceptos vistos en clases.

Desarrollo

Puntos y parches de superficie

Las entidades de puntos se pueden utilizar para encontrar un conjunto disperso de ubicaciones correspondientes en diferentes imágenes, a menudo como un precursor para calcular la postura de la cámara, que es un requisito previo para calcular un conjunto más denso de correspondencias utilizando la coincidencia estéreo. Estas correspondencias también se pueden utilizar para alinear diferentes imágenes, por ejemplo, al coser mosaicos de imágenes o realizar estabilización de vídeo. También se utilizan ampliamente para realizar el reconocimiento de instancias de objetos y categorías. Una ventaja clave de los puntos clave es que permiten la coincidencia incluso en presencia de desorden (oclusión) y cambios de escala y orientación a gran escala.

Las técnicas de correspondencia basadas en características se han utilizado desde los primeros días de la coincidencia estéreo y más recientemente han ganado popularidad para aplicaciones de costura de imágenes, así como el modelado 3D totalmente automatizado.

Hay dos enfoques principales para encontrar puntos de entidad y sus correspondencias. La primera es encontrar entidades en una imagen que se puedan rastrear con precisión utilizando una técnica de búsqueda local, como correlación o mínimos cuadrado. La segunda es detectar de forma independiente las características de todas las imágenes consideradas y, a continuación, hacer coincidir las características en función de su apariencia local. El primer enfoque es más adecuado cuando se toman imágenes desde puntos de vista cercanos o en rápida sucesión (por ejemplo, secuencias de video), mientras que el segundo es más adecuado cuando se espera una gran cantidad de movimiento o

cambio de apariencia, por ejemplo, en la unión de panoramas, el establecimiento de correspondencias en estéreo de línea de base amplia, o la realización de reconocimiento de objetos.

Durante la etapa de detección de características (extracción), se busca en cada imagen ubicaciones que probablemente coincidan bien en otras imágenes. En la fase de descripción de características, cada región alrededor de las ubicaciones de puntos clave detectadas se convierte en un descriptor más compacto y estable (invariable) que se puede comparar con otros descriptores. La etapa de coincidencia de características busca de manera eficiente candidatos probables coincidentes en otras imágenes. La etapa de seguimiento de entidades es una alternativa a la tercera etapa que solo busca en una pequeña vecindad alrededor de cada entidad detectada y, por lo tanto, es más adecuada para el procesamiento de vídeo [2].

Detectores de características

Como puede notar, los parches sin textura son casi imposibles de localizar. Los parches con grandes cambios de contraste (gradientes) son más fáciles de localizar, aunque los segmentos de línea recta en una sola orientación sufren del problema de apertura, es decir, sólo es posible alinear los parches a lo largo de la dirección normal a la dirección del borde. Los parches con degradados en al menos dos orientaciones (significativamente) diferentes son los más fáciles de localizar. Estas intuiciones se pueden formalizar mirando el criterio de coincidencia más simple posible para comparar dos parches de imagen, es decir, su diferencia cuadrada sumada (ponderada):

$$E_{WSSD}(u) = \sum_i w(x_i)[I_1(x_i + u) - I_0(x_i)]^2$$

Donde I_0 y I_1 son las dos imágenes que se comparan, $u = (u, v)$ es el vector de desplazamiento, $w(x)$ es una función de ponderación (o ventana) que varía espacialmente, y la suma i está sobre todos los píxeles del parche. Al realizar la detección de características, no sabemos con qué otras ubicaciones de imágenes terminarán coincidiendo la entidad. Por lo tanto, solo podemos calcular qué tan estable es esta métrica con respecto a pequeñas variaciones en la posición Δu comparando un parche de imagen contra sí mismo, lo que se conoce como una función de correlación automática o superficie.

$$E_{AC}(\Delta u) = \sum_i w(x_i)[I_0(x_i + \Delta u) - I_0(x_i)]^2$$

Usando una expansión de la serie de Taylor de la función de imagen $I_0(x_i + \Delta u) \approx I_0(x_i) + \nabla I_0(x_i)(\Delta u)$, podemos aproximar la superficie de autocorrección como:

$$E_{AC} = \Delta u^T A \Delta u$$

El clásico detector "Harris" utiliza un filtro [-2 -1 0 1 2], pero variantes más modernas modifica la imagen con los derivados horizontales y verticales de un Gaussiano (típicamente con $\sigma = 1$). La matriz de correlación automática A se puede escribir como:

$$A = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Donde hemos sustituido las sumas ponderadas por convoluciones discretas por el núcleo de ponderación w. Esta matriz se puede interpretar como una imagen tensorial (multibanda), donde los productos externos de los gradientes ∇I se transforman con una función de ponderación w para proporcionar una estimación por píxel de la forma local (cuadrática) de la función de autoconservación. La inversa de la matriz A proporciona un límite inferior en la incertidumbre en la ubicación de un parche coincidente. Por lo tanto, es un indicador útil de qué parches se pueden emparejar de manera confiable. La forma más fácil de visualizar y razonar sobre esta incertidumbre es realizar un análisis de valores propios de la matriz de auto coronación A, que produce dos valores propios (λ_0, λ_1) y dos direcciones de auto vector. Dado que la incertidumbre más grande depende del valor propio más pequeño, es decir, $\lambda_0^{-\frac{1}{2}}$, tiene sentido encontrar máximos en el valor propio más pequeño para localizar buenas características para rastrear [2].

Forstner–Harris.

Mientras que Lucas y Kanade (1981) fueron los primeros en analizar la estructura de incertidumbre de la matriz de autocorrección, lo hicieron en el contexto de asociar certezas con mediciones de flujo óptico. Forstner en 1986 y Harris y Stephens en 1988, fueron los primeros en proponer el uso de máximos locales en medidas escalares rotacionalmente invariantes derivadas de la matriz de autocorrelación para localizar puntos clave con el fin de la coincidencia de entidades dispersas; Triggs (2004) ofrece revisiones históricas más detalladas de los algoritmos de detección de características. Ambas técnicas también propusieron el uso de una ventana de ponderación gaussiana en lugar de los parches cuadrados utilizados anteriormente, lo que hace que la respuesta del detector sea insensible a las rotaciones de imágenes en el plano.

El valor propio mínimo λ_0 no es la única cantidad que se puede utilizar para encontrar puntos clave. Una cantidad más simple, propuesta por Harris y Stephens (1988), es:

$$\det(A) - \alpha\, trace(A)^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

Con $\alpha = 0.06$. A diferencia del análisis de valores propios, esta cantidad no requiere el uso de raíces cuadradas y, sin embargo, sigue siendo invariante rotacionalmente y también pondera hacia abajo las entidades de borde donde $\lambda_1 \gg \lambda_0$. Triggs (2004) sugiere

usar la cantidad $\lambda_0 - \alpha\lambda_1$ (por ejemplo, con α=0.05), que también reduce la respuesta en los bordes 1D, donde los errores de seudónimo a veces inflan el valor propio más pequeño. También muestra cómo el Hessian básico de 2×2 se puede extender a movimientos paramétricos para detectar puntos que también son localizables con precisión en escala y rotación. Brown, Szeliski y Winder, por otro lado, usan la media armónica que es una función más suave en la región donde $\lambda_0 \approx \lambda_1$ [2].

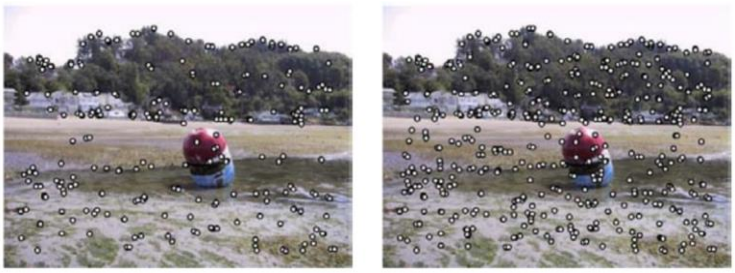
Esquema de un algoritmo de detección de características básico.

- ✓ Calcular las derivadas horizontales y verticales de la imagen I_x *e* I_y convolucionando la imagen original con derivadas de Gauss.
- ✓ Calcular las tres imágenes correspondientes a los productos externos de estos degradados. (La matriz A es simétrica, por lo que solo se necesitan tres entradas).
- ✓ Transformar cada una de estas imágenes con un gaussiano más grande.
- ✓ Calcular una medida de interés escalar mediante una de las fórmulas descritas anteriormente.
- ✓ Buscar máximos locales por encima de un determinado umbral e informarlos como ubicaciones de puntos de entidad detectados.

1.- Imagen de muestra, respuesta de Harris y respuesta de DoG [2].

Supresión adaptativa no máxima (ANMS).

Si bien la mayoría de los detectores de características simplemente buscan máximos locales en la función de interés, esto puede conducir a una distribución desigual de los puntos de entidad en toda la imagen, por ejemplo, los puntos serán más densos en regiones de mayor contraste. Para mitigar este problema, Brown, Szeliski y Winder, solo detectan características que son máximos locales y cuyo valor de respuesta es significativamente (10%) mayor que el de todos sus vecinos dentro de un radio r. Idean una manera eficiente de asociar los radios de supresión con todos los máximos locales clasificándolos primero por su fuerza de respuesta y luego creando una segunda lista ordenada por la disminución del radio de supresión.



2.- ANMS 250, r = 24 y ANMS 500, r = 16 [2].

Medición de la repetibilidad.

Schmid, Mohr y Bauckhage fueron los primeros en proponer la medición de la repetibilidad de los detectores de características, que definen como la frecuencia con la que los puntos clave detectados en una imagen se encuentran dentro (digamos, $\epsilon = 1.5$) píxeles de la ubicación correspondiente en una imagen transformada. En su papel, transforman sus imágenes planas aplicando rotaciones, cambios de escala, cambios de iluminación, cambios de punto de vista y agregando ruido. También miden el contenido de información disponible en cada punto de entidad detectado, que definen como la entropía de un conjunto de descriptores de escala de grises locales invariantes rotacionalmente. Entre las técnicas que examinan, encuentran que la versión mejorada (derivada gaussiana) del operador de Harris con $\sigma_d = 1$ (escala de la derivada gaussiana) y $\sigma_i = 2$ (escala de la integración gaussiana) funciona mejor [2].

Invariancia de escala

En muchas situaciones, la detección de entidades a la mejor escala estable posible puede no ser apropiada. Por ejemplo, al hacer coincidir imágenes con pocos detalles de alta frecuencia (por ejemplo, nubes), es posible que no existan entidades de escala fina. Una solución al problema es extraer entidades en una variedad de escalas, por ejemplo, realizando las mismas operaciones a varias resoluciones en una pirámide y luego haciendo coincidir las entidades en el mismo nivel. Este tipo de enfoque es adecuado cuando las imágenes que se emparejan no sufren cambios a gran escala, por ejemplo, al hacer coincidir imágenes aéreas sucesivas tomadas desde un avión o al coser panoramas tomados con una cámara de distancia focal fija. Sin embargo, para la mayoría de las aplicaciones de reconocimiento de objetos, la escala del objeto en la imagen es desconocida. En lugar de extraer entidades en muchas escalas diferentes y luego hacer coincidir todas ellas, es más eficiente extraer entidades que son estables tanto en la ubicación como en la escala.

Las primeras investigaciones sobre la selección de escalas fueron realizadas por Lindeberg, quien propuso por primera vez el uso de extrema en la función Laplaciana de Gauss (LoG) como ubicaciones de puntos de interés. Basándose en este trabajo, Lowe propuso calcular un conjunto de filtros gaussianos de diferencia de sub-octava, buscando máximos 3D

(espacio + escala) en la estructura resultante, y luego calcular una ubicación de espacio + escala de subpíxeles usando un ajuste cuadrático. El número de niveles de sub-octava se determinó, después de una cuidadosa investigación empírica, que es de tres, lo que corresponde a una pirámide de cuarto de octava, que es la misma que la utilizada por Triggs. Al igual que con el operador de Harris, los píxeles donde hay una fuerte asimetría en la curvatura local de la función del indicador (en este caso, el DoG) se rechazan. Esto es puesto en ejecución primero computando al Hessian local de la imagen D de la diferencia,

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

y, a continuación, rechazar los puntos clave para los que

$$\frac{Tr(H)^2}{Det(H)} > 10$$

Si bien la transformación de entidades invariantes de escala (SIFT) de Lowe funciona bien en la práctica, no se basa en la misma base teórica de máxima estabilidad espacial que los detectores basados en correlación automática. (De hecho, sus ubicaciones de detección son a menudo complementarias de las producidas por tales técnicas y, por lo tanto, se pueden utilizar junto con estos otros enfoques). Con el fin de añadir un mecanismo de selección de escala al detector de esquinas de Harris, Mikolajczyk y Schmid (2004) evalúan el laplaciano de la función gaussiana en cada punto de Harris detectado (en una pirámide multiescala) y mantienen sólo aquellos puntos para los que el laplaciano es extremo (más grande o más pequeño que sus valores de nivel más grueso y más fino). También se propone y evalúa un refinamiento iterativo opcional tanto para la escala como para la posición [2].

Estimación de la invariancia rotacional y la orientación

Además de tratar con los cambios de escala, la mayoría de los algoritmos de coincidencia de imágenes y reconocimiento de objetos deben tratar con (al menos) la rotación de imágenes en el plano. Una forma de tratar este problema es diseñar descriptores que sean rotacionalmente invariantes, pero tales descriptores tienen una discriminabilidad pobre, es decir, asignan parches de aspecto diferente al mismo descriptor.

Un mejor método es estimar una orientación dominante en cada punto clave detectado. Una vez estimadas la orientación local y la escala de un punto clave, se puede extraer un parche escalado y orientado alrededor del punto detectado y utilizarlo para formar un descriptor de entidades. La estimación de orientación más simple posible es el gradiente promedio dentro de una región alrededor del punto clave. Si se utiliza una función de ponderación gaussiana, este gradiente promedio es equivalente a un filtro orientable de

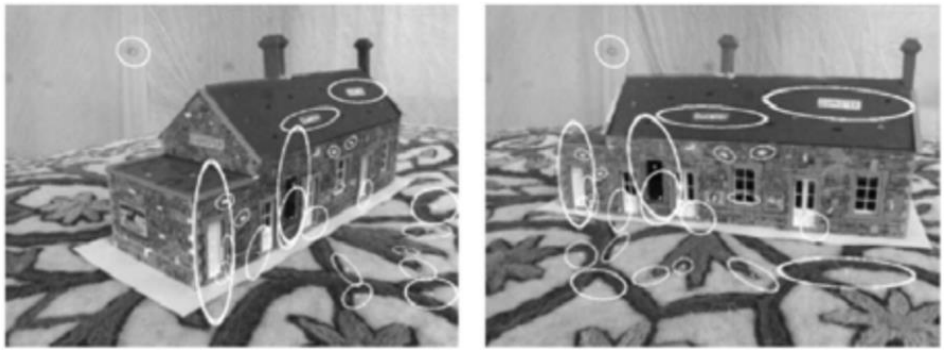
primer orden, es decir, se puede calcular utilizando una convolución de imagen con las derivadas horizontales y verticales del filtro gaussiano. Para que esta estimación sea más confiable, generalmente es preferible usar una ventana de agregación más grande (tamaño de kernel gaussiano) que una ventana de detección.

A veces, sin embargo, el gradiente promediado (con signo) en una región puede ser pequeño y, por lo tanto, un indicador poco fiable de orientación. Una técnica más confiable es observar el histograma de orientaciones calculadas alrededor del punto clave. Lowe calcula un histograma de 36-bin de orientaciones de borde ponderadas por la magnitud del gradiente y la distancia gaussiana al centro, encuentra todos los picos dentro del 80% del máximo global y, a continuación, calcula una estimación de orientación más precisa utilizando un ajuste parabólico de tres bins [2].

Invariancia afín

Si bien la invariancia de escala y rotación es altamente deseable, para muchas aplicaciones, como la coincidencia estéreo de línea de base amplia o reconocimiento de la localización), se prefiere la invariancia afín completa.

Los detectores invariantes afines no solo responden en ubicaciones consistentes después de los cambios de escala y orientación, sino que también responden consistentemente a través de deformaciones afines como el escorzo de perspectiva (local). De hecho, para un parche lo suficientemente pequeño, cualquier deformación continua de la imagen se puede aproximar bien por una deformación afín. Para introducir la invariancia afín, varios autores han propuesto ajustar una elipse a la autocorrelación o matriz hessiana (utilizando el análisis de valores propios) y luego usar los ejes principales y las relaciones de este ajuste como el marco de coordenadas afines. Otro importante detector de región invariante afín es el detector de región extrema máxima estable (MSER) desarrollado por Matas, Chum, Urban. Para detectar MSERs, las regiones binarias se calculan estableciendo un umbral de la imagen en todos los niveles de gris posibles (por lo tanto, la técnica solo funciona para imágenes en escala de grises). Esta operación se puede realizar de manera eficiente ordenando primero todos los píxeles por valor gris y luego agregando incrementalmente píxeles a cada componente conectado a medida que se cambia el umbral. A medida que se cambia el umbral, se supervisa el área de cada componente (región); las regiones cuya tasa de cambio de superficie con respecto al umbral es mínima se definen como máximamente estables. Por lo tanto, estas regiones son invariantes tanto para las transformaciones geométricas afines como a las fotométricas (ganancia lineal de sesgo o monotónica suave). Si lo desea, un marco de coordenadas afines se puede ajustar a cada región detectada utilizando su matriz de momento [2].



3.- Detectores de regiones afines utilizados para hacer coincidir dos imágenes tomadas desde puntos de vista diferentes.

Descriptores de características

Después de detectar las entidades (puntos clave), debemos coincidir con ellas, es decir, debemos determinar qué características provienen de las ubicaciones correspondientes en diferentes imágenes. En algunas situaciones, por ejemplo, para secuencias de vídeo o para pares estéreo que han sido rectificados, el movimiento local alrededor de cada punto de la característica puede ser sobre todo traslacional. En este caso, las métricas de error simples, como la suma de las diferencias al cuadrado o la correlación cruzada normalizada, se pueden utilizar para comparar directamente las intensidades en pequeños parches alrededor de cada punto de entidad. Debido a que los puntos de entidad pueden no estar ubicados exactamente, se puede calcular una puntuación de coincidencia más precisa realizando un refinamiento de movimiento incremental, pero esto puede llevar mucho tiempo y, a veces, incluso puede disminuir el rendimiento.

En la mayoría de los casos, sin embargo, la apariencia local de las entidades cambiará de orientación y escala, y a veces incluso sufrirá deformaciones afines. Por lo tanto, suele ser preferible extraer una escala local, una orientación o una estimación de fotograma afín y, a continuación, utilizarla para remuestrear el parche antes de formar el descriptor de entidades. Incluso después de compensar estos cambios, la apariencia local de los parches de imagen generalmente seguirá variando de una imagen a otra [2].

Sesgo y normalización de ganancias (MOPS).

Para las tareas que no exhiben grandes cantidades de escorzo, como la costura de imágenes, los parches de intensidad normalizados simples funcionan razonablemente bien y son fáciles de implementar. Para compensar ligeras inexactitudes en el detector de puntos de entidad (ubicación, orientación y escala), estos parches orientados a escala múltiple (MOPS) se muestrean con un espaciado de cinco píxeles en relación con la escala de detección, utilizando un nivel más grueso de la pirámide de imágenes para evitar el seudónimo. Para compensar las variaciones fotométricas afines (cambios de exposición

lineal o sesgo y ganancia), las intensidades de parche se reescalan para que su media sea cero y su varianza sea una [2].

Transformación de entidad invariante de escala (SIFT).

Las entidades SIFT se forman calculando el gradiente en cada píxel en una ventana de 16×16 alrededor del punto clave detectado, utilizando el nivel apropiado de la pirámide gaussiana en el que se detectó el punto clave. Las magnitudes de gradiente son minimizadas por una función de caída gaussiana, mostrada como un círculo azul adentro con el fin de reducir la influencia de los gradientes lejos del centro, ya que estos se ven más afectados por pequeños errores de registro. En cada cuadrante de 4×4, se forma un histograma de orientación de degradado agregando (conceptualmente) el valor de degradado ponderado a uno de los ocho bins de histograma de orientación. Para reducir los efectos de la ubicación y la orientación dominante, cada una de las 256 magnitudes de gradiente ponderadas originales se agrega suavemente a $2 \times 2 \times 2$ bins de histograma utilizando interpolación trilineal. La distribución suave de valores a bins de histogramas adyacentes es generalmente una buena idea en cualquier aplicación donde se estén calculando histogramas, por ejemplo, para transformaciones de Hough o ecualización de histograma local. Los 128 valores no negativos resultantes forman una versión sin procesar del vector descriptor SIFT. Para reducir los efectos de contraste o ganancia (las variaciones aditivas ya se eliminan por el gradiente), el vector 128-D se normaliza a la longitud de la unidad. Para que el descriptor sea robusto a otras variaciones fotométricas, los valores se recortan a 0,2 y el vector resultante se vuelve a normalizar a la longitud de la unidad [2].

PCA-SIFT

Ke y Sukthankar (2004) proponen una forma más sencilla de calcular descriptores inspirados en SIFT; calcula las derivadas x e y (gradiente) en un parche de 39×39 y luego reduce el vector de 3042 dimensiones resultante a 36 utilizando el análisis de componentes principales (PCA). Otra variante popular de SIFT es SURF (Bay, Tuytelaars y Van Gool 2006), que utiliza filtros de caja para aproximar los derivados e integrales utilizados en SIFT [2].

Histograma de orientación de ubicación de degradado (GLOH).

Este descriptor, desarrollado por Mikolajczyk y Schmid (2005), es una variante de SIFT que utiliza una estructura de bin log-polar en lugar de los cuatro cuadrantes utilizados por Lowe. Los bins espaciales son de radio 6, 11 y 15, con ocho bins angulares (excepto para la región central), para un total de 17 bins espaciales y 16 bins de orientación. El histograma de 272 dimensiones se proyecta en un descriptor de 128 dimensiones utilizando PCA entrenado en una base de datos grande. En su evaluación, Mikolajczyk y

Schmid encontraron que GLOH, que tiene el mejor desempeño en general, supera a SIFT por un pequeño margen [2].

Filtros orientables

Los filtros orientables son combinaciones de derivados de filtros gaussianos que permiten el cálculo rápido de características pares e impares (simétricas y antisimétricas) de borde y de esquina en todas las orientaciones posibles. Debido a que utilizan gaussianos razonablemente amplios, también son algo insensibles a los errores de localización y orientación [2].

Rendimiento de los descriptores locales.

Entre los descriptores locales que Mikolajczyk y Schmid compararon, encontraron que GLOH tuvo el mejor desempeño, seguido de cerca por SIFT. El campo de los descriptores de características continúa evolucionando rápidamente, con algunas de las técnicas más nuevas que analizan la información de color local. Winder y Brown desarrollan un marco de varias etapas para el cálculo de descriptores de características que subsuma tanto SIFT como GLOH y también les permite aprender parámetros óptimos para descriptores más nuevos que superan a los descriptores anteriores ajustados a mano. Hua, Brown y Winder amplían este trabajo aprendiendo proyecciones de menor dimensión de descriptores de dimensiones superiores que tienen el mejor poder discriminativo. Ambos documentos utilizan una base de datos de parches de imágenes del mundo real obtenidos mediante el muestreo de imágenes en ubicaciones que coincidieron de manera confiable utilizando un algoritmo robusto de estructura desde movimiento aplicado a colecciones de fotos de Internet. En trabajos simultáneos, Tola, Lepetit y Fua desarrollaron un descriptor DAISY similar para la coincidencia estéreo densa y optimizaron sus parámetros basados en datos estéreo de verdad de tierra.

Si bien estas técnicas construyen detectores de características que optimizan la repetibilidad en todas las clases de objetos, también es posible desarrollar detectores de características específicas de clase o instancia que maximizan la discriminabilidad de otras clases [2].

Estrategia de coincidencia y tasas de error

Determinar qué coincidencias de características son razonables para procesar más depende del contexto en el que se realiza la coincidencia. Sabemos que es probable que la mayoría de las entidades de una imagen coincidan con la otra imagen, aunque algunas pueden no coincidir porque están ocluidas o su apariencia ha cambiado demasiado. Por otro lado, si estamos tratando de reconocer cuántos objetos conocidos aparecen en una escena desordenada, es posible que la mayoría de las entidades no coincidan. Además, se debe buscar un gran número de objetos potencialmente coincidentes, lo que requiere

estrategias más eficaces. Para empezar, suponemos que los descriptores de entidades se han diseñado para que las distancias euclidianas (magnitud vectorial) en el espacio de entidades se puedan utilizar directamente para clasificar posibles coincidencias. Si resulta que ciertos parámetros (ejes) en un descriptor son más confiables que otros, generalmente es preferible volver a escalar estos ejes antes de tiempo, por ejemplo, determinando cuánto varían cuando se comparan con otras coincidencias buenas conocidas. Un proceso más general, que implica la transformación de vectores de características en una nueva base escalada, se denomina blanqueamiento. Dada una métrica de distancia euclidiana, la estrategia de coincidencia más simple es establecer un umbral (distancia máxima) y devolver todas las coincidencias de otras imágenes dentro de este umbral. Establecer el umbral demasiado alto da como resultado demasiados falsos positivos, es decir, coincidencias incorrectas que se devuelven. Establecer el umbral demasiado bajo da como resultado demasiados falsos negativos, es decir, que se pierdan demasiadas coincidencias correctas [2].

Seguimiento de características

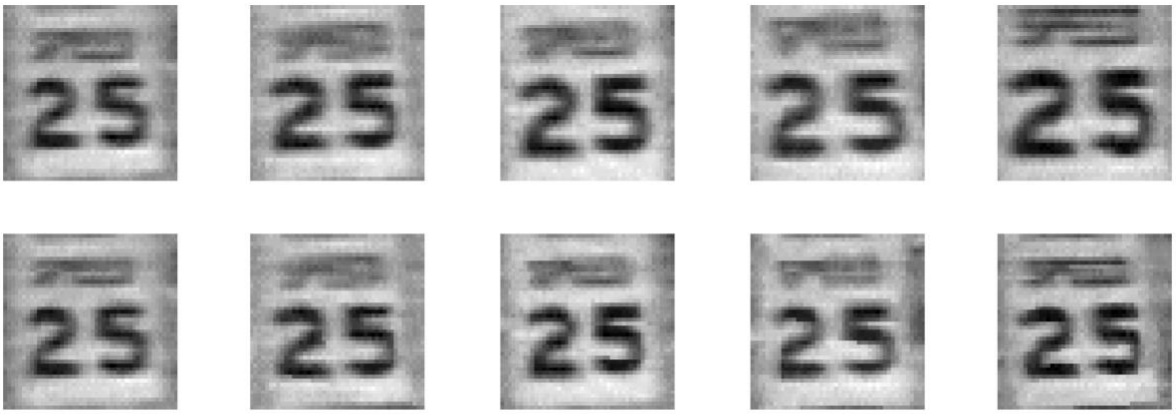
Una alternativa a la búsqueda independiente de entidades en todas las imágenes candidatas y, a continuación, su coincidencia es encontrar un conjunto de ubicaciones de entidades probables en una primera imagen y, a continuación, buscar sus ubicaciones correspondientes en imágenes posteriores. Este tipo de enfoque de detección y seguimiento se utiliza más ampliamente para aplicaciones de seguimiento de video, donde se espera que la cantidad esperada de movimiento y deformación de apariencia entre fotogramas adyacentes sea pequeña. El proceso de selección de buenas características para realizar un seguimiento está estrechamente relacionado con la selección de buenas características para aplicaciones de reconocimiento más generales. En la práctica, las regiones que contienen gradientes altos en ambas direcciones, es decir, que tienen valores propios altos en la matriz de autocorrección, proporcionan ubicaciones estables en las que encontrar correspondencias.

En fotogramas posteriores, la búsqueda de ubicaciones donde el parche correspondiente tiene una diferencia al cuadrado baja a menudo funciona lo suficientemente bien. Sin embargo, si las imágenes están experimentando un cambio de brillo, puede ser preferible compensar explícitamente dichas variaciones o usar una correlación cruzada normalizada. Si el rango de búsqueda es grande, a menudo también es más eficiente utilizar una estrategia de búsqueda jerárquica, que utiliza coincidencias en imágenes de menor resolución para proporcionar mejores conjeturas iniciales y, por lo tanto, acelerar la búsqueda. Las alternativas a esta estrategia implican aprender cuál debe ser el aspecto del remiendo que es seguido y después buscarlo en la vecindad de su posición prevista. Si se realiza un seguimiento de las entidades en secuencias de imágenes más largas, su

apariencia puede sufrir cambios más grandes. A continuación, debe decidir si desea continuar la coincidencia con el parche (característica) detectado originalmente o volver a muestrear cada fotograma posterior en la ubicación coincidente. La estrategia anterior es propensa al fracaso, ya que el parche original puede sufrir cambios de apariencia, como el escorzo. Este último corre el riesgo de que la característica se desvíe de su ubicación original a alguna otra ubicación en la imagen. Matemáticamente, los pequeños errores de registro se combinan para crear un Paseo Aleatorio de Markov, lo que conduce a una mayor deriva con el tiempo.

Una solución preferible es comparar el parche original con ubicaciones de imágenes posteriores utilizando un modelo de movimiento afín. Shi y Tomasi primero comparan parches en tramas vecinas utilizando un modelo traslacional y luego usan las estimaciones de ubicación producidas por este paso para inicializar un registro afín entre el parche en el marco actual y el marco base donde se detectó por primera vez una entidad. En su sistema, las entidades solo se detectan con poca frecuencia, es decir, solo en regiones donde el seguimiento ha fallado. En el caso habitual, se busca en un área alrededor de la ubicación predicha actual de la entidad con un algoritmo de registro incremental. El rastreador resultante a menudo se llama el rastreador Kanade-Lucas-Tomasi (KLT).

Desde su trabajo original en el seguimiento de características, el enfoque de Shi y Tomasi ha generado una serie de interesantes documentos y aplicaciones de seguimiento.



4.- Seguimiento de entidades mediante un modelo de movimiento afín [2].

Detección de bordes

Mientras que los puntos de interés son útiles para encontrar ubicaciones de imagen que se pueden emparejar con precisión en 2D, los puntos de borde son mucho más abundantes y a menudo llevan asociaciones semánticas importantes. Por ejemplo, los límites de los objetos, que también corresponden a eventos de oclusión en 3D, normalmente se delinean mediante contornos visibles. Otros tipos de aristas corresponden a límites de sombra o

aristas de pliegue, donde la orientación de la superficie cambia rápidamente. Los puntos de borde aislados también se pueden agrupar en curvas o contornos más largos, así como en segmentos de línea recta.

Cualitativamente, los bordes se producen en los límites entre regiones de diferente color, intensidad o textura. Desafortunadamente, segmentar una imagen en regiones coherentes es una tarea difícil. A menudo, es preferible detectar bordes utilizando solo información puramente local. En tales condiciones, un enfoque razonable es definir un borde como una ubicación de rápida variación de intensidad. Piense en una imagen como un campo de altura. En dicha superficie, los bordes se producen en ubicaciones de pendientes pronunciadas, o equivalentemente, en regiones de líneas de contorno estrechamente empaquetadas (en un mapa topográfico).

Una forma matemática de definir la pendiente y la dirección de una superficie es a través de su gradiente:

$$J(x) = \nabla I(x) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)(x)$$

El vector de gradiente local J apunta en la dirección del ascenso más pronunciado en la función de intensidad. Su magnitud es una indicación de la pendiente o fuerza de la variación, mientras que su orientación apunta en una dirección perpendicular al contorno local. Desafortunadamente, tomar derivados de imágenes acentúa las altas frecuencias y, por lo tanto, amplifica el ruido, ya que la proporción de ruido a la señal es mayor a altas frecuencias. Por lo tanto, es prudente suavizar la imagen con un filtro de paso bajo antes de calcular el degradado. Debido a que nos gustaría que la respuesta de nuestro detector de bordes fuera independiente de la orientación, es deseable un filtro de suavizado circularmente simétrico. El gaussiano es el único filtro simétrico circular separable y, por lo tanto, se utiliza en la mayoría de los algoritmos de detección de bordes. Dado que la diferenciación es una operación lineal, conmuta con otras operaciones de filtrado lineal. Por lo tanto, el degradado de la imagen suavizada se puede escribir como:

$$J_\sigma(x) = \nabla[G_\sigma(x) * I(x)] = [\nabla \mathrm{G}_\sigma](\mathrm{x}) * \mathrm{I}(\mathrm{x})$$

Es decir, podemos transformar la imagen con las derivadas horizontales y verticales de la función del núcleo gaussiano.

$$\nabla G_\sigma(x) = \left(\frac{\partial G_\sigma}{\partial x}, \frac{\partial G_\sigma}{\partial y}\right)(x) = [-x - y]\frac{1}{\sigma^3}\exp(-\frac{x^2 + y^2}{2\sigma^2})$$

Este es el mismo cálculo que realiza el filtro orientable de primer orden de Freeman y Adelson.

Para muchas aplicaciones, sin embargo, deseamos adelgazar una imagen de degradado continuo para devolver solo bordes aislados, es decir, como píxeles individuales en ubicaciones discretas a lo largo de los contornos de borde. Esto se puede lograr buscando máximos en la fuerza del borde (magnitud del gradiente) en una dirección perpendicular a la orientación del borde, es decir, a lo largo de la dirección del gradiente.

Encontrar este máximo corresponde a tomar una derivada direccional del campo de fuerza en la dirección del gradiente y luego buscar cruces de cero. La derivada direccional deseada es equivalente al producto escalar entre un segundo operador de gradiente y los resultados del primero.

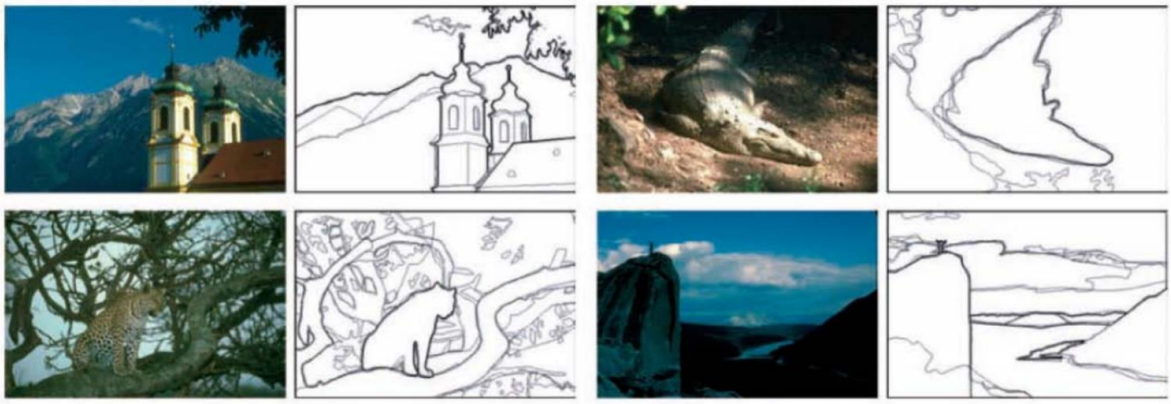
$$S_\sigma(x) = \nabla J_\sigma(x) = [\nabla^2 G_\sigma](x) * I(x)$$

El producto escalar del operador de gradiente con el gradiente se denomina laplaciano. Por lo tanto, el núcleo de convolución se llama el núcleo laplaciano Gaussiano (LoG). Este núcleo se puede dividir en dos partes separables, lo que permite una implementación mucho más eficiente utilizando el filtrado separable. En la práctica, es bastante común reemplazar el laplaciano de la convolución gaussiana con una diferencia de cálculo gaussiano (DoG), ya que las formas del núcleo son cualitativamente similares. Esto es especialmente conveniente si ya se ha calculado una "pirámide laplaciana".

De hecho, no es estrictamente necesario tomar diferencias entre los niveles adyacentes al calcular el campo de borde. Piense en lo que representa un cruce cero en una diferencia "generalizada" de la imagen gaussiano. El gaussiano más fino (núcleo más pequeño) es una versión reducida en ruido de la imagen original. El gaussiano más grueso (núcleo más grande) es una estimación de la intensidad promedio en una región más grande. Por lo tanto, cada vez que la imagen DoG cambia de signo, esto corresponde a la imagen (ligeramente borrosa) que va de relativamente más oscura a relativamente más clara, en comparación con la intensidad promedio en ese vecindario.

Se puede obtener una representación de bordes alternativa vinculando bordes adyacentes en la cuadrícula dual para formar bordes que viven dentro de cada cuadrado formado por cuatro píxeles adyacentes en la cuadrícula de píxeles original. La ventaja (potencial) de esta representación es que los bordes ahora viven en una cuadrícula desplazada por medio píxel de la cuadrícula de píxeles original y, por lo tanto, son más fáciles de almacenar y acceder. Como antes, las orientaciones y fortalezas de los bordes se pueden calcular interpolando el campo de degradado o estimando estos valores a partir de la diferencia de la imagen gaussiana. En aplicaciones en las que la precisión de la orientación del borde es más importante, se pueden utilizar filtros orientables de orden superior. Estos filtros son más selectivos para bordes más alargados y también tienen la posibilidad de modelar

mejor las intersecciones de curvas porque pueden representar múltiples orientaciones en el mismo píxel. Su desventaja es que son más costosos de calcular y la derivada direccional de la fuerza del borde no tiene una solución de forma cerrada simple [2].



5.- Detección de límites [2].

Selección de escala y estimación de desenfoque

Como mencionamos antes, la derivada laplaciana y la diferencia de los filtros gaussianos requieren la selección de un parámetro de escala espacial σ. Si sólo estamos interesados en detectar bordes afilados, el ancho del filtro se puede determinar a partir de las características de ruido de la imagen. Sin embargo, si queremos detectar bordes que ocurren a diferentes resoluciones, puede ser necesario un enfoque de espacio de escala que detecte y luego seleccione bordes a diferentes escalas. Elder y Zucker (1998) presentan un enfoque basado en principios para resolver este problema. Dado un nivel de ruido de imagen conocido, su técnica calcula, para cada píxel, la escala mínima a la que se puede detectar un borde de forma fiable. Su enfoque primero calcula gradientes densamente sobre una imagen seleccionando entre las estimaciones de gradiente calculadas a diferentes escalas, en función de sus magnitudes de gradiente. A continuación, realiza una estimación similar de la escala mínima para las segundas derivadas dirigidas y utiliza cruces cero de esta última cantidad para seleccionar robustamente las aristas. Como paso final opcional, el ancho de desenfoque de cada borde se puede calcular a partir de la distancia entre el extremo en la segunda respuesta derivada menos el ancho del filtro gaussiano [2].

Detección de bordes de color

Aunque la mayoría de las técnicas de detección de bordes se han desarrollado para imágenes en escala de grises, las imágenes en color pueden proporcionar información adicional. Por ejemplo, los bordes notables entre colores isoluminantes (colores que tienen la misma luminancia) son señales útiles, pero los operadores de bordes en escala de grises no pueden detectar. Un enfoque simple es combinar las salidas de los detectores

de escala de grises que se ejecutan en cada banda de color por separado. Sin embargo, se debe tener cierto cuidado. Por ejemplo, si simplemente sumamos los degradados en cada una de las bandas de color, los degradados firmados pueden cancelarse entre sí. También podríamos detectar bordes de forma independiente en cada banda y luego tomar la unión de estos, pero esto podría conducir a bordes engrosados o duplicados que son difíciles de vincular.

Un mejor enfoque es calcular la energía orientada en cada banda, por ejemplo, utilizando un filtro orientable de segundo orden, y luego resumir las energías ponderadas por orientación y encontrar su mejor orientación conjunta.

Desafortunadamente, la derivada direccional de esta energía puede no tener una solución de forma cerrada (como en el caso de los filtros orientables de primer orden firmados), por lo que no se puede usar una estrategia simple basada en el cruce cero. Sin embargo, la técnica descrita por Elder y Zucker se puede utilizar para calcular estos cruces cero numéricamente en su lugar. Un enfoque alternativo es estimar las estadísticas de color locales en regiones alrededor de cada píxel. Esto tiene la ventaja de que se pueden utilizar técnicas más sofisticadas (por ejemplo, histogramas de color 3D) para comparar estadísticas regionales y que también se pueden considerar medidas adicionales, como la textura [2].

Combinación de señales de entidad de arista

Si el objetivo de la detección de bordes es igualar el rendimiento de detección de límites humanos, en lugar de simplemente encontrar características estables para la coincidencia, se pueden construir detectores aún mejores combinando múltiples señales de bajo nivel como el brillo, el color y la textura. Martin, Fowlkes y Malik (2004) describen un sistema que combina bordes de brillo, color y textura para producir un rendimiento de vanguardia en una base de datos de imágenes en color natural segmentadas a mano. En primer lugar, construyen y entrenan 8 detectores de medio disco orientados separados para medir diferencias significativas en brillo (luminancia), color (canales a* y b*, respuestas sumadas) y textura. Algunas de las respuestas se afilan utilizando una técnica de supresión suave no máxima. Finalmente, las salidas de los tres detectores se combinan utilizando una variedad de técnicas de aprendizaje automático, a partir de las cuales se encuentra que la regresión logística tiene el mejor equilibrio entre velocidad, espacio y precisión. Se muestra que el sistema resultante supera a las técnicas desarrolladas anteriormente. Maire, Arbelaez, Fowlkes (2008) mejoran estos resultados combinando el detector basado en la apariencia local con un detector espectral (basado en la segmentación) [2].

Vinculación de bordes

Si bien los bordes aislados pueden ser útiles para una variedad de aplicaciones, como la detección de líneas y la coincidencia estéreo dispersa, se vuelven aún más útiles cuando se vinculan a contornos continuos. Si los bordes se han detectado utilizando cruces cero de alguna función, vincularlos es sencillo, ya que los bordes adyacentes comparten extremos comunes. Vincular los bordes en cadenas implica recoger un borde desvinculado y seguir a sus vecinos en ambas direcciones. Se puede utilizar una lista ordenada de bordes (ordenados primero por coordenadas x y, a continuación, por coordenadas y) o una matriz 2D para acelerar la búsqueda de vecinos. Si los bordes no se detectaron utilizando cruces cero, encontrar la continuación de un borde puede ser complicado. En este caso, la comparación de la orientación de los bordes adyacentes se puede utilizar para la desambiguación. Las ideas de la computación de componentes conectados también se pueden utilizar a veces para hacer que el proceso de vinculación de bordes sea aún más rápido. Una vez que los bordes se han ligado en cadenas, podemos aplicar un umbral opcional con histéresis para quitar los segmentos de contorno de baja resistencia.

La idea básica de la histéresis es establecer dos umbrales diferentes y permitir que una curva que se rastrea por encima del umbral más alto baje en fuerza hasta el umbral más bajo. Las listas de bordes vinculadas se pueden codificar de forma más compacta utilizando una variedad de representaciones alternativas. Un código de cadena codifica una lista de puntos conectados que se encuentran en una cuadrícula N8 utilizando un código de tres bits correspondiente a las ocho direcciones cardinales (N, NE, E, SE, S, SW, W, NW) entre un punto y su sucesor. Si bien esta representación es más compacta que la lista de bordes original (especialmente si se utiliza la codificación predictiva de longitud variable), no es muy adecuada para su posterior procesamiento. Una representación más útil es la parametrización de la longitud del arco de un contorno, x(s), donde s denota la longitud del arco a lo largo de una curva.

La ventaja de la parametrización de longitud de arco es que hace que las operaciones de coincidencia y procesamiento (por ejemplo, suavizado) sean mucho más fáciles. Si las curvas originales son las mismas (hasta una escala y rotación desconocidas), las transformadas de Fourier resultantes deben diferir solo por un cambio de escala en la magnitud más un desplazamiento de fase complejo constante, debido a la rotación, y un desplazamiento de fase lineal en el dominio, debido a diferentes puntos de partida para s. La parametrización de longitud de arco también se puede utilizar para suavizar curvas con el fin de eliminar el ruido de digitalización. Sin embargo, si solo aplicamos un filtro de suavizado regular, la curva tiende a encogerse sobre sí misma [2].

Aplicación: Edición y mejora de bordes

Aunque los bordes pueden servir como componentes para el reconocimiento de objetos o entidades para la coincidencia, también se pueden utilizar directamente para la edición de imágenes. De hecho, si la magnitud del borde y la estimación de desenfoque se mantienen junto con cada borde, se puede reconstruir una imagen visualmente similar a partir de esta información. Basándose en este principio, Elder y Goldberg (2001) proponen un sistema para la "edición de imágenes en el dominio del contorno". Su sistema permite a los usuarios eliminar selectivamente los bordes correspondientes a características no deseadas, como especularidades, sombras o elementos visuales que distraen.

Otra aplicación potencial es realzar los bordes perceptivamente salientes mientras se simplifica la imagen subyacente para producir una imagen estilizada de dibujos animados o "pluma y tinta" [2].

Detección de líneas

Mientras que los bordes y las curvas generales son adecuados para describir los contornos de los objetos naturales, el mundo hecho por el hombre está lleno de líneas rectas. La detección y coincidencia de estas líneas puede ser útil en una variedad de aplicaciones, incluido el modelado arquitectónico, la estimación de posturas en entornos urbanos y el análisis de diseños de documentos impresos [2].

Aproximación sucesiva

Describir una curva como una serie de ubicaciones 2D $x_i = x(s_i)$ proporciona una representación general adecuada para la coincidencia y el procesamiento posterior. En muchas aplicaciones, sin embargo, es preferible aproximar dicha curva con una representación más simple, por ejemplo, como una polilínea lineal por partes o como una curva.

Muchas técnicas se han desarrollado a lo largo de los años para realizar esta aproximación, que también se conoce como simplificación lineal. Uno de los más antiguos, y más simples, es el propuesto por Ramer y Douglas y Peucker (1973), que subdividen recursivamente la curva en el punto más alejado de la línea que une los dos extremos (o la aproximación actual de polilínea gruesa). Una vez calculada la simplificación de líneas, se puede utilizar para aproximar la curva original. Si se desea una representación o visualización más suave, se pueden utilizar splines o curvas de aproximación o interpolación [2].

Transformadas de Hough

Mientras que la aproximación de curvas con polilíneas a menudo puede conducir a una extracción de línea exitosa, las líneas en el mundo real a veces se dividen en componentes desconectados o se componen de muchos segmentos de línea colineales. En muchos casos, es deseable agrupar tales segmentos colineales en líneas extendidas. En una etapa de

procesamiento adicional, podemos agrupar dichas líneas en colecciones con puntos de fuga comunes. La transformada de Hough, llamada así por su inventor original (Hough 1962), es una técnica bien conocida para hacer que los bordes "voten" por ubicaciones de líneas plausibles. En su formulación original, cada punto de borde vota por todas las líneas posibles que pasan a través de él, y las líneas correspondientes a altos valores de acumulador o bin se examinan para los ajustes de línea potenciales. A menos que los puntos en una línea sean verdaderamente punteados, un mejor enfoque es usar la información de orientación local en cada borde para votar por una sola celda de acumulador. Una estrategia híbrida, donde cada borde vota por un número de posibles pares de orientación o ubicación centrados en la orientación de la estimación, puede ser deseable en algunos casos. Antes de que podamos votar por hipótesis de línea, primero debemos elegir una representación adecuada. Dado que las líneas se componen de segmentos de borde, adoptamos la convención de que la línea normal $\hat{n}$ apunta en la misma dirección (es decir, tiene el mismo signo) que el degradado de imagen $J(x) = \nabla I(x)$. Para obtener una representación mínima de dos parámetros para las líneas, convertimos el vector normal en un ángulo:

$$\theta = \tan^{-1}\frac{n_y}{n_x}$$

El rango de valores posibles (θ, d) es $[-180°, 180°] \times [-\sqrt{2}, \sqrt{2}]$, suponiendo que estamos utilizando coordenadas de píxeles normalizadas que se encuentran en [–1, 1]. El número de bins que se van a utilizar a lo largo de cada eje depende de la precisión de la estimación de posición y orientación disponible en cada borde y la densidad de línea esperada, y se establece mejor experimentalmente con algunas ejecuciones de prueba en imágenes de muestra.

Hay muchos detalles para que la transformación de Hough funcione bien, pero es mejor resolverlos escribiendo una implementación y probándola en datos de ejemplo.

Una alternativa a la representación polar 2D (θ, d) para las líneas es usar la ecuación de línea 3D $m = (\hat{n}, d)$ completa, proyectada sobre la esfera unitaria. Mientras que la esfera se puede parametrizar usando coordenadas esféricas:

$$\hat{m} = (\cos\theta\cos\emptyset, \sin\theta\cos\emptyset, \sin\emptyset)$$

Esto no muestra uniformemente la esfera y todavía requiere el uso de trigonometría. Se puede obtener una representación alternativa utilizando un mapa cúbico, es decir, proyectando m sobre la cara de un cubo unitario. Para calcular la coordenada del mapa de cubo de un vector 3D m, primero busque el componente más grande (valor absoluto) de m, es decir, $m = \pm max(|nx|, |ny|, |d|)$ y utilícelo para seleccionar una de las seis

caras del cubo. Divida las dos coordenadas restantes por m y utilícelas como índices en la cara del cubo. Si bien esto evita el uso de trigonometría, requiere cierta lógica de decisión. Una ventaja de usar el mapa de cubo es que todas las líneas que pasan a través de un punto corresponden a segmentos de línea en las caras del cubo, lo que es útil si se utiliza la variante original de la transformada de Hough [2].

Detección de línea basada en RANSAC.

Otra alternativa a la transformada de Hough es el algoritmo Random Sample Consensus (RANSAC). En resumen, RANSAC elige aleatoriamente pares de bordes para formar una hipótesis de línea y luego prueba cuántos otros bordes caen sobre esta línea. (Si las orientaciones de borde son lo suficientemente precisas, un solo borde puede producir esta hipótesis). A continuación, se seleccionan líneas con un número suficientemente grande de bordes coincidentes como los segmentos de línea deseados. Una ventaja de RANSAC es que no se necesita ninguna matriz de acumuladores y, por lo tanto, el algoritmo puede ser más eficiente en el espacio y potencialmente menos propenso a la elección del tamaño del bin. La desventaja es que puede ser necesario generar y probar muchas más hipótesis que las obtenidas al encontrar picos en la matriz de acumuladores. En general, no hay un consenso claro sobre qué técnica de estimación de línea funciona mejor [2].

Puntos de fuga

En muchas escenas, las líneas estructuralmente importantes tienen el mismo punto de fuga porque son paralelas en 3D. Ejemplos de tales líneas son los bordes horizontales y verticales de los edificios, los pasos de cebra, las vías del tren, los bordes de los muebles, como mesas y aparadores, y por supuesto, el patrón de calibración ubicuo.

Encontrar los puntos de fuga comunes a estos conjuntos de líneas puede ayudar a refinar su posición en la imagen y, en ciertos casos, ayudar a determinar la orientación intrínseca y extrínseca de la cámara. A lo largo de los años, se ha desarrollado un gran número de técnicas para encontrar puntos de fuga. La primera etapa en el algoritmo de detección de punto de fuga, se utiliza una transformación de Hough para acumular votos para los candidatos probables de punto de fuga. Al igual que con el ajuste de línea, un enfoque posible es hacer que cada línea vote para todas las direcciones de punto de fuga posibles, ya sea usando un mapa de cubo o una esfera gaussiana, utilizando opcionalmente el conocimiento sobre la incertidumbre en la ubicación del punto de fuga para realizar un voto ponderado. Sean $\hat{m}_i$ y $\hat{m}_j$ las ecuaciones de línea (norma unitaria) para un par de segmentos de línea y l_i y l_j sean sus longitudes de segmento correspondientes. La ubicación de la hipótesis del punto de fuga correspondiente se puede calcular como

$$v_{ij} = \widehat{m_i} * \widehat{m_j}$$

y el peso correspondiente establecido en

$$w_{ij} = \|v_{ij}\| l_i l_j$$

Esto tiene el efecto deseable de ponderar hacia abajo los segmentos de línea (casi) colineal y los segmentos de línea corta. El propio espacio de Hough se puede representar utilizando coordenadas esféricas o como un mapa de cubo. Una vez que se ha poblado el espacio del acumulador de Hough, los picos se pueden detectar de una manera similar a la discutida previamente para la detección de línea.

Considere la relación entre los dos extremos del segmento de línea $\{p_{i0}, p_{i1}\}$ y el punto de fuga v. El área A del triángulo dada por estos tres puntos, que es la magnitud de su triple producto es proporcional a la distancia perpendicular d_1 entre cada extremo y la línea a través de v y el otro extremo, así como la distancia entre p_{i0} y v. Suponiendo que la precisión de un segmento de línea ajustado es proporcional a su precisión de punto final, esto sirve por lo tanto como una métrica óptima para saber qué tan bien se ajusta un punto de fuga a un conjunto de líneas extraídas. Por lo tanto, una estimación de mínimos cuadrados para el punto de fuga puede escribirse como:

$$\varepsilon = \sum_i \rho(A_i) = v^T \left(\sum_i w_i(A_i) m_i m_i^T \right) v = v^T M v$$

donde $m_i = p_{i0} \times p_{i1}$ es la ecuación de la línea de segmento ponderada por su longitud l_i, y $w_i = \rho' \frac{Ai}{Ai}$ es la influencia de cada medición reponderada en el error final. El valor final deseado para v se calcula como el menor vector propio de M.

Mientras que la técnica descrita anteriormente procede en dos etapas discretas, se pueden obtener mejores resultados alternando entre la asignación de líneas a los puntos de fuga y el reacondicionamiento de las ubicaciones de los puntos de fuga. Los resultados de detectar puntos de fuga individuales también se pueden hacer más robustos mediante la búsqueda simultánea de pares o trillizos de puntos de fuga mutuamente ortogonales [2].

6.- Puntos de fuga del mundo real: arquitectura, muebles y patrones de calibración.

Aplicación: Detección de rectángulos

Una vez que se han detectado conjuntos de puntos de fuga mutuamente ortogonales, ahora es posible buscar estructuras rectangulares 3D en una imagen. Durante la última

década, se han desarrollado una variedad de técnicas para encontrar tales rectángulos, centrados principalmente en escenas arquitectónicas. Después de detectar direcciones de fuga ortogonales, Kosecka y Zhang (2005) refinan las ecuaciones de línea ajustadas, buscan esquinas cerca de intersecciones de líneas y luego verifican hipótesis de rectángulo rectificando los parches correspondientes y buscando una preponderancia de bordes horizontales y verticales. En el trabajo de seguimiento, Micusık, Wildenauer y Kosecka (2008) usan un campo aleatorio de Markov (MRF) para desambiguar entre hipótesis de rectángulo potencialmente superpuestas. También utilizan un algoritmo de barrido de planos para hacer coincidir rectángulos entre diferentes vistas. Un enfoque diferente es propuesto por Han y Zhu, que utilizan una gramática de las formas de rectángulo potenciales y las estructuras de anidamiento (entre rectángulos y puntos de fuga) para inferir la asignación más probable de segmentos de línea a rectángulos [2].

Reconocimiento

Todo sistema de visión por computador, al cual se conoce como visión artificial, en la actualidad se emplean cada vez más en procesos de reconocimiento automático ya sea en industria agroalimentaria o agroexportadora, estas técnicas forman parte de las herramientas indispensables para dichas empresas para mantenerse en un nivel competitivo (Granados Montelongo, 2000). Estas técnicas se emplean en la clasificación y supervisión de calidad de productos, ya que estos sistemas ofrecen el potencial necesario para automatizar las prácticas manuales de selección, a la vez estandarizan las técnicas reduciendo las costosas tareas humanas de clasificación e inspección, a pesar de conocer las limitaciones humanas y la susceptibilidad a cometer errores. Así mismo, el desarrollo de tecnologías digitales aplicadas en la agricultura otorga nuevas posibilidades para automatizar algunos procesos tecnológicos (Rodriguez Pérez, 2015).

La Visión Artificial gira esencialmente en torno al reconocimiento de patrones o características específicas en las imágenes. Para esto necesitamos de algoritmos que analicen todas las condiciones y patrones que hacen que en una imagen pueda identificarse la presencia de un objeto determinado, lo que se conoce como entrenamiento.

Los algoritmos son un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permiten, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.

El entrenamiento se puede lograr mediante la repetición. Las computadoras deben recibir tantas imágenes identificadas o etiquetadas como sea posible. Por ejemplo, si quisiera enseñar a una computadora a identificar ovejas, se le mostraría numerosas imágenes con ovejas etiquetadas. Para etiquetar la oveja la computadora reconocería qué píxeles

específicos tienen patrones o características de una oveja y luego asociaría esa estructura de píxeles con ovejas.

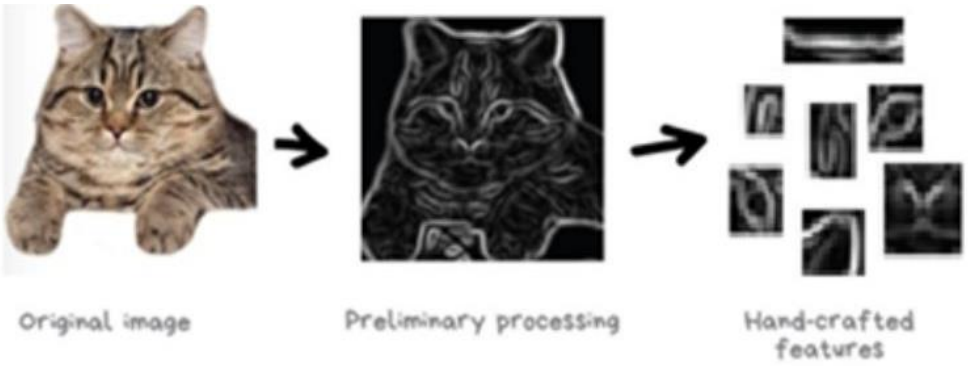
Este caso se conoce como Aprendizaje Supervisado, dentro del área del Aprendizaje Automático.

Ahora, estos patrones o características pueden ser millones, y para ello nos ayudaremos de la Inteligencia Artificial, en este ejemplo, de las redes neuronales, que es uno de los tipos de aprendizaje automático (Machine Learning) más populares de la inteligencia artificial [3].

Identificación de patrones y redes neuronales

Hasta hace algunos años se optaba por etiquetar manualmente las imágenes para enseñarle a la máquina dónde estaban las orejas o la cola de un gato. Esto se conoció como "características de fabricación artesanal" y solía ser muy utilizado, pero resultaba muy laborioso y puramente "comparativo" y no de aprendizaje real de las características de las imágenes.

En la siguiente imagen vemos la figura de un gato, luego la conversión a grises, que nos permitirá identificar mejor las líneas principales en grupos de pixeles, y posteriormente la selección de partes del gato (orejas, boca, nariz, etc.). Con esa información nuestra red neuronal debería ser capaz de identificar un gato en la imagen [3].



7.- Identificación de patrone [3]s.

Pero, qué pasa si un gato tiene las orejas hacia abajo o se aleja de la cámara: la red neuronal no logrará identificarlo.

Por eso tenemos que intentar que la máquina aprenda las características por sí misma, basándose en líneas básicas. Un método puede ser dividir la imagen completa en bloques de 8 × 8 pixels y asignarle a cada uno un tipo de línea dominante, ya sea horizontal [-], vertical [|] o diagonal [/].

La salida de nuestro proceso, serán varias tablas formadas por "palitos" que son, de hecho, las características más simples que representan los bordes de los objetos en la imagen. Son solamente imágenes, pero construidas de palitos. Así que, una vez más, podemos tomar un bloque de 8×8 y ver si coinciden. Y una y otra vez.

Esta operación se llama "convolución", y este nombre le dio la denominación al algoritmo. La convolución se puede representar como una capa de una red neuronal, porque cada neurona puede actuar como cualquier función.

Cuando alimentamos nuestra red neuronal con muchas fotos de gatos, asigna automáticamente pesos (importancia) más grandes a las combinaciones de palitos que ve con más frecuencia. No importa si se trata de una línea recta de la espalda de un gato o un objeto geométricamente complicado como la cara de un gato.

Para obtener nuestra salida pondríamos una neurona simple que buscará las combinaciones más comunes y, en base a esto, podríamos diferenciar a los gatos de los perros.

Lo interesante de esta idea de "palitos" es que tenemos una red neuronal que busca las características más distintivas de los objetos por sí misma. No necesitamos recogerlos manualmente. Podemos alimentar cualquier cantidad de imágenes de cualquier objeto con solo buscar miles de millones de imágenes y nuestra red creará mapas de características a partir de palitos y aprenderá a diferenciar cualquier objeto por sí solo.

Algo similar, pero con otras técnicas, lo podemos ver en el área de reconocimiento facial, una de las aplicaciones de la Visión Artificial. En este caso, en lugar de palitos se buscan puntitos como referencias faciales, pero finalmente el proceso de aprendizaje es similar a la del gato [3].

Aplicaciones de la Visión Artificial

La Visión Artificial es un área con más de 40 años de investigación, por lo que ya contamos con varias aplicaciones de las técnicas desarrolladas. Estas aplicaciones incluyen:

- Reconocimiento óptico de caracteres (OCR): Que consiste en la identificación automática a partir de una imagen de símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos.
- Inspección robotizada: La inspección rápida de las piezas para garantizar la calidad de los componentes de fabricación utilizando una visión estéreo con iluminación especializada.
- Construcción de modelos 3D: La construcción automatizada de modelos 3D a partir de fotografías.

- Imágenes médicas: Tecnología utilizada para tomar radiografías y las técnicas para detectar tumores malignos y anomalías en las mismas.
- Seguridad automotriz: Ayudar a detectar obstáculos mediante un sistema de conducción asistida utilizando diferentes cámaras.
- Captura de movimiento: Utilizar marcadores retro-reflexivos vistos desde múltiples cámaras u otras técnicas para la captura de movimientos de los actores para utilizar en animación por computadora.
- Vigilancia: Monitoreo de intrusos, análisis del tráfico vial, y monitoreo de piscinas para víctimas de ahogamiento.
- Reconocimiento de huellas dactilares y biometría: Para la identificación automática de accesos y también utilizada para aplicaciones forenses.
- Detección de caras: Utilizado para mejorar el foco de las cámaras y para hacer una búsqueda más relevante de personas en imágenes [3].

Conclusión

A través de este informe pude conocer como la computadora puede llegar a reconocer contornos, líneas, bordes, etc. Y complementando con lo visto anteriormente que fue el tratamiento de imágenes, el poder identificar ya sea formas, objetos, entre otras cosas, es una parte fundamental de la visión por computadora ya que es a través de estos métodos de reconocimiento que podemos interactuar con el entorno de manera eficiente. Por ejemplo, poder identificar un objeto de interés para poder manipularlo por medio de un brazo robótico, y de esta manera poder interactuar.

Las aplicaciones de la visión artificial son muy bastas y faltan varias cosas por descubrir, por lo que es importante conocer todo el proceso que lleva la visión por computadora, y de esta manera poder hacer uso de esta para resolver problemáticas de manera correcta.

Bibliografía

1. Ramírez Manzanares, A. (2021). Búsqueda, detección y conteo de objetos [PDF]. Consultado de https://www.cimat.mx/~alram/VC/ramirez_segmObjDetecc.pdf
2. SZELISKI, R. (2020). COMPUTER VISION (pp. 207-257). SPRINGER NATURE.
3. La Visión Artificial y el reconocimiento de imágenes: procesamiento automatizado. (2021). Consultado el 5 de Julio del 2021, de https://santanderglobaltech.com/vision-artificial-reconocimiento-imagenes-procesamiento-automatizado/