Informe de proyecto:
Implementación del algoritmo de rectificación de Loop & Zhang

Antonio Álvarez Caballero
Alejandro García Montoro
analca3@correo.ugr.es
agarciamontoro@correo.ugr.es

1. Descripción del problema

La rectificación de imágenes es el proceso de aplicar sendas homografías a un par de imágenes cuya geometría epipolar conocemos, de modo que las líneas epipolares queden horizontales, paralelas entre sí y con la misma coordenada vertical.

La motivación de la rectificación es simple: al tener las líneas epipolares horizontales y con la misma coordenada vertical, buscar correspondencias entre las dos imágenes es mucho más fácil y eficiente; como sabemos que las correspondencias se encuentran en las mismas líneas epipolares, si suponemos que las imágenes están rectificadas basta buscar en la correspondiente fila de píxeles de la otra imagen, donde se encuentra la línea epipolar del punto que estemos considerando. Si las imágenes no estuvieran rectificadas, el tiempo de ejecución aumentaría considerablemente; la búsqueda en líneas inclinadas —no paralelas con ninguno de los dos ejes— es más compleja computacionalmente.

Así, es claro que tener un par de imágenes rectificadas mejora la eficiencia de muchos de los algoritmos que precisan conocer la geometría epipolar y tienen que buscar correspondencias entre ambas imágenes. Por ejemplo, de esta situación se consigue la mejora computacional de los algoritmos que calculan mapas de disparidad, donde hay que medir el desplazamiento relativo entre los píxeles de una y otra imagen. Esta distancia relativa es lo que conocemos por *disparidad*, y con esta información se puede reconstruir la profundidad incluso sin noción alguna de cámaras; basta un par de imágenes de una misma escena estática.

Este proyecto implementa el método de rectificación propuesto por Charles Loop y Zhengyou Zhang en [**?**].

Su algoritmo es totalmente geométrico y no precisa conocimiento de las cámaras. Se consigue descomponiendo la homografía que se desea calcular para cada imagen en una proyección, una transformación euclídea y una cizalla, con especial cuidado en minimizar la distorsión del resultado con respecto a las imágenes originales.

Es de hecho este último criterio el que hace del algoritmo una de las mejores soluciones actuales, ya que se consigue una rectificación que mantiene casi todas las propiedades de las imágenes originales.

2. Enfoque de la implementación y detalles de eficiencia

La implementación del algoritmo se basa en la descomposición de la homografía H que se quiere obtener en las siguientes transformaciones:

- H_p: transformación proyectiva.
- H_r: transformación de semejanza.
- H_s: composición de una transformación de cizalla con un escalado y una traslación.

2.1. Descomposición

Siguiendo la notación de [?], sea H la homografía siguiente:

$$H = \begin{pmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & w_c \end{pmatrix}$$

Si descomponemos H en su parte proyectiva y afín —y esta a su vez en una parte de semejanza y otra de cizalla—, obtenemos que

$$H = H_s H_r H_p$$

donde

$$H_s = \begin{pmatrix} s_a & s_b & s_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; \; H_r = \begin{pmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{pmatrix}; \; H_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{pmatrix}$$

El cálculo de cada transformación se hace por separado, teniendo en cuenta las matrices calculadas anteriormente y criterios de minimización que se explicarán en cada subsección.

Antes de todo este cálculo, sin embargo, hace falta calcular la geometría epipolar del par de imágenes, que el algoritmo supone conocida. Este paso previo se ha implementado buscando las correspondencias con un detector `BRISK`, calculando la matriz fundamental con la llamada a la función `findFundamentalMat` de *OpenCV*—usando el algoritmo de los 8 puntos y `RANSAC`— y usando las llamadas a `computeCorrespondEpilines` de la misma librería.

2.2. Transformación proyectiva

La transformación proyectiva lleva los epipolos e y e' al infinito, de manera que tras aplicar H_p y H'_p, las líneas epipolares son ya paralelas.

El cálculo de w_a, w_b y w'_a, w'_b se basa en un criterio de minimización de la distorsión que puede resumirse en lo siguiente: buscamos unas transformaciones proyectivas H_p y H'_p que sean lo más cercanas que podamos a transformaciones afines.

La idea es que las líneas $w = [w_a, w_b, 1]$ y $w' = [w'_a, w'_b, 1]$ transformen los puntos de la manera más compensada posible; es decir, dado un punto $p_i = [p_{i,u}, p_{i,v}, 1]$, la transformación H_p lo transformará en el punto $[\frac{p_{i,u}}{\omega_i}, \frac{p_{i,v}}{\omega_i}, 1]$, donde los pesos ω_i pueden medir de alguna manera cuán proyectiva es nuestra transformación: si todos los pesos son idénticos, la transformación es afín.

Buscamos por tanto unas líneas w y w' que consigan que los pesos ω_i se parezcan todo lo posible.

Toda la discusión matemática se encuentra en [?], pero

2.3. Transformación de semejanza

Euclídea

2.4. Transformación de cizalla

3. Experimentos realizados

Lola es muy guapa y adorable. V impone.

4. Valoración de resultados

Todo se ve de lujo

5. Conclusiones

Este algoritmo es pro. Mejor que el de OpenCV.