

PERCEPTUAL KERNELS

NOMBRE: John Haiber Osorio Ríos. **CÓDIGO:** 1093212116

El código del proyecto con sus entradas se encuentra en <https://github.com/kala855/viscourse>, allí hay varias carpetas, este proyecto se encuentra en la carpeta **perceptualkernels**, allí se encuentra una carpeta de artículos donde se pueden ver algunos de los usados en este trabajo, una carpeta informe donde se encuentra el informe, también hay una carpeta de outputs que contiene los kernels de salida y las imágenes asociadas.

La implementación realizada buscó apoyarse en 3 artículos donde se habla de la temática de **Perceptual Kernels** [1], [2] y [3]. Hubo alguna información que se tomó de ellos, sin embargo hay contenido que no quedó muy claro, sin embargo a continuación se cuenta cómo se realizó la investigación.

Durante la construcción del algoritmo para realizar la tarea de perceptual kernels se utilizó una técnica conocida como **Triplet Matching** [1], ésta técnica plantea que para la decisión de la tripleta se tendrá una imagen de referencia y se le irá preguntando al usuario de un set de otras dos imágenes cual es la más parecida a ésta referencia, de este modo y según [1] al usuario se le presentarán un conjunto de opciones más claras, permitiéndole tener una decisión que es netamente binaria. Según la investigación realizada en [1] este tipo de técnica es mucho mas confiable que **Triplet Discrimination** y que **Spatial Arrangement**.

También existen varios métodos para realizar la asignación de los puntajes dentro del Perceptual Kernel tal y como se puede ver en [1] y [2], sin embargo el algoritmo construido utiliza una técnica simple que aún es susceptible de mejora. En este caso se generan N^3 (este valor podría cambiar) índices aleatorios que permiten decidir que conjunto de imágenes se van a comparar, todo esto siguiendo algunas recomendaciones dadas en [1], donde plantean que es muy importante que al usuario se le muestren los datos de forma aleatoria, e inclusive plantean que deberían evaluarse también casos sin sentido, como por ejemplo una comparación entre tres imágenes iguales. Una vez se generan los índices y el usuario realiza una elección lo que se hace es ir a la matriz perceptual tomando como índices los de la imagen seleccionada y la referencia y se acumulan en ésta.

No se evaluó por completo el kernel de 27x27 que combina forma, tamaño y color; sin embargo se realizaron un par de pruebas con imágenes de entrada reducidas, la primera de ellas que se puede ver en la **Figura 1**, nos muestra un perceptual kernel obtenido de 6 personas diferentes, una vez todas las personas llenaron su perceptual kernel se procedió a realizar un promedio entre todas y a graficar un kernel que condensa la información ingresada por todos.

En todas las figuras el color blanco muestra las imágenes más parecidas entre sí.

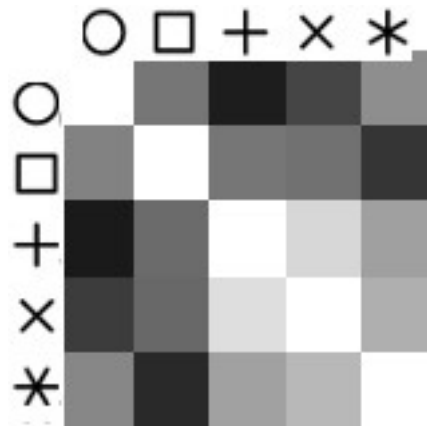


Figura 1. Promedio de 6 Personas de Triplet Matching de Forma.

En la **Figura2** se puede ver un kernel perceptual de prueba llenado por una sola persona que muestra la comparación de forma, tamaño y color.

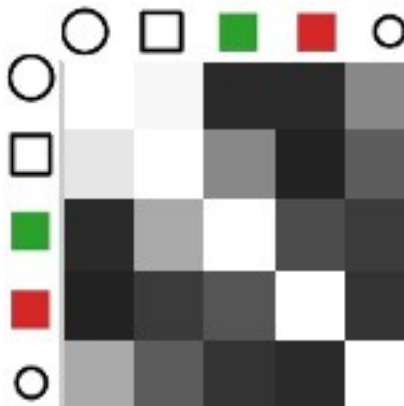


Figura 2. Combinacion de Color, Forma y Tamaño

El kernel en estos casos permitiría conocer del conjunto de imágenes de entrada, cuáles son las que se deberían utilizar si por ejemplo se quisieran mostrar datos en un gráfico que no tienen ninguna relación entre sí en el caso de los cuadros más oscuros, o podría ayudarnos a mostrar también en alguna gráfica datos que de algún modo tienen alguna relación en el caso de las imágenes que puedan ser tomadas como más parecidas que son representadas en el kernel como los cuadros más blancos.

BIBLIOGRAFÍA

- [1] Cagatay Demiralp, Michael Bernstein, Jeffrey Heer. Learning Perceptual Kernels for Visualization Design.
- [2] Sameer Agarwal, Josh Wills, Lawrence Cayton. Generalized Non-metric Multidimensional Scaling.
- [3] Josh Wills, Sameer Agarwal, David Kriegman. Toward a Perceptual Space for Gloss.