



GRADO EN MATEMÁTICA COMPUTACIONAL

ESTANCIA EN PRÁCTICAS Y PROYECTO FINAL DE GRADO

La interpolación aplicada al procesamiento de imágenes digitales

Autor:
Alba RONCERO GARCÍA

Tutor académico:
Alejandro MIRALLES MONTOLÍO

Fecha de lectura: __ de Octubre de 2019 Curso académico 2018/2019

Resumen

Durante este proyecto se tratan conocimientos generales del procesamiento de imágenes, y más concretamente cómo se aplica la interpolación en este proceso. También se explican en detalle los métodos de interpolación más relevantes en este campo y algunos conocimientos previos sobre las imágenes digitales que son necesarios para entender cómo se aplican los métodos de interpolación.

Cabe destacar que durante esta memoria también queda reflejado mi paso por la empresa Opentix.S.L en la cual realice prácticas extracurriculares y posteriormente fui contratada.

Palabras clave

Interpolación, imágenes digitales, MATLAB, procesamiento de imágenes, píxel

Abstract

During this project in general terms it is treated the knowledge of image processing and to be precise, how the interpolation takes part in this process. Moreover, the most relevant methods of this subject are well explained with full details and on top of that the previous knowledge or background about the digital image which are mandatory in order to understand how the interpolation methods are applied.

In the process of writing down this thesis (también sirve document) it is important to underline that I took my work placement in Opentix and then I was hired after the probation period.

Keywords

Interpolation, digital images, MATLAB, image processing, pixel

Agradecimientos

En primer lugar, como no, me gustaría agradecer a mi familia, en especial a mis abuelos Rosa y Curro y a mi padre, que siempre han estado ahí para apoyarme , durante mis altibajos y dudas, que han aguantado cada llamada y cada preocupación durante esta etapa. A mis compañeros de carrera Jessica que mas que una compañera la puedo considerar como mi hermana y Alejandro, con los que por suerte he compartido grandes momentos y espero poder seguir compartiendolos, ya que siguen en mi vida .

En segundo lugar, agradecer a todos los profesores que me han ayudado a llegar hasta aquí, a unos más, a otros menos, pero en especial a Juan José Font, Pablo Gregori, David Llorens y María José Aramburu. Destacar a Mercedes Segarra que me ha ayudado a encontrar mi camino. Esta memoria refleja el resumen de cinco años en el grado en Matemáticas Computacional en la Universidad Jaume I. Cinco años en los que he adquirido multitud de conocimientos en diversos ámbitos de la vida, no solo en la profesional sino también en la personal. Durante esta etapa he conocido a muchas personas, algunas que se han quedado en mi vida y otras que se han ido de ella, pero tanto las que se han quedado como las que se han ido me han dado experiencias de las que aprender, aquellas que me han ayudado y apoyado tanto en los buenos y malos momentos, a los que han intentado hacerme este camino más corto y ameno. Cinco años que me han servido para sentir que puedes llegar a donde quieres si te esfuerzas, que no hay limites. Es por ello que me gustaría agradecer a todas las personas que me han ayudado y apoyado durante este proceso y en este trabajo en particular. Finalmente, agradecer a quienes han hecho posible que salga adelante este proyecto. A mi supervisora de prácticas extracurriculares Adriana Venete y Opentix S.L, por brindarme esta oportunidad. A toda la gente de Opentix S.L, en especial al equipo de desarrollo quienes me han ayudado a iniciarme en el mundo laboral. Y, a mi tutor Alejandro Miralles, por su gran ayuda y colaboración.

Gracias a todos.

Alba Roncero Garcia.

Índice general

1. Introducción	13
1.1. Contexto y motivación del proyecto	13
1.2. Estructura de la memoria	14
2. Estancia en prácticas	15
2.1. Introducción	15
2.2. Proyecto formativo	16
2.3. Trabajo desarrollado	17
2.4. Tecnologías empleadas	18
3. La interpolación	19
3.1. Introducción	19
3.2. Interpolación polinómica	20
3.2.1. Método de interpolación de Lagrange	21
3.2.2. Método de Newton (Diferencias divididas)	24
3.2.3. Splines	26

3.3. Métodos de interpolación	30
3.3.1. Interpolación lineal	30
3.3.2. Interpolación Bilineal	31
3.3.3. Interpolación bicúbica	35
3.3.4. Interpolación del vecino más cercano	39
4. Procesamiento de Imágenes	43
4.1. Introducción	43
4.2. Historia del Procesamiento de Imágenes	44
4.3. Conceptos Previos	45
4.3.1. Imágenes digitales	45
4.3.2. Tipos de imágenes Digitales	46
4.3.3. Píxel	46
4.3.4. Tamaño	50
4.3.5. Resolución	51
4.4. Interpolación en el procesamiento de imágenes	51
4.5. Remuestreo de imágenes	51
4.5.1. Interpolación Bilineal	52
4.5.2. ¿Qué método es mejor?	55
5. Ejemplos de código para el procesamiento de imágenes digitales en MATLAB	57
5.1. Introducción	57
5.2. MATLAB	60

5.3. Interpolación lineal	61
5.4. Interpolación cúbica	61
5.5. Interpolación splines	62
6. Conclusiones	65
6.1. Ámbito formativo	65
6.2. Ámbito profesional	66
6.3. Ámbito personal	66
A. Anexo I	67
B. Anexo II	69
B.1. Demostración teorema fundamental del álgebra	69
B.2. Semejanza de triángulos	73

Índice de figuras

3.1. Splines de grado 0	27
3.2. Splines de grado 1	28
3.3. Interpolación bilineal.	32
3.4. Píxeles escala de grises	34
3.5. Interpolación del vecino más cercano	39
3.6. Búsqueda lineal	40
3.7. Diagrama de Voronoi	41
4.1. Imagen monocromática	47
4.2. Imagen con 16 colores	47
4.3. Imagen con 256 colores	48
4.4. Imagen con 6 millones de colores	48
4.5. Píxeles	49
4.6. Propiedades de una Imágen	51
4.7. Ejemplo interpolación bilineal remuestreo	52
4.8. Ejemplo interpolación bilineal remuestreo	53

4.9. Ejemplo interpolación bilineal remuestreo	53
4.10. Ejemplo interpolación bilineal remuestreo	55
5.1. Imagen Original	58
5.2. Imagen con mayor resolución	59
5.3. Imagen con menor resolución	59
5.4. Imagen con 10x zoom	60
5.5. Imagen con interpolación lineal	61
5.6. Imagen con interpolación cúbica	62
5.7. Imagen con interpolación de splines	63
B.1. Semejanza de triángulos.	73

Índice de cuadros

3.1. Datos ejemplo método de Lagrange	23
3.2. Valores splines cúbicos	28
3.3. Ejemplo Interpolación lineal	31

Capítulo 1

Introducción

Contents

1.1. Contexto y motivación del proyecto	13
1.2. Estructura de la memoria	14

1.1. Contexto y motivación del proyecto

Este proyecto abarca la realización del Trabajo de Fin de Grado (TFG) en el Grado de Matemáticas Computacional y tiene como objetivo el estudio de la interpolación aplicada al procesamiento de imágenes.

Durante la memoria se habla también de mi realización de prácticas en la empresa Open-tix.SL, aunque brevemente, ya que las prácticas han sido exendidas por un contrato de prácticas extracurriculares y más tarde laboral.

En cuanto a la memoria no corresponde con un proyecto relacionado con las prácticas sino con un trabajo bibliográfico sobre la aplicación de la interpolación en el procesamiento de imágenes digitales.

1.2. Estructura de la memoria

Este proyecto final de grado está compuesto en su totalidad por seis capítulos.

En el capítulo 2, se comenta de forma breve mi estancia en la empresa Opentix.S.L en la cual realicé prácticas extracurriculares y posteriormente fui contratada.

Para continuar, en el capítulo 3, hablaré sobre la interpolación y serán desarrollados los métodos de interpolación más relacionados con el procesamiento de imágenes digitales.

En el capítulo 4, veremos conceptos generales sobre el procesamiento de imágenes, en los que se incluyen conceptos previos sobre las imágenes digitales, necesarios para tener una idea general del procesamiento de imágenes digitales. También en este capítulo veremos el concepto de remuestreo y daremos un ejemplo de remuestreo de imágenes con la interpolación bilineal.

En el capítulo 5, veremos ejemplos de remuestreo de imágenes con tres métodos de interpolación diferentes: interpolación lineal, cubica y de splines con la ayuda del lenguaje de programación MATLAB.

Para finalizar, en el capítulo 6, daré unas conclusiones generales en cuanto al grado en Matemáticas computacional, en concreto como ha afectado en mi vida, tanto en el ámbito formativo y laboral como en el personal.

Capítulo 2

Estancia en prácticas

Contents

2.1. Introducción	15
2.2. Proyecto formativo	16
2.3. Trabajo desarrollado	17
2.4. Tecnologías empleadas	18

2.1. Introducción

Durante este capítulo describiré brevemente mi estancia en prácticas realizadas durante el año 2019 en la empresa Opentix.SL en la provincia de Castellón.

Opentix S.L [1], nació de la mano de Daniel Segarra en el año 2012, que tras estudiar en la UJI fundó su propia sociedad dentro del CEEI (Centro Europeo Empresa Innovadora) con el objetivo de dar soluciones tecnológicas como ERP y CMR a pequeñas y medianas empresas. Una de estas tecnologías empleadas ha sido Openbravo, que es la tecnologías con la que he trabajado dentro de la empresa, aunque también trabajan con otras tecnologías como Navision o Sage.

Desde su nacimiento en 2012, la empresa ha crecido significativamente. Teniendo un reconocimiento en la 'Gala Empresa del Año' como la mejor empresa joven de 2014 y 2015. Desde su primera oficina de Castellón, la empresa se ha extendido hasta Valencia, Madrid, Barcelona, Zaragoza y Sevilla. Desde el año 2017, Opentix pertenece a un grupo de empresas complementarias, en las cuales incluye Fadrell y Aitana, siendo esta última la empresa matriz. Estas ofrecen al cliente el software que más se ajusta a sus necesidades.

Aitana [2], es una empresa con más de 40 años de experiencia dedicada al sector tecnológico, que ofrece soluciones a los problemas de gestión de sus clientes o de aquellas empresas que necesiten mejorar y optimizar sus procesos de negocio.

Las soluciones tecnológicas que ofrece son ERP, CRM y Business Intelligence, sobre potentes plataformas como Sage y Microsoft Navision. También disponen de soluciones sectoriales y horizontales como la gestión de recursos humanos, gestión documental y retail que completan la optimización de los recursos de la empresa.

Fadrell GT (Grupo Tecnológico) [3], es una empresa dedicada al desarrollo, implantación y mantenimiento de tecnologías de la información y comunicación. Como consultores asesoran a los clientes para encontrar la solución que más rentabilidad suma a su negocio, especializándose en áreas IT como el cloud Computing, networking, centros de procesos de datos y soluciones para la industria 4.0.

Openbravo ERP es un software abierto en la nube especializado en soluciones para el sector minorista. La arquitectura en la que está especializada Openbravo se centra en dos modelos de desarrollo y está basado en Java. El primer modelo de desarrollo es la Ingeniería orientada a modelos, en la que los desarrolladores describen la aplicación en términos de modelos en lugar de código. Y la segunda en el patrón de diseño Modelo, Vista y Controlador (MVC), en la cual se mantiene la lógica de presentación y la lógica de negocios aislados.

Estos dos modelos permiten la integración de otros programas con una sencilla interfaz. Debido a las normas de la aplicación de código abierto se puede integrar con otras aplicaciones.

Mi estancia en la empresa OPENTIX, S.L. se divide en dos partes: las prácticas extracurriculares con una duración de 1 mes y 21 días en la cual el objetivo principal era la adquisición de conocimientos sobre la tecnología empleada y las herramientas utilizadas dentro del departamento de desarrollo de la empresa y posteriormente un contrato de trabajo temporal con una duración de 3 meses (a tiempo parcial).

2.2. Proyecto formativo

El proyecto formativo durante la estancia en prácticas tiene como objetivo la adquisición de conocimientos prácticos en tareas de formación, consultoría y soporte de Openbravo, soporte básico y formación básica a clientes y parametrizaciones varias básicas del sistema, así como la formación de nuevas herramientas empleadas para el desarrollo del soporte en Openbravo.

2.3. Trabajo desarrollado

En este apartado explicaré desde principio a fin mi paso por la empresa Opentix, con algún ejemplo práctico de las tareas que desarrolle allí como programadora.

Durante en el primer contrato de prácticas extracurriculares, las primeras semanas dentro de la empresa fue recolectar información de la jerarquía de la empresa y como estaba estructurada departamentalmente. Después se me asignó al equipo de desarrolladores de Openbravo, en la que realice los siguientes cursos:

TT1 (Technical Training 1) oficial de Openbravo.

TT2 (Technical Training 1) oficial de Openbravo.

Tras finalizar los cursos, comence haciendo pequeñas tareas de mantenimiento en Openbravo de diferentes clientes para asentar los conocimientos adquiridos en los cursos ya que se requiere de tiempo para aprender bien la tecnología.

La consultoría realizada consiste en explicar y resolver dudas a los clientes sobre el funcionamiento de ERP.

Una vez adquiridos estos conocimientos me asignaron 3 clientes a los cuales darles soporte junto con otros clientes para soporte puntual, con los cuales me comunicaba casi diariamente.

El soporte consiste tanto en solventar problemas que le surgen a los clientes derivados de errores de programación o una mala configuración de la parametrización dentro de Openbravo en cuanto a la facturación, contabilidad, inventario, transacciones, etc de la empresas dentro de la plataforma, como crear nuevas funcionalidades o procesos que necesiten para la expansión, mejoras o cambios en la empresa o gestión de ella, por ejemplo, algunas de las nuevas funcionalidades podrían ser: crear un botón, ventana, campo, evento, modificación/creación de informes etc.

Ejemplo Práctico de soporte

Crear una ventana que permita añadir, por cada Business Partner y producto la cantidad vendida de ese producto, junto con las medidas de este y su precio (de forma anual y mensual).

2.4. Tecnologías empleadas

Durante el soporte a clientes se utilizan las siguientes tecnologías:

- Postgresql: sistema de base de datos relacional el cual usaba para acceder a las bases de datos de la empresa para obtener, comparar, insertar o actualizar datos.
- Java: lenguaje de programación para la realización, modificación y actualización de procesos, evenhandler, etc.
- SQL: lenguaje utilizado para consultar las bases de datos en postgres
- Gnome: utilizado para acceder a las máquinas de test y producción para subir los módulos modificados o hacer modificaciones de algún dato en la base de datos de producción.
- Modularidad: Openbravo utiliza modulos con versiones con el fin de reutilizar funcionalidades en otros clientes y facilitar la corrección de errores.

Capítulo 3

La interpolación

Contents

3.1. Introducción	19
3.2. Interpolación polinómica	20
3.2.1. Método de interpolación de Lagrange	21
3.2.2. Método de Newton (Diferencias divididas)	24
3.2.3. Splines	26
3.3. Métodos de interpolación	30
3.3.1. Interpolación lineal	30
3.3.2. Interpolación Bilineal	31
3.3.3. Interpolación bicúbica	35
3.3.4. Interpolación del vecino más cercano	39

3.1. Introducción

La interpolación es un proceso matemático que tiene como objetivo calcular datos a partir de datos ya conocidos, es un subcampo matemático del análisis numérico.

En otros campos científicos es frecuente disponer de cierto número de puntos obtenidos por muestreo o a partir de un experimento y tener como objetivo construir una función que los ajuste.

Otro objetivo de la interpolación es el que consiste en aproximar una función más compleja a partir de una más simple. Si nos encontramos con una función en la que la realización de cálculos

resulta costoso, podemos partir de ciertos valores e interpolar estos construyendo una función más simple. No obtendremos los mismos resultados evaluando la función obtenida en lugar de la función original, aunque dependiendo del tipo de problema y del método de interpolación utilizado, compensa ganar eficiencia frente al error cometido.

Podemos ver la interpolación de un modo más general de la siguiente manera:

Definición 1 Consideramos n parejas de puntos (x_i, y_i) , a las que denominaremos **datos conocidos**. Queremos construir una función $f(x)$ que verifique que :

$$f(x_i) = y_i \text{ con } i = 1, 2, \dots, n,$$

a la cual se denomina función interpolante de dichos puntos. A los puntos x_i se les llama nodos.

Existen diferentes métodos para encontrar dicha función. Los más conocidos son los que interpolan $f(x)$ mediante un polinomio o una función racional.

Planteamiento del problema :

Dada una función $f : [a, b] \rightarrow \mathbb{R}$, cuyo valor se conoce en $n + 1$ puntos: x_0, x_1, \dots, x_n del intervalo $[a, b]$: $f(x_0) = y_0, \dots, f(x_n) = y_n$, se debe construir una nueva función $F(x)$, perteneciente a una determinada clase de funciones, de tal modo que $F(x_i) = y_i$, para todo $i = 0, 1, \dots, n$.

3.2. Interpolación polinómica

Definición 2 Dados $n + 1$ puntos (x_i, y_i) con $i = 0, 1, \dots, n$ y $x_i \neq x_k$ si $i \neq k$, llamamos interpolación polinómica a determinar un polinomio de grado menor o igual que n tal que:

$$P(x_i) = y_i \text{ para todo } i=0, 1, \dots, n.$$

El Teorema Fundamental del álgebra garantiza que el polinomio de grado $n - 1$ existe y además es único, como demostraremos en el anexo I.

Teorema 1 Dados x_0, x_1, \dots, x_n números reales distintos entre sí y valores arbitrarios y_0, y_1, \dots, y_n , existe un único polinomio $P(x)$ de grado menor o igual que n tal que $P(x_i) = y_i$ ($i = 0, \dots, n$).

La demostración del teorema anterior se encuentra en el anexo I.

Existen diferentes métodos para determinar el polinomio interpolador. Independientemente de cual hayamos usado, el resultado será el mismo.

Aunque existe uno y solamente un polinomio de interpolación de grado menor o igual que n asociado a los datos de partida, dicho polinomio puede expresarse de maneras distintas y se llega a él a través de distintos algoritmos, lo que se traduce en polinomios de interpolación con nombres diversos.

A continuación veremos dos de estos métodos: el método de Lagrange y de Newton.

3.2.1. Método de interpolación de Lagrange

El método de interpolación de Lagrange es un problema clásico de interpolación.

El objetivo de este método es encontrar el polinomio interpolador $P(x)$ de grado menor o igual a n tal que $P(x_i) = y_i$.

Los puntos x_0, x_1, x_n se llaman nodos de interpolación.

Veremos en primer lugar como se construye el polinomio interpolador de grado 1.

Construcción del polinomio interpolador de Lagrange **grado 1:**

Se busca un polinomio de primer grado que interpole los valores de $f(x)$ en dos puntos: $(x_0, y_0), (x_1, y_1)$

Este polinomio resulta ser el p de grado uno (recta) que pasa por los dos puntos distintos (x_0, y_0) y (x_1, y_1) .

Para definirlo, consideraremos:

$$L_0(x) = \frac{x-x_1}{x_0-x_1} \text{ y } L_1(x) = \frac{x-x_0}{x_1-x_0}$$

y el polinomio :

$$P_1(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = \frac{x-x_1}{x_0-x_1}f(x_0) + \frac{x-x_0}{x_1-x_0}f(x_1)$$

Como $L_0(x_0) = 1$, $L_0(x_1) = 0$, $L_1(x_0) = 0$ y $L_1(x_1) = 1$, se tiene que:

$$P_1(x_0) = 1f(x_0) + 0f(x_1) = f(x_0) \text{ y } P_1(x_1) = 0f(x_0) + 1f(x_1) = f(x_1).$$

Así, $P_1(x) = L_0(x)f(x_0) + L_1(x)f(x_1)$ es la única función lineal que pasa por (x_0, y_0) y (x_1, y_1) .

Construcción del polinomio interpolador de Lagrange **grado n**:

Este método consiste en construir el polinomio interpolador de grado n que pasa por n + 1 puntos (x_i, y_i) de la forma

$$P_n(x) = \sum_{i=0}^n L_i(x)y_i$$

donde las funciones $L_i(x)$ cumplen que $L_i(x_j) = 0$ si $i \neq j$ y $L_i(x_i) = 1$. Esta propiedad garantiza que $P_n(x_i) = y_i$.

Las funciones $L_i(x_i)$ vienen dadas por :

$$L_i(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_1)(x_i-x_2)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \text{ para todo } i= 1, 2, \dots, n$$

Se tiene claramente que $L_i(x_i) = 1$ y $L_i(x_j) = 0$.

El método de Lagrange permite construir a partir de un conjunto de nodos x_1, x_2, \dots, x_n de forma explícita el polinomio interpolador.

Teorema 2 Si x_0, x_1, \dots, x_n son $n+1$ números distintos entre sí y f es una función de la que se conocen sus valores en dichos números, entonces el polinomio interpolador de Lagrange viene dado por la expresión:

$$P(x) = L_0(x)f(x_0) + \dots + L_n(x)f(x_n) = \sum_{i=0}^n L_i(x)f(x_i)$$

donde,

$$L_i = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \text{ para cada } i=0, 1, \dots, n.$$

El principal inconveniente del método de Lagrange es el momento en el que hay un gran número de nodos, ya que también supone aumentar el grado del polinomio, lo que implica dificultades en el cálculo.

Error de interpolación

En esta sección calcularemos la cota del error cometido al aproximar una función mediante el polinomio interpolador de Lagrange.

Si $P(x)$ interpola la función $f(x)$ en los puntos distintos entre sí x_0, x_1, \dots, x_n de un intervalo $[a, b]$ y $f \in C_{n+1}([a, b])$, entonces, para cada x en $[a, b]$ existe un número $c(x) \in (a, b)$ tal que :

$$f(x) = P(x) + \frac{f^{(n+1)}(c(x))}{(n+1)!}(x - x_0)(x - x_1)\dots(x - x_n)$$

Si la derivada de orden $n+1$ es una función acotada en $[a, b]$, entonces:

$$|f^{(n+1)}(x)| \leq M \Rightarrow |f(x) - P(x)| \leq \frac{M}{(n+1)!}(b - a)^{n+1}$$

Y así puede estimarse el error cometido en la aproximación $f(x) \approx P(x)$ mediante esta acotación. Para disminuir el error puede acortarse la amplitud del intervalo (a, b) o bien aumentar el número de puntos.

Aunque, hay que tener cuidado pues, si los datos tienen errores, la exactitud disminuye al aumentar n .

Ejemplo método interpolación de Lagrange

Ejercicio 1 *Calcular el polinomio de Lagrange usando los siguientes datos:*

x	y
1	-2
-3	1
5	2
7	-3

Cuadro 3.1: Datos ejemplo método de Lagrange

Aplicamos la fórmula anterior:

$$L_0 = \frac{(x+3)(x-5)(x-7)}{(1+3)(1-5)(1-7)} = \frac{(x+3)(x-5)(x-7)}{96}$$

$$L_1 = \frac{(x-1)(x-5)(x-7)}{(-4)(-8)(-10)} = \frac{(x-1)(x-5)(x-7)}{-320}$$

$$L_2 = \frac{(x-1)(x+3)(x-7)}{(5)(8)(-2)} = \frac{(x-1)(x+3)(x-7)}{-80}$$

$$L_3 = \frac{(x-1)(x+3)(x-5)}{(6)(10)(2)} = \frac{(x-1)(x+3)(x-5)}{120}$$

Sustituyendo, el polinomio de Lagrange queda definido de la siguiente manera:

$$f(x) = -2 \frac{(x+3)(x-5)(x-7)}{96} + \frac{(x-1)(x-5)(x-7)}{-320} + 2 \frac{(x-1)(x+3)(x-7)}{-80} - 3 \frac{(x-1)(x+3)(x-5)}{120}$$

$$f(x) = -\frac{(x-3)(x-5)(x-7)}{48} - \frac{(x-1)(x-5)(x-7)}{-320} - \frac{(x-1)(x+3)(x-7)}{40} - \frac{(x-1)(x+3)(x-5)}{40}$$

$$f(x) = -0,0739x^3 + 0,3906x^2 + 0,624x - 2,978$$

3.2.2. Método de Newton (Diferencias divididas)

Dada una colección de n puntos, x_0, x_1, \dots, x_n y sus respectivas imágenes, buscaremos escribir el polinomio de interpolación en la forma:

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_{n-1}(x - x_0)\dots(x - x_{n-1})$$

donde los a_k se definen de la siguiente manera:

Definición 3 Los coeficientes a_k , se denotan por $a_k = f[x_0, x_1, \dots, x_k]$ y se generan de forma recursiva mediante la fórmula :

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_0, x_1, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{(x_k - x_0)} \quad k=1, 2, \dots, n$$

partiendo de $f[x_0] = f(x_0)$. A estos coeficientes se les denominan diferencias divididas de la función f en los puntos x_0, x_1, \dots, x_k .

Con esta notación, el polinomio de interpolación puede escribirse como:

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ + \dots + f[x_0, x_1, x_2, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1})$$

Ejemplo método de Newton (diferencias divididas)

Ejercicio 2 Dados los puntos:

$$(1, 2), (3, 3), (4, 2) \text{ y } (8, 10)$$

se quiere obtener el polinomio interpolante que pasa por ellos.

Como tenemos 4 puntos, sabemos el grado del polinomio interpolador será tres.

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$f[x_k, x_{k+1}, x_{k+2}, x_{k+3}]$
1	$f[1] = 2$	$f[1, 3] = \frac{3-2}{3-1} = \frac{1}{2}$	$f[1, 3, 4] = \frac{-1-\frac{1}{2}}{4-1} = -\frac{1}{2}$	$f[1, 3, 4, 8] = \frac{-\frac{1}{2}-\frac{3}{5}}{8-1} = \frac{11}{70}$
3	$f[3] = 3$	$f[3, 4] = \frac{2-3}{4-3} = -1$	$f[3, 4, 8] = \frac{2+1}{8-3} = \frac{3}{5}$	
4	$f[4] = 2$	$f[4, 8] = \frac{10-2}{8-4} = 2$		
8	$f[8] = 10$			

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$f[x_k, x_{k+1}, x_{k+2}, x_{k+3}]$
1	2			
3	3	$\frac{1}{2}$		
4	2	-1	$-\frac{1}{2}$	
8	10	2	$\frac{3}{5}$	$\frac{11}{70}$

3.2.3. Splines

La interpolación polinómica común presenta algunos inconvenientes cuando hay muchos nodos o si la función a interpolar está lejos de ser un polinomio. El incremento en el grado de polinomio que se produce al introducir nuevos puntos en la interpolación, lejos de dar más precisión, generalmente provoca mayores oscilaciones. La interpolación polinómica, spline, evita estas restricciones al limitar el grado de polinomio.

Definición 4 *Dados $n+1$ puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ tales que los usados cumplen $x_0 < x_1 < \dots < x_n$ (nodos), un spline de grado k es una función $S(x)$ cuya gráfica pasa por estos puntos de forma que:*

- a) *en cada intervalo $[x_{i-1}, x_i]$, $S \in \leq k$,*
- b) *S tiene derivada de orden $k-1$ continua en $[x_0, x_n]$, es decir, S es de clase $C^{k-1}([x_0, x_n])$.*

Por consiguiente, S es un polinomio continuo a trozos de grado a lo sumo k , que tiene derivadas continuas hasta el orden $k-1$.

Spline de grado $k = 0$:

Los splines de grado cero son funciones constantes a trozos. Una forma explícita de presentar un spline de grado cero es la siguiente:

$$S(x) = \begin{cases} c_0 & \text{si } x \in [x_0, x_1) \\ c_1 & \text{si } x \in [x_1, x_2) \\ \dots & \\ c_{n-1} & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

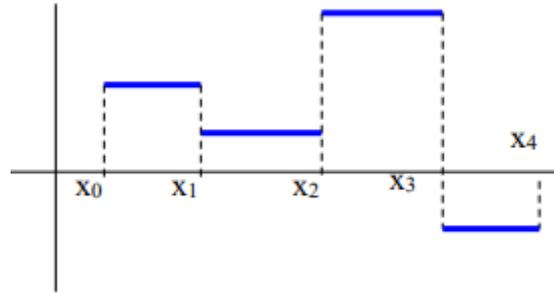


Figura 3.1: Splines de grado 0

Spline de grado $k = 1$:

Los splines de grado uno son funciones lineales a trozos que se unen de manera continua:

$$S(x) = \begin{cases} S_0 = a_0x + b_0 & \text{si } x \in [x_0, x_1) \\ S_1 = a_1x + b_1 & \text{si } x \in [x_1, x_2) \\ \dots & \\ S_{n-1} = a_{n-1}x + b_{n-1} & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

Para que sea continúa, los polinomios que la definen deben coincidir en los nodos, es decir:

$$a_i x_{i+1} + b_i = a_{i+1} x_{i+1} + b_{i+1}, i = 0, 1, \dots, n-1$$

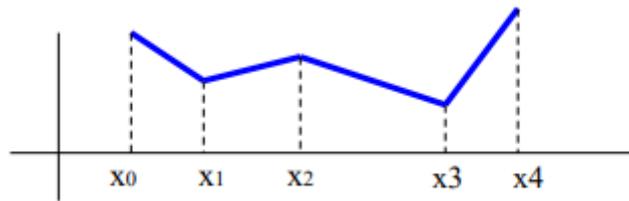


Figura 3.2: Splines de grado 1

Spline de grado k :

En general, los splines de grado k son polinomios de grado menor o igual que k que se unen de manera continua ellos y sus derivadas hasta el orden $k-1$.

Splines cúbicos

Los splines cúbicos son los más utilizados. Son polinomios de grado menor o igual que tres que se unen de forma continua entre ellos y sus derivadas de primer y segundo orden.

Dada la tabla de valores

x	y
x_0	y_0
x_1	y_1
...	...
x_n	y_n

Cuadro 3.2: Valores splines cúbicos

El objetivo es construir un spline cúbico para interpolar la tabla (3.1). Para esto tenemos que obtener los coeficientes del polinomio cúbico de cada subintervalo y determinar de forma simultánea los coeficientes en todos los subintervalos.

El spline viene dado por:

$$S(x) = \begin{cases} S_0(x) = a_{03}x^3 + a_{02}x^2 + a_{01}x + a_{00} & \text{si } x \in [x_0, x_1) \\ \dots & \\ S_{i-1}(x) = a_{i-13}x^3 + a_{i-12}x^2 + a_{i-11}x + a_{i-10} & \text{si } x \in [x_{i-1}, x_i) \\ S_i(x) = a_{i3}x^3 + a_{i2}x^2 + a_{i1}x + a_{i0} & \text{si } x \in [x_i, x_{i+1}) \\ \dots & \\ S_{n-1}(x) = a_{n-13}x^3 + a_{n-12}x^2 + a_{n-11}x + a_{n-10} & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

Hay cuatro coeficientes en cada subintervalo, por lo tanto, tendremos $4n$ incógnitas en total. Los polinomios S_{i-1} y S_i interpolan el mismo valor en el punto x_i , luego :

$$S_{i-1}(x_i) = y_i \text{ y } S_i(x_i) = y_i, \quad i=1, 2, \dots, n-1$$

esto hace que S sea continua y da lugar a $2(n-1) = 2n-2$ condiciones que debemos exigir. Por otra parte, $S_0(x_0) = y_0$ y $S_{n-1}(x_n) = y_n$ son 2 condiciones más que debemos exigir. $S'(x)$ debe ser continua, por tanto

$$S'_{i-1}(x_i) = S'_i(x_i), \quad i=1, 2, \dots, n-1$$

Análogamente se obtienen otras $n-1$ condiciones imponiendo que la derivada segunda S'' sea también continua.

Por tanto, se han de verificar: $(2n-2) + 2 + 2(n-1) = 4n-2$ condiciones.

Es decir, imponiendo las condiciones de un spline cúbico, se obtiene un sistema de $4n-2$ ecuaciones con $4n$ incógnitas.

Se necesitan, por tanto, dos condiciones adicionales para que los coeficientes estén unívocamente determinados.

Para construir un spline cúbico interpolante a la función f en los nodos $x_0 < x_1 < \dots < x_n$, conviene expresar los polinomios cúbicos de la forma:

$$S_j(x) = a_j(x - x_j)^3 + b_j(x - x_j)^2 + c_j(x - x_j) + d_j$$

ya que de esta forma el sistema de ecuaciones que resulta de imponer las condiciones de la definición es más fácil de resolver.

Las interpolaciones más utilizadas son la lineal y la parabólica. De la interpolación lineal y bilineal hablaremos más adelante ya que están muy relacionadas con la interpolación en el procesamiento de imágenes.

A continuación veremos algunos métodos de interpolación.

3.3. Métodos de interpolación

3.3.1. Interpolación lineal

La interpolación lineal es un procedimiento muy utilizado para estimar los valores que toma una función en un intervalo en el cual conocemos sus valores en los extremos (x_0, y_0) , (x_1, y_1) . Para estimar este valor utilizamos la aproximación de la función $f(x)$ por medio de una recta.

Podríamos decir que entre todos los métodos de interpolación, la interpolación lineal es el método más sencillo, pero no es muy preciso. En este método tenemos como datos dos puntos (x_1, y_1) y (x_2, y_2) de la función f y nuestro objetivo es averiguar el valor de 'y' cuando se proporciona un valor de la abscisa tal que $x_1 < x < x_2$, como la función interpolante para este método es una recta usaremos la ecuación de la recta para averiguar dicho valor (y):

$$\frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0}$$

La interpolación lineal se basa en la semejanza de triángulos, la cual se explica más en detalle en el anexo I.

De modo que derivando la expresión anterior, podemos obtener 'y' que representa el valor desconocido para x :

$$y - y_0 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

$$y = y_0 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

$$y = y_0 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

Con la finalidad de entender mejor este método de interpolación veremos un ejemplo sencillo:

Ejercicio 3 *Consideramos una tabla que relaciona los gastos y los ingresos. Una persona quiere saber cuáles son los ingresos de Marzo si su empresa tuvo un gasto de 100.000 €*

Gastos(x)	Ingresos(y)
10.000	15.000
15.000	21.000
$x_0 = 25.000$	$y_0 = 45.000$
$x = 30.000$	$y = ?$
$x_1 = 50.000$	$y_1 = 95.000$

Cuadro 3.3: Ejemplo Interpolación lineal

Aplicamos la fórmula de la interpolación lineal, la cual ya hemos visto con anterioridad en (2) y sustituyendo los datos :

$$y = 45000 + \frac{95000-45000}{50000-25000} (30000 - 25000)$$

$$y = 45000 + \frac{50000}{25000} (5000)$$

$$y = 45000 + 2 (5000)$$

$$y = 55000$$

Por tanto, en el mes de Marzo esta empresa tuvo unos ingresos de 55.000 €

3.3.2. Interpolación Bilineal

La interpolación bilineal extiende de la interpolación lineal en el ámbito de funciones de dos variables.

Este método de interpolación consiste en realizar una interpolación lineal en una dirección y después en la dirección ortogonal, aunque cada uno de estos pasos es lineal, en conjunto la interpolación es cuadrática.

En este método tenemos como datos cuatro puntos de la función f:

$$(x_1, y_1), (x_1, y_2), (x_2, y_1) \text{ y } (x_2, y_2)$$

y queremos encontrar el valor de f en un punto (x,y) .

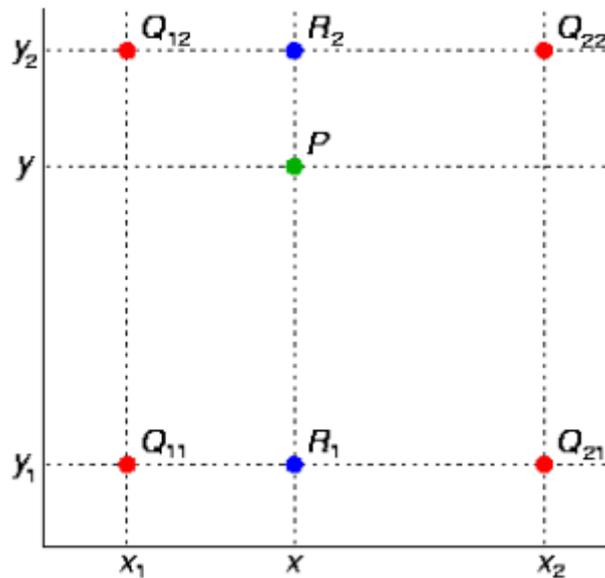


Figura 3.3: Interpolación bilineal.

donde $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, $Q_{22} = (x_2, y_2)$

En primer lugar se realiza una interpolación lineal en dirección x

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_1)$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_2)$$

En segundo lugar se realiza una interpolación lineal en dirección y

$$f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)$$

Esto proporciona una estimación de $f(x,y)$, es decir, el valor desconocido :

$$\begin{aligned}
f(x, y) &\approx \left. \begin{aligned}
&\frac{f(x_1, y_1)}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \\
&\frac{f(x_2, y_1)}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) + \\
&\frac{f(x_1, y_2)}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \\
&\frac{f(x_2, y_2)}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1) + \\
&= \frac{f(x_2, y_1)(y_2 - y_1)}{(x_2 - x_1)(y_2 - y_1)} (f(x_1, y_1)(x - x_1)(y_2 - y_1) + \\
&f(x_1, y_2)(x_2 - x_1)(y - y_1) + \\
&f(x_2, y_2)(x - x_1)(y - y_1)
\end{aligned} \right)
\end{aligned}$$

En este caso hemos interpolado primero en la dirección x y después en la dirección y , pero si hubieramos interpolado primero en la dirección y saldrían los mismos resultados.

Cuadrado unidad

Si se escoge un sistema de coordenadas en los cuales los cuatro puntos donde f es conocido sean $(0, 0)$, $(0, 1)$, $(1, 0)$ y $(1, 1)$, entonces la fórmula de interpolación se simplifica notablemente a:

$$f(x, y) \approx f(0, 0)(1 - x)(1 - y) + f(1, 0)x(1 - y) + f(0, 1)(1 - x)y + f(1, 1)xy$$

o, equivalentemente, en forma matricial:

$$f(x, y) \approx \begin{bmatrix} 1 - x & x \end{bmatrix} \begin{bmatrix} f(0, 0) & f(0, 1) \\ f(1, 0) & f(1, 1) \end{bmatrix} \begin{bmatrix} 1 - y \\ y \end{bmatrix}$$

La interpolación bilineal no es lineal

Definición 5 Diremos que la función $f : D \mapsto R$ es lineal si:

$$f(ax + by) = af(x) + bf(y) \quad \text{para todo } x, y \in D \text{ y } a, b \in R.$$

Ejercicio 4 Sea $f : \mathbf{R}^2 \rightarrow \mathbf{R}$ tal que $f(x_1, x_2) = (x_1 - x_2, 2x)$

Probar que es una aplicación lineal.

Sea $\vec{v} = (x_1, y_1)$ y sea $\vec{w} = (x_2, y_2)$

$$\vec{v} + \vec{w} = (x_1 + x_2, y_1 + y_2)$$

$$f(\vec{v}) = (x_1 - y_1, 2 * x_1)$$

$$f(\vec{w}) = (x_2 - y_2, 2 * x_2)$$

$$f(x_1 + x_2, y_1 + y_2) = (x_1 + x_2 - y_1 - y_2, 2 * x_1 + 2 * x_2)$$

Como $f(\vec{v} + \vec{w}) = f(\vec{v}) + f(\vec{w})$ f es lineal.

La interpolación bilineal dentro del contexto del procesamiento de imágenes, es una técnica básica para el remuestreo.

A continuación mostraremos un ejemplo sencillo en el que aplico el método de interpolación bilineal:

Ejercicio 5 Calcular el valor de Intensidad en el píxel 4 de la Figura 3.4

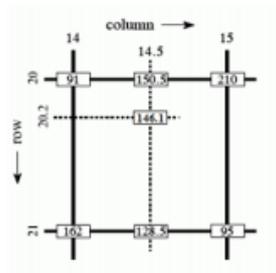


Figura 3.4: Píxeles escala de grises

Nota: En el siguiente capítulo veremos que una imagen digital puede definirse como una función bidimensional $f(x, y)$ donde x e y son las coordenadas espaciales. El valor de f en cualquier par de coordenadas (x, y) se denomina intensidad de píxel.

El valor de la intensidad en el píxel 4 lo calcularemos interpolando linealmente entre los valores 1 y 3 en cada una de las filas (es decir, también entre los valores 5 y 7) :

$$I_{20,14,5} = \frac{15-14,5}{15-14} * (91) + \frac{14,5-14}{15-14} * (95) = 150,5$$

$$I_{20,2,14,5} = \frac{15-14,5}{15-14} * (162) + \frac{14,5-14}{15-14} * (95) = 128,5$$

y luego interpolar linealmente entre estos valores :

$$I_{0,21,14,5} = \frac{21-20,2}{21-20} * (150,5) + \frac{21-20,2}{21-20} * (285) = 146,1$$

Por tanto la intensidad del píxel 4 es 146.1 .

Por último mencionar que también existe el método de interpolación trilineal, o también llamada cúbica, que explicaremos a continuación (interpolación lineal a tres dimensiones).

3.3.3. Interpolación bicúbica

La interpolación bicúbica en matemáticas es una extensión de la interpolación cúbica que se utiliza para interpolar puntos de datos en una malla regular de dos dimensiones.

La interpolación bicúbica se puede obtener utilizando polinomios de Lagrange, splines cúbicos, o algoritmos de convolución cúbica.

Para este método supongamos que los valores de la función $f(x,y)$ y las derivadas f_x , f_y y f_{xy} son conocidas en las cuatro esquinas $(0, 0)$, $(1, 0)$, $(0, 1)$ y $(1, 1)$ en el cuadrado unitario. La superficie interpolada puede entonces ser escrita.

$$P(x,y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

El problema de interpolación consiste en determinar los 16 coeficientes a_{ij} . Correspondiente de $p(x, y)$ con los valores de la función produce cuatro ecuaciones,

1. $f(0, 0) = p(0, 0) = a_{00}$
2. $f(1, 0) = p(1, 0) = a_{00} + a_{10} + a_{20} + a_{30}$
3. $f(0, 1) = p(0, 1) = a_{00} + a_{01} + a_{02} + a_{03}$

$$4. f(1, 1) = p(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}$$

Del mismo modo, ocho ecuaciones para las derivadas en la dirección de x y la dirección de y:

$$1. f_x(0, 0) = p_x(0, 0) = a_{10}$$

$$2. f_x(1, 0) = p_x(1, 0) = a_{10} + 2a_{20} + 3a_{30}$$

$$3. f_x(0, 1) = p_x(0, 1) = a_{10} + a_{11} + a_{12} + a_{13}$$

$$4. f_x(1, 1) = p_x(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}i$$

$$5. f_y(0, 0) = p_y(0, 0) = a_{01}$$

$$6. f_y(1, 0) = p_y(1, 0) = a_{01} + a_{11} + a_{21} + a_{31}$$

$$7. f_y(0, 1) = p_y(0, 1) = a_{01} + 2a_{02} + 3a_{03}$$

$$8. f_y(1, 1) = p_y(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}j$$

Y cuatro ecuaciones para la derivada cruzada xy :

$$1. f_{xy}(0, 0) = p_{xy}(0, 0) = a_{11}$$

$$2. f_{xy}(1, 0) = p_{xy}(1, 0) = a_{11} + 2a_{21} + 3a_{31}$$

$$3. f_{xy}(0, 1) = p_{xy}(0, 1) = a_{11} + 2a_{12} + 3a_{13}$$

$$4. f_{xy}(1, 1) = p_{xy}(1, 1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}ij$$

donde en las expresiones anteriores se han utilizado las siguientes identidades:

$$p_x(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}iX^{i-1}Y^j$$

$$p_y(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}jX^iY^{j-1}$$

$$p_{xy}(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}ijX^{i-1}Y^{j-1}$$

Este procedimiento produce una superficie $p(x, y)$ en el cuadrado unitario $[0, 1] \times [0, 1]$ la cual es continua y con derivadas continuas. La interpolación bicúbica en una cuadrícula de tamaño arbitrariamente regular se puede lograr mediante un conjunto de parches de dichas superficies,

asegurando que los instrumentos derivados corresponden en los límites. La agrupación de los parámetros desconocidos a_{ij} en un vector :

$$\alpha = [a_{00}, a_{10}, a_{20}, a_{30}, a_{01}, a_{11}, a_{21}, a_{31}, a_{02}, a_{12}, a_{22}, a_{32}, a_{03}, a_{13}, a_{23}, a_{33}]^T$$

y dejar :

$$x = [f(0,0), f(1,0), f(0,1), f(1,1), f_x(0,0), f_x(1,0), f_x(0,1), f_x(1,1), f_y(0,0), f_y(1,0), f_y(0,1), f_y(1,1), f_{xy}(0,0), f_{xy}(1,0), f_{xy}(0,1), f_{xy}(1,1)]^T$$

El sistema anterior de ecuaciones se puede reformular en una matriz para la ecuación lineal $A_\alpha = x$. La inversa de la matriz da la ecuación lineal más útil $A^{-1}x = \alpha$ permite que α pueda calcularse rápida y fácilmente, donde

$$A^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 0 & 0 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 1 & 1 & 0 & 0 \\ -3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & -1 & 0 \\ 9 & -9 & -9 & 9 & 6 & 3 & -6 & -3 & 6 & -6 & 3 & -3 & 4 & 2 & 2 & 1 \\ -6 & 6 & 6 & -6 & -3 & -3 & 3 & 3 & -4 & 4 & -2 & 2 & -2 & -2 & -1 & -1 \\ 2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ -6 & 6 & 6 & -6 & -4 & -2 & 4 & 2 & -3 & 3 & -3 & 3 & -2 & -1 & -2 & -1 \\ 4 & -4 & -4 & 4 & 2 & 2 & -2 & -2 & 2 & -2 & 2 & -2 & 1 & 1 & 1 & 1 \end{bmatrix}$$

No puede haber otra forma de matriz concisa de 16 coeficientes

$$\begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 \end{bmatrix},$$

o

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix},$$

donde,

$$p(x, y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix}$$

Si las derivadas son desconocidas por lo general son aproximadas a partir de los valores de la función en los puntos vecinos de las esquinas del cuadrado unitario, por ejemplo, usando diferencias finitas.

Para encontrar cualquiera de las derivadas individuales, f_x o f_y , utilizando ese método, encontrar la pendiente entre los dos puntos de alrededor en el eje apropiado. Por ejemplo, para calcular f_x para uno de los puntos, encontramos $f(x, y)$ para los puntos de la izquierda y derecha del punto de búsqueda y calculamos su pendiente, y lo mismo para f_y . Para calcular la derivada cruzada, f_{xy} , tomamos la derivada en ambos ejes, uno a la vez. Por ejemplo, se puede calcular primero f_x (derivada de x) de los puntos encima y por debajo del punto de búsqueda, luego, calculamos f_y en esos valores para obtener el valor de $f_{xy}(x, y)$ para el punto de destino.

También se puede hacer en dirección opuesta, en primer lugar calcular f_y y luego f_x , los dos resultados son equivalentes.

En los bordes de la base de datos, cuando falta algunos de los puntos circundantes, los puntos que faltan pueden aproximarse mediante uno de una serie de metodos. Un método simple y común es asumir que la pendiente desde el punto hasta el punto de destino existente continúa sin más cambios, y usando esto para calcular un valor hipotético para el punto que falta.

3.3.4. Interpolación del vecino más cercano

La interpolación del vecino más cercano también es conocida como interpolación proximal o muestreo del punto. Es un método sencillo de interpolación en una o más dimensiones.

El algoritmo de este método sólo tiene en cuenta el valor del punto más cercano ignorando los valores de los puntos vecinos.

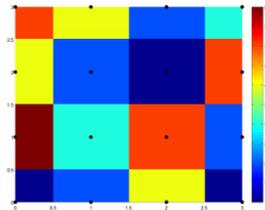


Figura 3.5: Interpolación del vecino más cercano

Búsqueda del vecino más cercano

La búsqueda del vecino más cercano, búsqueda de proximidad o búsqueda del punto más cercano (NNS), es un problema de optimización para encontrar el punto más cercano (o más similar) teniendo en cuenta un conjunto de puntos dados.

El planteamiento del problema de búsqueda del vecino más cercano es el siguiente:

Dado un conjunto de puntos C en un espacio E ($C \subset E$) y $p \in E$, queremos encontrar el punto más cercano de C para p (C_p).

Existen diferentes formas de resolver el problema NNS. Veamos tres métodos:

- la búsqueda radial
- el espacio de partición

- búsqueda lineal

La búsqueda lineal

La búsqueda lineal es la solución más simple para el problema NNS, consiste en calcular la distancia desde el punto p a cada uno de los puntos del conjunto de puntos C , haciendo un seguimiento mirando " el mejor hasta ahora ". Este algoritmo tiene un tiempo de ejecución $O(dN)$ donde N es la cardinalidad de C y d es la dimensionalidad de E .

La búsqueda lineal es la solución más simple al problema NNS.

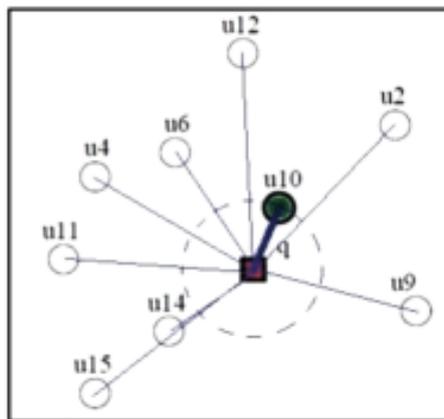


Figura 3.6: Búsqueda lineal

Relación con el diagrama de Voronoi

Los diagramas de Voronoi son fundamentales dentro de la Geometría Computacional, contienen toda la información de la proximidad entre puntos y tienen muchas aplicaciones.

Dado un conjunto de puntos C en un espacio E ($C \subset E$), un diagrama de Voronoi es una descomposición del espacio en celdas, cada celda contiene un punto situado de tal forma que en cualquier lugar del espacio, el punto más cercano es el interior de la celda.

Si nos fijamos podemos deducir que esto es equivalente al método de interpolación del vecino más cercano, si asignáramos el valor de la función en el punto p a todos los puntos dentro de la celda.

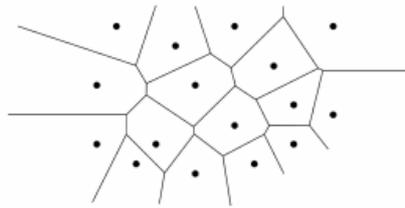


Figura 3.7: Diagrama de Voronoi

El algoritmo de Fortune

El algoritmo de Fortune es un algoritmo de línea de barrido para generar un diagrama de Voronoi de un conjunto de puntos en un plano utilizando un tiempo $O(n \log n)$ y espacio $O(n)$. Fue publicado originalmente por Steven Fortune en 1986 en su artículo "Un algoritmo sweepline para los diagramas de Voronoi."

Descripción del algoritmo de Fortune :

El algoritmo mantiene tanto una línea de barrido y una línea de varado, la cual ambas se mueven a través del plano a medida que el algoritmo avanza. La línea de barrido es una línea recta, que es posible que, por convención, suponemos que es vertical y moviéndose de izquierda a derecha a través del plano. En cualquier momento durante el algoritmo, los puntos de entrada a la izquierda de la línea de barrido se han incorporado en el diagrama de Voronoi, mientras que los puntos a la derecha de la línea de barrido no se han considerado todavía. La línea de varado no es una línea, sino una curva a trozos complicada a la izquierda de la línea de barrido, compuesto por piezas de parábolas; se divide la porción del plano en el que el diagrama de Voronoi puede ser conocido, independientemente de lo que otros puntos pueden ser derecha de la línea de barrido, del resto del plano.

Para cada punto de la izquierda de la línea de barrido, se puede definir una parábola de puntos equidistantes a partir de ese punto y de la línea de barrido; la línea de varado es el límite de la unión de estas parábolas. A medida que la línea de barrido avanza, los vértices de la línea de varado, en el que se cruzan dos parábolas, trazan los bordes del diagrama de Voronoi. La línea de varado progresa manteniendo cada base de la parábola exactamente a medio camino entre los puntos de barrido inicialmente sobre la línea de barrido, y la nueva posición de la línea de barrido.

El algoritmo mantiene como estructuras de datos un árbol de búsqueda binaria que describe

la estructura combinatoria de la línea de varado, y una cola de prioridad lista de posibles eventos futuros que podrían cambiar la estructura de la línea de varado. Estos eventos incluyen la adición de otra parábola a la línea de varado (cuando la línea de barrido cruza otro punto de entrada) y la eliminación de una curva de la línea de varado (cuando la línea de barrido se hace tangente a un círculo a través de unos tres puntos de entrada cuyo parábolas forman segmentos consecutivos de la línea de varado). Cada uno de estos eventos puede ser priorizado por la coordenada x de la línea de barrido en el punto que se produce el evento.

El propio algoritmo entonces consiste en retirar repetidamente el próximo evento de la cola de prioridad, la búsqueda de los cambios que el evento provoca en la línea de varado, y la actualización de las estructuras de datos.

Como hay $O(n)$ eventos en proceso (estando cada uno asociado con alguna característica del diagrama Voronoi) y $O(\log n)$ tiempo para procesar un evento (cada uno compuesto de un número constante de operaciones árbol de búsqueda y cola de prioridad binarios) el tiempo total es $O(n \log(n))$.

Capítulo 4

Procesamiento de Imágenes

Contents

4.1. Introducción	43
4.2. Historia del Procesamiento de Imágenes	44
4.3. Conceptos Previos	45
4.3.1. Imágenes digitales	45
4.3.2. Tipos de imágenes Digitales	46
4.3.3. Píxel	46
4.3.4. Tamaño	50
4.3.5. Resolución	51
4.4. Interpolación en el procesamiento de imágenes	51
4.5. Remuestreo de imágenes	51
4.5.1. Interpolación Bilineal	52
4.5.2. ¿Qué método es mejor?	55

4.1. Introducción

El procesamiento de imágenes digitales es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

El procesamiento de imágenes es un campo muy importante de la visión artificial, que hace posible que un ordenador procese imágenes o fotografías bidimensionales. El procesamiento

digital de imágenes se puede dividir en las siguientes áreas: Adquisición, realce y mejora, segmentación y extracción de características.

La adquisición o captura se ocupa de las diferentes maneras de obtener una imagen , por ejemplo, una cámara.

El realce y la mejora de la imagen son las técnicas que se utilizan para mejorar la apariencia visual de las imágenes bien para restaurar imágenes que han sido degradadas o para recuperar imágenes. Esta es la parte del procesamiento de imágenes en la que más insistiremos puesto que es la parte en la que se aplica la interpolación.

La segmentación esta parte del proceso se ocupa de la división de las imágenes en áreas o regiones significativas.

La extracción de características engloba la detección y localización de entidades geométricas simples (líneas y puntos) y complejas (curvas y cuádricas)

Las aplicaciones del procesamiento de imágenes son muchas, normalmente se utiliza para darle estética a la imagen, esto se consigue modificando el brillo, el contraste, el color para eliminar los defectos de la imagen en el momento de la fotografía, etc.

4.2. Historia del Procesamiento de Imágenes

El concepto de imagen digital desde el punto de vista del procesamiento de imágenes se remonta a la década de 1920, entre sus primeras aplicaciones podemos destacar el envío de imágenes para prensa que se hacían entre Londres y Nueva York (con esta aplicación consiguieron acortar el tiempo de envío de las imágenes de una semana a tres horas).

Algunos años más tarde en 1957, el mundo pudo ver lo que se consideró la primera imagen digital recreada (no reproducida) en píxeles de la mano de Russel Kirsch. Este científico transformó una imagen en una matriz de ceros y unos, pudiendo así visualizarla en SEAC (el único ordenador programable de la época), su resolución era de 175×175 píxeles.

La imagen digital se realiza potencialmente en la década de los sesenta, fue en esta década en la que empezaron a usarse técnicas de manipulación de imágenes electrónicas utilizando el ordenador para mejorar las imágenes recibidas por sondas espaciales, estas imágenes eran escaneadas a baja velocidad para codificarlas digitalmente y más tarde escanearlas y volver a transmitir las.

En 1970, se produce un salto muy importante para la imagen digital con la llegada de los microprocesadores, que permitieron aumentar la capacidad de almacenamiento y procesamiento de imágenes. Durante esta época se construyó la primera cámara digital y de forma continuada la matriz de filtros y color, que permitiría captar los píxeles de la imagen en color.

En 1988, se da la primera videocamara totalmente digital, presentada por Fuji.

Cuando se implementaron las imágenes en los ordenadores, se necesitó un conjunto de elementos facilmente manipulables apra plasmar la imagen en la pantalla tal cual es en la fotografia. RGB, CMY, CMYK, HSV, entre otros, estos permitieron la combinación de colores, de luz, de contraste y valores de transparencia necesarios para ver la imagen en la pantalla o para su impresión.

Con esto se pudo comenzar a aplicar funciones matemáticas, álgebra lineal y métodos de procesamiento de datos en el campo de la programación para obtener mejores resultados. Además ya existían programas que permitian editar las imágenes con píxeles de forma sencilla.

4.3. Conceptos Previos

4.3.1. Imágenes digitales

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica, comunmente en binario (0 y 1).

Si tenemos en cuenta si la resolución de la imagen es dinámica o estática, puede ser un gráfico vectorial o una imagen matricial (mapa de bits).

Durante el proyecto nos centraremos más en las imágenes de mapa de bits también conocidas como imágenes rasterizadas, bitmap o pixmap, son imágenes formadas por píxeles.

Definición 6 *Una imagen digital se puede definir como una función bidimensional $f(x,y)$ donde (x,y) son las coordenadas espaciales, y el valor de f en cualquier (x,y) es la intensidad de la imagen en dicho punto.*

Viendo la imagen digital desde el punto de vista de la definición anterior, una imagen digital se puede considerar una función discreta sobre un dominio discreto.

4.3.2. Tipos de imágenes Digitales

Imágenes Digitales Rasters o Mapas de bits

En este caso la imagen es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación.

En este tipo de imágenes digitales cada píxel de la imagen tiene un color específico y mientras más resolución o más píxeles tenga, la imagen tendrá más calidad. Uno de los programas más utilizados para editar imágenes es el conocido Photoshop.

Imágenes vectoriales

Las imágenes vectoriales, no tienen píxeles como las mencionadas anteriormente, sino que parten de gráficos vectoriales que dependen de unos puntos de control, gracias a su contenido de fórmulas matemáticas están en la capacidad de formar curvaturas entre un punto y otro y cuando se aplica algún programa de edición, este simplemente calcula la fórmula y adapta la imagen al requerimiento del usuario.

Las imágenes vectoriales se utilizan para la elaboración de imágenes sencillas.

4.3.3. Píxel

Un píxel es la unidad mínima de visualización de una imagen digital compuesta normalmente por dígitos binarios (0 y 1 en escala de grises), contienen información sobre el color, la saturación y brillo. Estas características suelen estar dadas en coordenadas "x", y "y" de la pantalla, valores generalmente R (rojo), G (verde), B (azul) y valor Alfa (en color)

Para entender mejor este concepto podríamos pensar en una cuadrícula, donde cada celda tiene un valor numérico, se puede somparar con una matriz con el tamaño de la imagen en donde se colocan los valores que representan a la misma, si somos más específicos en el caso del modelo RGB se puede hablar de tres matrices en donde cada una tiene el valor de rojo, verde y azul correspondiente.

Existen diferentes estructuras de datos usadas para obtener datos de color.

La estructura más sencilla está compuesta de un bit, en donde se determina sólo el valor de

blanco o negro a un pixel. Es decir $2^1 = 2$ colores. Ejemplo:



Figura 4.1: Imagen monocromática

Después tenemos a la estructura con 4 bits, es decir, $2^4 = 16$ colores. Ejemplo:



Figura 4.2: Imagen con 16 colores

A continuación vemos en la figura (x) la de 8 bits, es decir, $2^8 = 256$ colores. Ejemplo:



Figura 4.3: Imagen con 256 colores

Y llegamos al estándar que son imágenes de 24 bits, es decir, $2^{24} = 6$ millones de colores. Ejemplo:

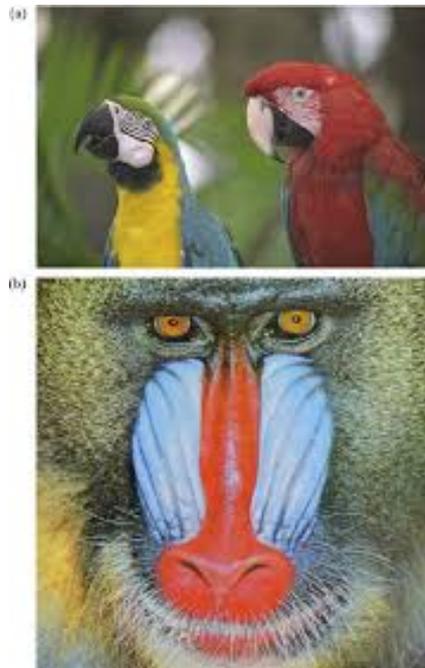


Figura 4.4: Imagen con 6 millones de colores

Existen más estructuras de datos, pero estas son las principales en cuanto a como se desarrolló tecnológicamente la manera de representar imágenes, aún sigue siendo un estándar utilizar

imagenes de 24 bits RGB.

Para las imagenes estandar tenemos 256 diferentes combinaciones de rojo, verde y azul:

R = 9 bits = 0000 1111 → Bits más significativos

G = 9 bits = 1010 1000

B = 9 bits = 1110 0000 → Bits menos significativos

Para hacer un total de RGB = 24 bits = 0000 1111 1010 1000 1110 0000 = valor de color.

Podemos hacernos una idea mas acertada del concepto de píxel con la figura 4.5, cada cuadrado es un píxel.

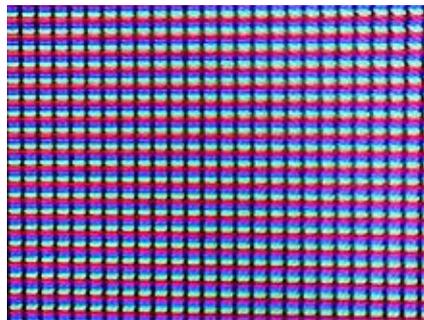


Figura 4.5: Píxeles

Nivel Digital (ND)

Al a hora de obtener la imagen en el proceso de ... del procesamiento de imágenes, el sensor detecta la radiancia media de la aprcela equivalente al tamaño de un píxel, este valor medio se traduce por el sensor a un valor numéricos a partir del cual se realiza el tratamiento de imágenes digitales.

Por ello cada píxel viene definido por un número entero, el cual denominamos nivel digital (ND)

4.3.4. Tamaño

Tamaño en píxeles

Como hemos mencionado anteriormente una imagen digital es el resultado de una matriz de píxeles, estos se reparten en filas y columnas de píxeles. El tamaño de la imagen es la suma de todos los píxeles que forman la imagen.

El tamaño en píxeles de la imagen está limitado por la capacidad del dispositivo de captura (teléfono móvil, cámara fotográfica, escáner, etc). Por ejemplo si empleamos un teléfono móvil con un sensor de 22 megapíxeles, solo podremos capturar imágenes como máximo de 22000000 píxeles (1 megapíxel = 1000000 píxeles)

Para verlo con claridad veremos un ejemplo, si vemos las propiedades de una imagen digital, en concreto la de la figura 4.6, en nuestro ordenador y observamos que el tamaño es de 259×195 , significa que la matriz de píxeles de esta imagen está compuesta por 259 columnas y 195 filas de píxeles, es decir el tamaño de esta imagen es de $259 * 195 = 50505$ píxeles.

Tamaño físico de la imagen

El tamaño físico de una imagen digital son las dimensiones físicas de esta (anchura y altura) expresadas en una unidad de longitud, normalmente en centímetros (cm).

El tamaño físico de la imagen depende de tamaño de la imagen en píxeles y de la resolución:

$$\text{tamaño físico (cm)} = (\text{tamaño en píxeles} * 2,54 \frac{\text{cm}}{\text{pulgada}}) / (\text{resolución en ppp})$$

Tamaño (peso) de archivo de la imagen

El peso o tamaño de archivo de una imagen digital es la cantidad de memoria que se ocupa al almacenar la imagen y se mide en bytes o múltiplos :

$$1 \text{ GB} = 1024 \text{ MB} = 1024 \times 1024 \text{ Bytes} = 1048576 \text{ Bytes}$$

El tamaño de archivo de una imagen digital depende de su tamaño en píxeles y de su profundidad en bytes

4.3.5. Resolución

La resolución de una imagen es la relación entre el tamaño en píxeles de una imagen, que como hemos visto anteriormente depende del dispositivo con el que ha sido capturada la imagen y sus dimensiones físicas, estas dependen del dispositivo de salida de la imagen como una impresora o una pantalla.

La resolución normalmente se expresa en píxeles por pulgada (ppp o dpi), esta propiedad sólo dependerá del dispositivo de salida de la imagen, contra más resolución tenga el dispositivo de salida más píxeles por unidad de longitud, dicho de otra forma el nivel de detalle de la imagen es mayor (mayor resolución = más definición)

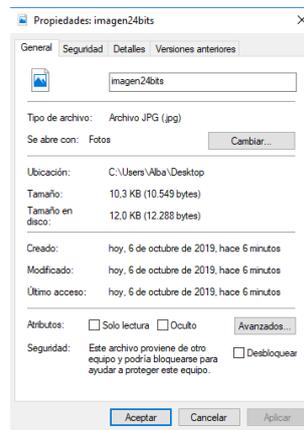


Figura 4.6: Propiedades de una Imágen

4.4. Interpolación en el procesamiento de imágenes

4.5. Remuestreo de imágenes

El remuestreo o interpolación radiométrica de una imagen digital esta dentro del proceso de realce y mejora en el procesamiento de imágenes.

Dentro las operaciones o transformaciones que se pueden realizar a una imagen con el fin de realzarlas y mejorarlas, el remuestreo está dentro de la categoría de transformaciones geométricas. El resultado de las transformaciones geométricas depende de las diferentes posiciones de los niveles de gris en la imagen de entrada.

Las transformaciones geométricas tienen los siguientes pasos de corrección geométrica :

- Definición de las funciones de transformación.
- Cálculo de las coordenadas de las esquinas de la imagen transformada y el número de filas y columnas que tendrá la nueva imagen en su matriz de píxeles.
- Creación de la nueva imagen con las características calculadas en el paso anterior.
- Para cada celda de la nueva imagen se determina su posición en la imagen original
- Remuestreo; Se asigna el ND (Nivel Digital) al píxel de la nueva imagen.

4.5.1. Interpolación Bilineal

En este apartado veremos un ejemplo de remuestreo de imágenes con el método de interpolación lineal.

En primer lugar debemos buscar que posición ocuparán los centros de los píxeles de la imagen transformada en la imagen original

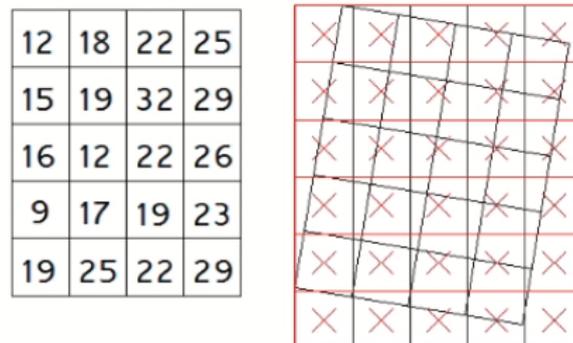


Figura 4.7: Ejemplo interpolación bilineal remuestreo

Si quisieramos calcular el nivel digital del píxel sombreado en la Figura X, lo que tenemos que hacer es calcular que posición ocupa el centro de ese píxel en la imagen original mediante las funciones de transformación inversa.

Ese píxel aparecerá en la imagen original y su nivel digital será calculado mediante interpolación bilineal a partir de los niveles digitales de los 4 vecinos más próximos, como hemos explicado anteriormente.

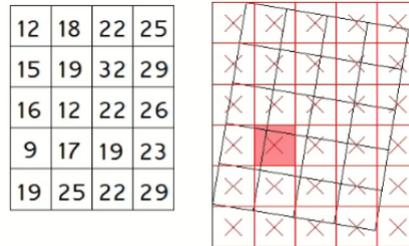


Figura 4.8: Ejemplo interpolación bilineal remuestreo

Las distancias que hay entre los centros de los píxeles que se van a tener en cuenta como más próximos, para el nivel digital, se corresponden con la resolución geométrica de la imagen original.

Esa resolución geométrica, es decir, esa distancia entre centro, la vamos a denominar d , tal y como podemos ver en la Figura 4.9.

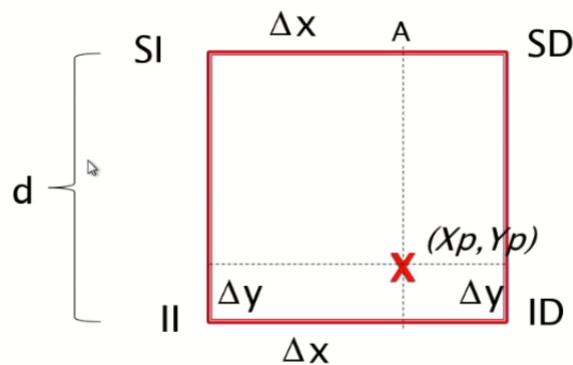


Figura 4.9: Ejemplo interpolación bilineal remuestreo

Como las coordenadas de los píxeles en la imagen original son conocidas, es decir, los centros de los píxeles de la imagen original son conocidos y además hemos calculado la posición que ocupa el centro del píxel de la imagen transformada, en la imagen original podemos calcular los incrementos de x (Δ_x) y de y (Δ_y) Con estos incrementos y con la distancia d podemos calcular los niveles digitales de los puntos A y B.

Para el cálculo del nivel digital del punto A vamos a recurrir a una regla de tres. Si la distancia entre 16 y 12, que son los centros de los píxeles de la imagen original, es decir, es d ($d=4$), podremos calcular la variación que hay en el nivel digital para la distancia Δ_x

$$\begin{array}{ccc} d & \text{—————} & (12-16) \\ & & \Delta_x \text{ ————— } a \end{array}$$

De la misma forma vamos a proceder con el punto B, si para la distancia d hay una variación en los niveles digitales de 8 (17-9). Para Δ_x habrá una variación que será igual a b. Con esa variación calcularemos el nivel digital que se corresponde con el punto B

$$\begin{array}{ccc} d & \text{—————} & (12-16) \\ & & \Delta_y \text{ ————— } b \end{array}$$

Los niveles digitales que se corresponden con A y con B en este caso son:
 Suponemos que $d = 25$ cm, $\Delta_x = 18$, $A = 16 + a = 16 - 2,9 = 13,1$
 $\Delta_y = , B = 9 + b = 9 + 5,8 = 14,8$

La variación de los niveles digitales entre A y B es de -2,9. Para la distancia Δ_y la variación será igual a -0,4, con esta variación más el nivel digital de B, podemos calcular el nivel digital que le corresponderá al punto buscado.

En este caso el valor del nivel digital es:

$$ND = 14,8 - 0,4 = 14,4$$

Como todos los niveles digitales de la imagen son números enteros, en realidad el valor que se transfiere al píxel es 14.

El método de remuestreo con Interpolación Bilineal lo que va a permitir es calcular los niveles digitales de una imagen corregida geoméricamente a partir de los niveles digitales de 4 píxeles de la imagen original.

Cada uno de esos píxeles influirá más o menos en función de la distancia que los separa de la posición calculada

Formula general de el nivel de píxel:

$$\begin{aligned}
 ND = & [(d - \Delta_x) * (\Delta_y) * ND(SI) + \\
 & (d - \Delta_x) * (d - \Delta_y) * ND(II)] + \\
 & (\Delta_x) * (\Delta_y) * ND(SD) + \\
 & (\Delta_x) * (d - \Delta_y) * ND(ID)] \cdot (dx \cdot dy)
 \end{aligned}$$

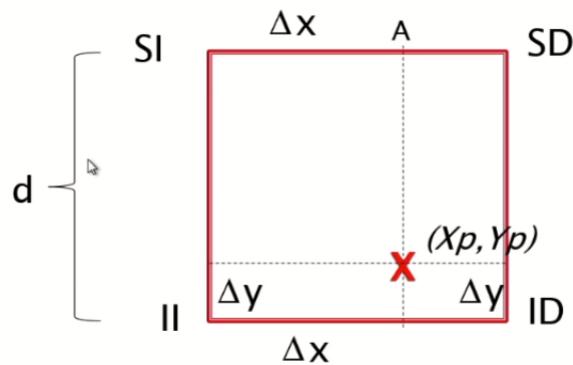


Figura 4.10: Ejemplo interpolación bilineal remuestreo

4.5.2. ¿Qué método es mejor?

En el procesamiento de imágenes, la interpolación bicúbica se elige a menudo sobre la interpolación bilineal o del vecino más cercano para el remuestreo de imagen, cuando la velocidad no es importante. En contraste con la interpolación bilineal, que sólo toma en cuenta 4 píxeles (2 2), la interpolación bicúbica considera 16 píxeles (4 4). Si se utiliza interpolación bicúbica la interpolación es más suave y tienen menos artefactos de interpolación.

Capítulo 5

Ejemplos de código para el procesamiento de imágenes digitales en MATLAB

Contents

5.1. Introducción	57
5.2. MATLAB	60
5.3. Interpolación lineal	61
5.4. Interpolación cúbica	61
5.5. Interpolación splines	62

5.1. Introducción

Durante este capítulo veremos ejemplos de como se remuestrean las imágenes. El código para generar las imágenes del ejemplo se situarán en el Anexo I

Utilizaremos la siguiente imagen de ejemplo:



Figura 5.1: Imagen Original

El remuestreo de una imagen es útil para ajustar la resolución y el tamaño, y también se puede utilizar para suavizar los píxeles después del zoom.

En las siguientes figuras podemos ver la misma imagen pero con mayor y menor resolución :

En el caso de mayor resolución, por cada 5 píxeles de la imagen original, la imagen interpolada tiene 6 píxeles.

Mayor Resolución



Figura 5.2: Imagen con mayor resolución

Menor Resolución



Figura 5.3: Imagen con menor resolución

A medida que amplía una imagen, los píxeles de la región de interés se vuelven más y más grandes y el detalle de la imagen se pierde rápidamente, como podemos observar en la siguiente figura:

10x Zoom

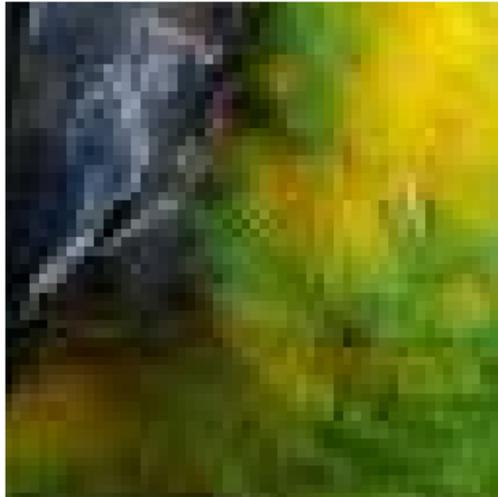


Figura 5.4: Imagen con 10x zoom

Estos ejemplos los veremos con la herramienta MATLAB.

5.2. MATLAB

MATLAB es el nombre abreviado de "MATrix LABoratory". MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Como caso especial, también puede funcionar con números escalares, tanto reales como complejos, con cadenas de caracteres y otras estructuras de información más complejas. Una de las características más atractivas es hacer una variedad de gráficos en dos y tres dimensiones. MATLAB también tiene su propio lenguaje de programación.

Es un programa interactivo orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.

Matlab es una herramienta interactiva basada en matrices para cálculos científicos y de ingeniería (de hecho, el termino matlab procede de matrix laboratory). Desde el punto de vista del control, matlab se puede considerar un entorno matemático de simulación que puede utilizarse

para modelar y analizar sistemas. Permitir a el estudio de sistemas continuos, discretos, lineales y no lineales, mediante descripción interna y externa, en el dominio temporal y frecuencia. Matlab constituye un entorno abierto, para el cual numerosas paquetes específicos adicionales (toolboxes) han sido desarrollados. En el caso que nos ocupa se utilizara fundamentalmente el Control System Toolbox

5.3. Interpolación lineal

Durante este apartado veremos la imagen que se genera al aplicar el método de interpolación lineal en la imagen original (el código está en el Anexo I)



Figura 5.5: Imagen con interpolación lineal

5.4. Interpolación cúbica

Durante este apartado veremos la imagen que se genera al aplicar el método de interpolación cúbica en la imagen original (el código está en el Anexo I)

Método cúbico

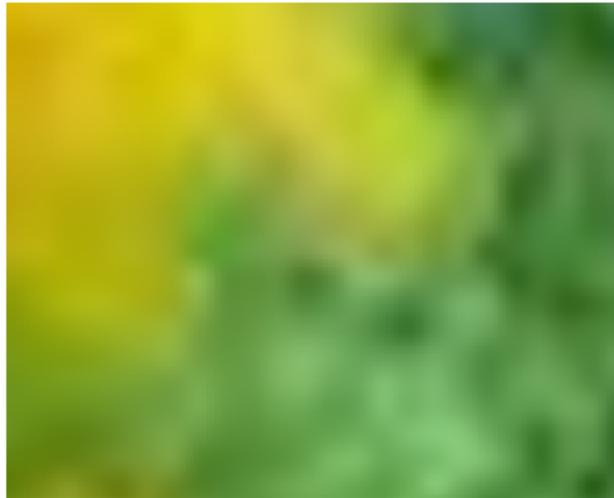


Figura 5.6: Imagen con interpolación cúbica

5.5. Interpolación splines

Durante este apartado veremos la imagen que se genera al aplicar el método de interpolación de splines en la imagen original (el código está en el Anexo I)

Método de splines



Figura 5.7: Imagen con interpolación de splines

Capítulo 6

Conclusiones

Contents

6.1. Ámbito formativo	65
6.2. Ámbito profesional	66
6.3. Ámbito personal	66

Durante este último capítulo se presentarán las conclusiones obtenidas durante la realización del proyecto y estancia en la empresa.

6.1. Ámbito formativo

En primer lugar esta experiencia me ha servido para poner en práctica todos los conocimientos adquiridos durante el Grado de Matemáticas Computacional en la Universitat Jaume I. He contado con una buena formación y me ha permitido desarrollar un trabajo de calidad para la empresa. Algunas de las tecnologías empleadas durante la estancia en prácticas no han sido vistas durante los diferentes cursos académicos o se han visto de una manera superficial, pero me ha servido para aprender JavaScript ya que es un lenguaje en el cual no estaba familiarizada y que se utiliza principalmente en el diseño de interfaces. Por otro lado he adquirido conocimientos nuevos sobre distintas herramientas y tecnologías que no se han visto durante los 4 cursos de la carrera.

Para finalizar considero que ha sido muy útil aprender a desarrollar proyectos de forma correcta, ya que es de gran utilidad para un futuro profesional. Sin esta experiencia, seguro que habría sido mas difícil abordar un trabajo como este.

6.2. Ámbito profesional

En el ámbito profesional debo comentar que ha sido mi primer contacto con una empresa tecnológica en un entorno de trabajo real, durante el periodo en la empresa he podido adquirir conocimientos muy útiles que podré poner en práctica durante mi futura carrera profesional. Mis compañeros de trabajo tenían un gran conocimiento de las herramientas y tecnologías que he empleado en el desarrollo del proyecto que sin duda me ayudaron a abordarlas y llevar a cabo las tareas que debía realizar.

También me gustaría agradecer a la empresa Opentix S.L ofrecerme la oportunidad de realizar mi proyecto de final de grado con ellos y ayudarme en lo que he necesitado durante los meses de trabajo realizado con ellos.

6.3. Ámbito personal

En el ámbito personal he crecido como persona. Me siento capaz de cumplir proyectos, la confianza ha aumentado viendo cómo he podido resolver todos los problemas que se han presentado durante el desarrollo de este proyecto.

También me ha ofrecido la oportunidad de conocer a grandes personas, que han enriquecido mi vida, tanto profesional como personal.

Anexo A

Anexo I

```
A = imread('ave.jpg');  
  
imshow(A);  
F = griddedInterpolant(double(A));  
  
[sx, sy, sz] = size(A);  
xq = (0:5/6:sx)';  
yq = (0:5/6:sy)';  
zq = (1:sz)';  
vq = uint8(F({xq,yq,zq}));  
figure  
imshow(vq)  
title('Mayor_Resoluci n')  
  
xq = (0:1.55:sx)';  
yq = (0:1.55:sy)';  
zq = (1:sz)';  
vq = uint8(F({xq,yq,zq}));  
figure  
imshow(vq)  
title('Menor_Resoluci n')  
  
imshow(A(1:570,10:600,:), 'InitialMagnification','fit')  
zoom(10)  
title('Imagen_Original ,_10x_Zoom')  
  
xq = (1:0.1:sx)';  
yq = (1:0.1:sy)';  
zq = (1:sz)';  
figure
```

```
F.Method = 'linear';  
vq = uint8(F({xq,yq,zq}));  
imshow(vq(1:5700,150:5900,:), 'InitialMagnification', 'fit')  
zoom(10)  
title('M todo_de_interpolaci_n_lineal')
```

```
figure  
F.Method = 'cubic';  
vq = uint8(F({xq,yq,zq}));  
imshow(vq(1:5700,150:5900,:), 'InitialMagnification', 'fit')  
zoom(10)  
title('M todo_de_interpolaci_n_c_bica')
```

```
figure  
F.Method = 'spline';  
vq = uint8(F({xq,yq,zq}));  
imshow(vq(1:5700,150:5900,:), 'InitialMagnification', 'fit')  
zoom(10)  
title('M todo_de_splines')
```

Anexo B

Anexo II

B.1. Demostración teorema fundamental del álgebra

Una demostración del teorema fundamental del álgebra

Teorema 3 *Si $P(z)$ es un polinomio de grado $n \geq 1$, entonces tiene al menos un cero.*

Primero demostraremos la existencia de un cero del polinomio $P(z)$ cuando los coeficientes son números reales.

Sea $P(z) = a_0z^n + a_1z^{n-1} + \dots + a_{n-1}z + a_n$, ($a_0 \neq 0$, $a_n \neq 0$)

Pongamos $z = r(\cos \theta + i \operatorname{sen} \theta)$, entonces podemos escribir

$$\begin{aligned} P(z) &= \sum_{k=0}^n a_{n-k} r^k (\cos k\theta + i \operatorname{sen} k\theta) \\ &= \sum_{k=0}^n a_{n-k} r^k \cos k\theta + i \sum_{k=0}^n a_{n-k} r^k \operatorname{sen} k\theta \end{aligned}$$

Llamemos

$$A = \sum_{k=0}^n a_{n-k} r^k \cos k\theta$$

y

$$B = \sum_{k=0}^n a_{n-k} r^k \operatorname{sen} k\theta$$

Formaremos la función $V = \arctan \frac{B}{A}$ y derivémosla formalmente con respecto a r y θ , entonces:

$$\frac{\partial V}{\partial r} = \frac{A \frac{\partial B}{\partial r} - B \frac{\partial A}{\partial r}}{A^2 + B^2} = f(r, \theta)$$

$$\frac{\partial V}{\partial \theta} = \frac{A \frac{\partial B}{\partial \theta} - B \frac{\partial A}{\partial \theta}}{A^2 + B^2} = g(r, \theta)$$

Consideremos la región $0 \leq r \leq R$, $0 \leq \theta \leq 2\pi$ y supongamos que $A^2 + B^2 \neq 0$ para cada valor de R . En esta región las funciones $f(r, \theta)$ y $g(r, \theta)$ son continuas y además $\frac{\partial f}{\partial \theta} = \frac{\partial g}{\partial r}$ en efecto.

$$\begin{aligned} \frac{\partial f}{\partial \theta} &= \frac{-A^2 \frac{\partial A}{\partial r} \frac{\partial B}{\partial \theta} + B^2 \frac{\partial B}{\partial r} \frac{\partial A}{\partial \theta} + A^2 (A \frac{\partial^2 B}{\partial r \partial \theta} - \frac{\partial B}{\partial r} \frac{\partial A}{\partial \theta} - B \frac{\partial^2 A}{\partial r \partial \theta})}{(A^2 + B^2)^2} \\ &+ \frac{B^2 (\frac{\partial A}{\partial r} \frac{\partial B}{\partial \theta} + A \frac{\partial^2 B}{\partial r \partial \theta} - B \frac{\partial^2 A}{\partial r \partial \theta}) + 2AB (\frac{\partial A}{\partial r} \frac{\partial A}{\partial \theta} - \frac{\partial B}{\partial r} \frac{\partial B}{\partial \theta})}{(A^2 + B^2)^2} = \frac{\partial g}{\partial r} \end{aligned}$$

Como $h(r, \theta) = \frac{\partial f}{\partial \theta} = \frac{\partial g}{\partial r}$ es una función continua en el recinto $0 \leq r \leq R$, $0 \leq \theta \leq 2\pi$ entonces se debe tener que

$$(I) \int_0^{2\pi} \int_0^R h(r, \theta) dr d\theta = \int_0^R \int_0^{2\pi} h(r, \theta) d\theta dr$$

Pero, calculando separadamente las dos integrales repetidas, tenemos

$$\int_0^R \int_0^{2\pi} h(r, \theta) d\theta dr = \int_0^R [f(r, \theta)]_0^{2\pi} dr$$

$$\begin{aligned}
&= \int_0^R \frac{(\sum_{k=0}^n a_{n-k} r^k \cos k\theta)(\sum_{k=0}^n k a_{n-k} r^{k-1} \cos k\theta)}{A^2 + B^2} - \frac{(\sum_{k=0}^n a_{n-k} r^k \sen k\theta)}{A^2 + B^2} \\
&\quad x(-\sum_{k=0}^n k a_{n-k} r^{k-1} \sen k\theta)]_0^{2\pi} dr \\
&= \int_0^R \frac{(\sum_{k=0}^n a_{n-k} r^k)(\sum_{k=0}^n k a_{n-k} r^{k-1}) - (\sum_{k=0}^n a_{n-k} r^k)(\sum_{k=0}^n k a_{n-k} r^{k-1})}{A^2 + B^2} d\theta dr = 0
\end{aligned}$$

Por otro lado,

$$\int_0^{2\pi} \int_0^R h(r, \theta) dr d\theta = \int_0^{2\pi} g(r, \theta) \Big|_0^R d\theta = \int_0^{2\pi} (g(R, \theta) - g(0, \theta)) d\theta > 0$$

en efecto

$$g(r, \theta) = \frac{(\sum_{k=0}^n a_{n-k} r^k \cos k\theta)(\sum_{k=0}^n k a_{n-k} r^k \cos \theta)}{A^2 + B^2} + \frac{(\sum_{k=0}^n a_{n-k} r^k \sen k\theta)(\sum_{k=0}^n k a_{n-k} r^k \sen k\theta)}{A^2 + B^2}$$

es decir,

$$g(r, \theta) = \frac{n a_0^2 r^{2n} + \text{monomios en } r \text{ de grado menor que } 2n}{a_0^2 r^{2n} + \text{monomios en } r \text{ de grado menor que } 2n}$$

asi que

$$\lim_{r \rightarrow \infty} g(r, \theta) = n$$

luego dado $0 < \epsilon < 1$, existe un número N tal que $g(r, \theta) \in (n - \epsilon, n + \epsilon)$ para todo $r > N$.

Si fijamos $R > N$ tendremos entonces que $g(R, \theta) > n - \epsilon$, $g(0, \theta) = 0$ ya que r aparece como potencia ≥ 1 en cada monomio del numerador de $g(r, \theta)$ y en el denominador aparece el término a_n y en los otros monomios la potencia de r es mayor o igual a 1.

$$g(R, \theta) - g(0, \theta) > n - \epsilon$$

Luego

$$\int_0^{2\pi} (g(R, \theta) - g(0, \theta)) d\theta > \int_0^{2\pi} (n - \epsilon) d\theta = (n - \epsilon)2\pi > 0$$

Como los valores de las integrales repetidas en (I) nos resultaron distintas, esto nos indica que la continuidad de la función $h(r, \theta)$ debe fallar por lo menos en un punto z_0 y en este punto debe tenerse que

$$A^2 + B^2 = 0 \Rightarrow A = 0, B = 0 \Rightarrow P(z_0) = 0$$

Consideramos ahora $P(z) = a_0 z^n + \dots + a_{n-1} z + a_n$ ($a_0 \neq 0$) un polinomio de grado n con coeficientes complejos.

Sea $Q(z)$ el polinomio de grado n cuyos coeficientes son los conjugados de a_i ($i = 0, \dots, n$)

$$Q(z) = \bar{a}_0 z^n + \bar{a}_1 z^{n-1} + \dots + \bar{a}_{n-1} z + \bar{a}_n$$

El polinomio $R(z) = P(z)Q(z)$ es de grado $2n$ y tiene coeficientes reales, en efecto

$$\begin{aligned} R(z) &= (a_0 z^n + \dots + a_{n-1} z + a_n)(\bar{a}_0 z^n + \dots + \bar{a}_{n-1} z + \bar{a}_n) \\ &= a_0 \bar{a}_0 z^{2n} + (a_0 \bar{a}_1 + a_1 \bar{a}_0) z^{2n-1} + (a_0 \bar{a}_2 + \bar{a}_0 a_2 + a_1 \bar{a}_1) z^{2n-2} + \dots + \left(\sum_{i=0}^k \bar{a}_i a_{k-i} \right) z^{2n-k} + \dots + a_n \bar{a}_n \end{aligned}$$

Como el polinomio $R(z)$ tiene coeficientes reales, él tiene por lo menos un cero, digamos z_0 , es decir $Q(z_0)P(z_0) = 0$ y esto último implica que $Q(z_0) = 0$ o $P(z_0) = 0$

Si $P(z_0) = 0$ entonces z_0 es un cero de $P(x)$.

Si $Q(z_0) = 0$ entonces \bar{z}_0 es un cero de $P(x)$.

B.2. Semejanza de triángulos

La semejanza de triángulos es una característica que hace que dos o más triángulos sean semejantes.

Dos triángulos son semejantes cuando tienen sus ángulos iguales (o congruentes) y sus lados correspondientes (u homólogos) son proporcionales.

Son lados homólogos los opuestos a ángulos iguales.

Aquí tenemos un caso, donde se ven los elementos homólogos (ángulos y lados) con la igualdad o congruencia de sus ángulos y la proporcionalidad de los lados:

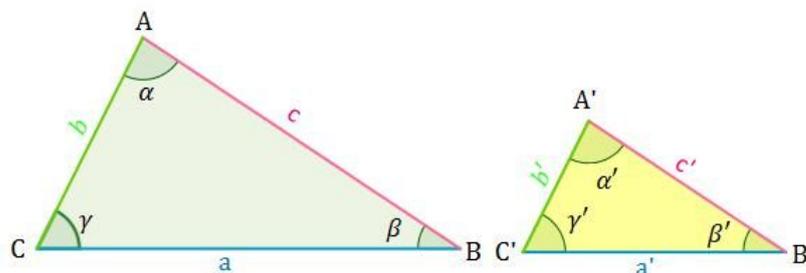


Figura B.1: Semejanza de triángulos.

En los triángulos semejantes se cumplen las condiciones siguientes:

-Los ángulos homólogos son iguales: $\beta = \beta'$, $\alpha = \alpha'$, $\gamma = \gamma'$

Los lados homólogos son proporcionales:

$$\frac{a}{a'} = \frac{b}{b'} = \frac{c}{c'} = r$$

A r se le denomina razón de semejanza.

-Se cumple que la razón de los perímetros de dos triángulos semejantes es también la razón de semejanza y que la razón de sus áreas es el cuadrado de la razón de semejanza:

$$\frac{\text{perímetro}}{\text{perímetro}'} = \frac{a+b+c}{a'+b'+c'} = r$$

$$\frac{\text{área}}{\text{área}'} = r^2$$

Para saber si dos triángulos son semejantes no es necesario conocer sus tres ángulos y sus tres lados.