



7.1 Bibliografía consultada

- **Acuicultura**

Junta Nacional Asesora de Cultivos Marinos online. Disponible en:

<http://www.mapya.es/jacumar/jacumar.asp>

Observatorio español de acuicultura (OESA). Disponible en:

<http://www.observatorio-acuicultura.org/>

- **Visión Artificial**

Gonzalez R. y Woods R. (1993). "Digital Image Processing", Addison-Wesley.

Ollero, Aníbal. Apuntes de la asignatura Sistemas de Percepción y Reconocimiento de Formas.

A. Fernández, Luis. Optimización de la ecualización del histograma en el procesamiento de imágenes digitales

- **Programación**

Bjarne Stroustrup. (2001). “El Lenguaje de programación C++”. Addison Wesley.

García de Jalón, Javier. “Aprenda C++”, UPN

Kernighan, Brian W. y Ritchie, Dennis M. (1991). “El lenguaje de programación C”, Prentice Hall.

- **Manuales**

Matlab Image Processing Toolbox. The Math Works Inc.

- **Memorias**

Munárriz Fernández, David. (2002). Proyecto Fin de Carrera: “Software para muestreo automático mediante visión artificial de jaulas marinas de doradas en piscifactorías”, E.S.I. Universidad de Sevilla.

Mangas Hernández, Ángel Manuel. (2002). Proyecto Fin de Carrera: “Sistema de visión estéreo para el muestreo de estanques de peces en piscifactorías”, E.S.I. Universidad de Sevilla.

De la Rosa Mozo, Jerónimo. (2004). Proyecto Fin de Carrera:” Sistema para la realización de experimentos de estimación de distribución de tamaños en piscifactorías”, E.S.I. Universidad de Sevilla.

7.2 Apéndices

7.2.1 Apéndice I: fichero constantes.h

```
#include <stdlib.h>
```

```
//CONSTANTES IMAGENES
```

```
//tamaño imagen
#define SIZE_X          640
#define SIZE_Y          480
//carpeta imagenes originales
#define PATH_ORIGEN
                        "E:\Proyecto\Imagenes
nuevas\C4_3\"

//carpeta destino
#define PATH_DESTINO
                        "C:\Imag
lubinas\prueba3\"

//nombre imagen entrada
#define IMG_IN          "Imagen"
//numero imagenes a procesar0
#define NUM_IMG        499

//CONSTANTES PREPROCESADO

//realizar preprocesado
#define PREPROCESADO    false
#define DESENTRELAZADO false
#define ECUALIZADO_OPTIMIZADO false
#define ECUALIZADO      false
//modelo iluminacion
#define ILUMINACION     false
#define UMBRALIZACION   true
#define PORCENTAJE_ILU  0.89

//DATOS AUMENTO CONTRASTE

#define MAPEADO          true

#define V_CONTRASTE_A    0.55
#define V_CONTRASTE_B    0.8
#define V_CONTRASTE_C    0.2
#define V_CONTRASTE_D    0.7

//DATOS DETECCION OJOS

#define AREA_MAX         80
#define AREA_MIN         15
#define ELONGACION_MAX   1.45
#define ELONGACION_MIN   0.7
//angulo separacion radios ojos en grados
#define ALPHA            10
//porcentaje radios correctos
#define PORCENTAJE_RADIOS 0.75
```

```
//Datos generales

#define NUMERO_NIVELES                256

//Datos para la corrección de iluminación por filas

#define UMBRAL_SATU_F                  10
#define PORCENTAJE_F                  40
#define UMBRAL_ACUM_F                  50
#define MEDIA_REF_F                    80
#define CONTRASTE_REF_F                180

//Datos para la corrección de iluminación por columnas

#define UMBRAL_SATU_C                  10
#define PORCENTAJE_C                  40
#define UMBRAL_ACUM_C                  50
#define MEDIA_REF_C                    80
#define CONTRASTE_REF_C                180

//Entorno de vecindad

#define DERECHA                        7
#define IZQUIERDA                      13

/*Procesamiento de los blobs*/
#define MAX_BLOBS                      1000
#define MIN_BLOB_SIZE                  25
#define MIN_BLOB_ELONG                 4

#define PI                              3.1416
```

7.2.2 Apéndice II: clase CImagen

```
#if !defined(CLASE_IMAGEN)
#define CLASE_IMAGEN

#include <stdio.h>
#include <windows.h>
#include "Constantes.h"

class CImagen{

public:

    int ancho,alto;
    /* almacena las coordenadas del centro de los blobs detectados */
```

```
int ***tabla_blobs;
/* almacena las coordenadas del centro de los ojos detectados */
int ***tabla_ojos;
unsigned char ***tabla;

struct CabeceraBMP {

char Tipo[2];
unsigned int Tamano;
unsigned int Reservado;
unsigned int Offset;
unsigned int TamCab;
unsigned int Ancho;
unsigned int Alto;
unsigned short Planos;
unsigned short BitsPorPixel;
unsigned int Compresion;
unsigned int TamImagen;
unsigned int XPixelsPorM;
unsigned int YPixelsPorM;
unsigned int ColoresUsados;
unsigned int ColoresImportantes;
}cab;

public:
    CImagen();
    int AbrirImagen(char *);
    int AbrirImagen(int,int);
    int DescargarImagen(void);
    int InvertirImagen(CImagen);

    /*Compensacion iluminacion por filas*/
    int CompensarIluminacionF(CImagen,unsigned char,int,int,int,int);
    int DeterminarLineasSaturadas(int **,unsigned char,int);
    int FiltradoLineasSaturadas(int **,int *);
    int CompensacionIluminacionFilas(int,int,int,int *);
    /*Compensacion iluminacion por columnas  AJM 28-5-04*/
    int CompensarIluminacionC(CImagen,unsigned char,int,int,int,int);
    int DeterminarColumnasSaturadas(int **,unsigned char,int);
    int FiltradoColumnasSaturadas(int **,int *);
    int CompensacionIluminacionColumnas(int,int,int,int *);

    int CopiarImagen(CImagen);

    void imErode();
    void imDilate();
    void imOpen(int i);
    int Mediana(void);
    void OrdenarVecinos(unsigned char * vecinos);
```

```
int EliminarEntrelazado(CImagen);
int RecortarImagenFila(CImagen,int);
int GuardarImagen(char *);
int Preprocesar(void);
int Ecuallar(CImagen);
int EcuallizadoOptimizado(CImagen);
int Mapeado(CImagen);
int Umbralizar(int,CImagen);
int LimpiarPixelSolitario(void);
int *DetectarOjos(int,CImagen);
int *BlobsBySize(int);
int *BlobsRedondeados(int,int *);
int *BlobsRadios(CImagen,int,int *);

private:

int ReservarEspacio(void);
int ComprobarCabecera(FILE *);
int CargarDatos(FILE *fichero);
};

#endif
```